Post-Quantum Anonymity of Kyber

Varun Maram[1] and Keita Xagawa[2]

[1] Department of Computer Science,
ETH Zurich, Switzerland.
[2] NTT Social Informatics Laboratories, Japan.
vmaram@inf.ethz.ch,
keita.xagawa.zv@hco.ntt.co.jp

Abstract. Kyber is a key-encapsulation mechanism (KEM) that was recently selected by NIST in its PQC standardization process; it is also the *only* scheme to be selected in the context of public-key encryption (PKE) and key establishment. The main security target for KEMs, and their associated PKE schemes, in the NIST PQC context has been IND-CCA security. However, some important modern applications also require their underlying KEMs/PKE schemes to provide *anonymity* (Bellare *et al.*, ASIACRYPT 2001). Examples of such applications include anonymous credential systems, cryptocurrencies, broadcast encryption schemes, authenticated key exchange, and auction protocols. It is hence important to analyze the compatibility of NIST's new PQC standard in such "beyond IND-CCA" applications.

Some starting steps were taken by Grubbs *et al.* (EUROCRYPT 2022) and Xagawa (EUROCRYPT 2022) wherein they studied the anonymity properties of most NIST PQC third round candidate KEMs. Unfortunately, they were unable to show the anonymity of Kyber because of certain technical barriers.

In this paper, we overcome said barriers and resolve the open problems posed by Grubbs *et al.* (EUROCRYPT 2022) and Xagawa (EUROCRYPT 2022) by establishing the anonymity of Kyber, and the (hybrid) PKE schemes derived from it, in a post-quantum setting. Along the way, we also provide an approach to obtain tight IND-CCA security proofs for Kyber with *concrete* bounds; this resolves another issue identified by the aforementioned works related to the post-quantum IND-CCA security claims of Kyber from a provable security point-of-view. Our results also extend to Saber, a NIST PQC third round finalist, in a similar fashion.

Keywords: anonymity, post-quantum cryptography, NIST PQC standardization, KEM, hybrid PKE, quantum random oracle model

1 Introduction

Roughly six years after kicking-off its post-quantum cryptography (PQC) standardization process, the US National Institute of Standards and Technology (NIST) has finally announced the first set of cryptographic algorithms that will be standardized (along with a set of alternate algorithms that will be considered for future standardization) [2]. Among this first set of algorithms, CRYSTALS-Kyber [39] (or Kyber, for short) is the *only* key-encapsulation mechanism (KEM) selected by NIST for standardization, in the context of public-key encryption (PKE) and key-establishment. One of NIST's main criteria for evaluating and selecting PQC standards in the PKE/KEM category was on the algorithms' ability to offer semantic security with respect to adaptive chosen ciphertext attacks (a.k.a. IND-CCA security). IND-CCA security is widely accepted as a standard notion of security for PKE schemes and KEMs since the property suffices for many important use cases. However, as a NIST PQC standard, since Kyber is intended to be widely used for decades to come, it is also important to study the scheme's compatibility with emerging modern applications that require security properties beyond IND-CCA.

One such important security property is *anonymity* (or key privacy). Roughly speaking, a PKE scheme is said to be anonymous [6] if a ciphertext hides the receiver's information by not leaking anything about the public key used for encryption; anonymous KEMs are defined analogously [26,44]. Such anonymous cryptographic primitives are fundamental in several deployed privacy-enhancing systems, such as anonymous cryptocurrencies like Zcash [8], anonymous broadcast encryption schemes [5,35], anonymous credential systems [14], anonymous authenticated key exchange [12,22,23,40], auction protocols [38], and so on. The recent works of [26,44] have hence looked into anonymity properties of the NIST PQC third round candidate KEMs, and the hybrid PKE schemes derived from them via the "KEM-DEM" paradigm [15]. Collectively, both those works have established the post-quantum anonymity of all nine candidate KEMs except for three, which unfortunately includes the current standard Kyber (the other two KEMs being Saber [18] and Streamlined NTRU Prime [9]).

To see why the works of [26,44] could not establish the anonymity of Kyber, it helps to first look at how the NIST PQC candidate KEMs are constructed. The KEM candidates first specify a weakly secure (e.g., IND-CPA secure) "base" PKE scheme and then apply some variant of the *Fujisaki-Okamoto (FO) transform* [24,25,19,27] to obtain their respective KEMs. The "original" FO transforms of [24,25,19,27] were heavily analyzed in the idealized *Random Oracle Model (ROM)* [7], and later, in the *Quantum ROM (QROM)* [11] which is relevant for studying post-quantum security; it was shown in a long sequence of works (e.g., [37,30,10,33,20,29]) that such original transforms boost an IND-CPA secure PKE scheme to an IND-CCA secure KEM in the QROM. In the context of anonymity, it was shown in [26,44] that the FO transforms also elevate a weakly anonymous (i.e., ANO-CPA secure) base PKE scheme to a strongly anonymous (i.e., ANO-CCA secure) KEM in the QROM.

However, the specific variant of FO transform used in Kyber deviates quite significantly from the original transforms above. At a high-level, Kyber hashes more "intermediate" values in its internal computations than is the case in FO transforms in the literature. At the same time, this additional hashing is done in a way which creates barriers in applying the proof strategies used in [26,44] to show the anonymity boosting properties of the original FO transforms in the QROM. Hence, this raises the following question:

Is Kyber (provably) ANO-CCA secure in the QROM?

At the same time, as observed in [26,44], the additional hashing in Kyber also acts as a barrier in proving even the scheme's IND-CCA security in the QROM with the concrete bounds claimed in its specification document [4]. Given the importance placed on IND-CCA security in the NIST PQC standardization process, this raises another question:

Can we obtain a (tight) proof of IND-CCA security for Kyber in the QROM with concrete bounds?

1.1 Our Contributions

We answer the above questions in the affirmative by presenting the following results, thereby resolving the corresponding open problems posed in [26,44]:

- We show that Kyber and the hybrid PKE schemes derived from it are ANO-CCA secure in the QROM, under the standard hardness assumption of solving the *module learning-with-error (MLWE) problem* [13,34].
- We describe an approach to obtain tight IND-CCA security with *concrete* bounds for Kyber in the QROM, under the MLWE hardness assumption.

It is worth mentioning that the NIST PQC third round finalist Saber [18] implements the *same* variant of FO transform as Kyber in its KEM construction. Hence, our above results on anonymity and tight IND-CCA security also apply to Saber in a similar fashion, where we would instead need to rely on the hardness of solving the *module learning-with-rounding (MLWR) problem* [17].

We hope that our above results provide further confidence to cryptographic scheme designers in using the new PQC standard Kyber not only in general-purpose applications that need IND-CCA security but also in emerging modern applications that require anonymity.

1.2 Technical Overview

Here we give a high-level description of our approach to obtain proofs of anonymity (i.e., ANO-CCA security) and (tight) IND-CCA security for Kyber in the QROM. We first focus on the familiar setting of IND-CCA security and later consider ANO-CCA security.

IND-CCA Security of Kyber. We begin by first describing an *alternative* – and "*simpler*" – approach to prove IND-CCA security of Kyber in the QROM, and then contrasting it with our approach. As noted above, virtually all NIST PQC candidate KEMs, including Kyber, use variants of the FO transformation in their respective KEM constructions. Before discussing the specific variant

```
$\mathsf{KGen}'$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
2 : $\boxed{s \leftarrow_{\$} \mathcal{M}}$
3 : $s \leftarrow \perp$
4 : $\mathsf{sk}' \leftarrow (\mathsf{sk}, s)$
5 : return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
1 : $m \leftarrow_{\$} \mathcal{M}$
2 : $r \leftarrow G_r(m)$
3 : $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
4 : $\overline{k} \leftarrow G_k(m)$
5 : return $(c, \overline{k})$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
1 : Parse $\mathsf{sk}' = (\mathsf{sk}, s)$
2 : $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 : $r' \leftarrow G_r(m')$
4 : $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 : if $c' = c$ then
6 :   return $G_k(m')$
7 : $\boxed{\textbf{else return } G_k(s, c)}$
8 : else return $\perp$
```

Fig. 1. The KEMs $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, G_r, G_k]$ and $\boxed{\mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, G_r, G_k]}$. Here $\mathcal{M}$ is the message space of $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ and G_r, G_k are hash functions with appropriate domain and co-domain. For notational simplicity, we set $s \leftarrow \perp$ for $\mathsf{FO}_m^{\perp}$.

used by Kyber, let us first consider the *standard* FO transforms introduced by Dent [19] and Hofheinz et al. [27], namely the *explicitly-rejecting* $\mathsf{FO}_m^{\perp}$ and the *implicitly-rejecting* $\mathsf{FO}_m^{\not\perp}$, described in Figure 1.

For ease of exposition, we consider a simplified version of Kyber's FO variant where the only main difference compared to $\mathsf{FO}_m^{\perp}$ is that, instead of stopping at "$\overline{k} \leftarrow G_k(m)$" (Line 4 in $\mathsf{Encap}(\mathsf{pk})$, Fig. 1) during encapsulation, there is an extra layer of hashing to compute the final encapsulated key. Namely, Kyber outputs keys of the form "$k \leftarrow H'(\overline{k}, H(c))$" where H, H' are two additional hash functions; decapsulation proceeds analogously where instead of returning a $\perp$ when rejecting a ciphertext, Kyber *implicitly* rejects by returning $H'(s, H(c))$. Hence, (this simplified version of) Kyber can be seen as a "wrapper" scheme w.r.t. the $\mathsf{FO}_m^{\perp}$ KEM with appropriate modifications to the encapsulation and decapsulation steps. As a result, the IND-CCA security of Kyber can be easily shown by relying on the IND-CCA security of the underlying $\mathsf{FO}_m^{\perp}$ KEM.

To sketch out the proof, we start with the IND-CCA security game w.r.t. (the simplified) Kyber where the adversary gets a challenge ciphertext c^* and the *real* encapsulated key "$H'(\overline{k}^*, H(c^*))$" (refer to Subsection 2.2 for a precise description of the IND-CCA security games for KEMs). We then modify the game via the following "hybrids":

1. In the first hybrid, we provide the adversary with a new encapsulated key "$H'(\overline{k}', H(c^*))$", where $\overline{k}'$ is an independent and uniformly random value. This modification is justified by relying on IND-CCA security of the underlying $\mathsf{FO}_m^{\perp}$ KEM. Because note that $\overline{k}^*$ can be seen as the "real" encapsulated key of the $\mathsf{FO}_m^{\perp}$ KEM and $\overline{k}'$ a "random" key, and IND-CCA security of $\mathsf{FO}_m^{\perp}$ implies (computational) indistinguishability of both these keys. One important thing worth noting here is that in the reduction to IND-CCA security of $\mathsf{FO}_m^{\perp}$, we can simulate the decapsulation oracle of Kyber as follows. We

first sample the secret $s \leftarrow_{\$} \mathcal{M}$. Then to simulate the "Kyber-decapsulation" of a ciphertext c, we first perform the "$\mathsf{FO}_m^{\perp}$-decapsulation" of c: if the result is a key $\overline{k}$, we return the "Kyber-key" as $H'(\overline{k}, H(c))$; if the result is $\perp$, we return the "Kyber-key" as $H'(s, H(c))$. Note that for this reduction to work, it is crucial that the underlying FO transform, $\mathsf{FO}_m^{\perp}$, is explicitly rejecting, in order to perfectly simulate the rejection of ciphertexts during decapsulation.

2. In the second and final hybrid, we again switch back to the IND-CCA security game w.r.t. Kyber where the adversary gets a uniformly *random* encapsulated key "$\hat{k}$" which is independent of c^*. This modification is again justified by relying on the *pseudorandomness* provided by the quantum random oracle $H'(\overline{k}', \cdot)$: i.e., since the "PRF key" $\overline{k}'$ is independent of c^*, one can argue the (statistical) indistinguishability of the keys "$H'(\overline{k}', H(c^*))$" and "$\hat{k}$".

The IND-CCA security of (the simplified) Kyber in the QROM hence follows since the adversary cannot efficiently distinguish between the real and random encapsulated keys "$H'(\overline{k}^*, H(c^*))$" and "$\hat{k}$" respectively in the above hybrids.

However, a major issue with the above approach to prove *concrete* (and *tight*) IND-CCA security of Kyber is related to our dependence on the IND-CCA security of $\mathsf{FO}_m^{\perp}$ *in the QROM* in the first place. IND-CCA security of the $\mathsf{FO}_m^{\perp}$ transform, with concrete bounds, has been notoriously hard to prove in the QROM. To put things in context, let us first consider $\mathsf{FO}_m^{\not\perp}$, the *implicitly-rejecting* variant of $\mathsf{FO}_m^{\perp}$. A long sequence of prior works [37,30,10,28,33] provided concrete IND-CCA security proofs for $\mathsf{FO}_m^{\not\perp}$ in the QROM, with each follow-up improving the tightness of the corresponding reduction. For example, Kuchta *et al.* [33] were the first to provide a security proof that avoided a square-root advantage loss w.r.t. the weak (IND-CPA/OW-CPA) security of the underlying PKE scheme; this loss seemed inherent with previous reductions for the FO transforms in the QROM. To also showcase the relative simplicity of analyzing the IND-CCA security of $\mathsf{FO}_m^{\not\perp}$ in the QROM, Unruh [43] showed a framework for *formally verifying* the corresponding post-quantum security proof of the implicitly-rejecting transform provided in [28].

When it comes to the *explicitly-rejecting* $\mathsf{FO}_m^{\perp}$ transform, the story is arguably more complicated. Looking at prior work, some starting steps were taken in [41,27,31,3] in this regard wherein concrete IND-CCA security proofs for *modified* versions of the $\mathsf{FO}_m^{\perp}$ transform – which include an additional "key confirmation" hash in the ciphertext – were provided (however, security proofs in [41,3] were later found to have bugs in them [3]). The *unmodified* $\mathsf{FO}_m^{\perp}$ transform was later analyzed in [46,32] in the QROM; however, the provided security proofs had some subtle gaps [20]. Quite recently, these gaps were resolved in [20,29] resulting in the first IND-CCA security proofs for the original $\mathsf{FO}_m^{\perp}$ transform in the QROM with concrete bounds. However, there are a couple of issues:

– The IND-CCA security analyses of $\mathsf{FO}_m^{\perp}$ by Don et al. [20] and Hövelmanns et al. [29] assume certain computational and statistical properties of the underlying PKE scheme which are not well-studied w.r.t. the NIST PQC candidates – *especially Kyber*. These properties include γ*-spreadness*, so-called *Find Failing Plaintext (FFP) security* (as introduced in [29]), etc.

- Even if the above properties are properly analyzed, the resulting IND-CCA security bounds for the final $\mathsf{FO}_m^{\perp}$-based KEM are non-tight when compared to the corresponding state-of-the-art bounds for the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$. E.g., all known IND-CCA security proofs for $\mathsf{FO}_m^{\perp}$ transform in the QROM incur a square-root advantage loss w.r.t. passive security of the underlying PKE scheme. This is in contrast to the tight proof of IND-CCA security for $\mathsf{FO}_m^{\not\perp}$ shown in [33]. In other words, we would also incur these non-tight bounds in our "wrapper-based" IND-CCA security analysis of Kyber in the QROM, when relying on the corresponding post-quantum security of $\mathsf{FO}_m^{\perp}$.

This brings us to one of the main technical contributions of this paper. In essence, we provide a way to obtain tight proofs of IND-CCA security for Kyber in the QROM by salvaging the above "wrapper-based" approach – *even when the underlying FO transform is implicitly-rejecting.* As noted in the above reduction, we crucially relied on the explicit-rejection of $\mathsf{FO}_m^{\perp}$ in order to perfectly simulate decapsulation oracles. But if we start with the $\mathsf{FO}_m^{\not\perp}$ transform, it is not so straightforward how to simulate the "Kyber-decapsulation" oracle using the "$\mathsf{FO}_m^{\not\perp}$-decapsulation" oracle especially when the latter oracle rejects ciphertexts; as described in Figure 1 (Line 9), the rejection output $G_k(s, c)$ still "looks" like a valid key.

To resolve the above simulation issue, we start with the $\mathsf{FO}_m^{\not\perp}$ transform and modify its decapsulation algorithm in a way such that the overall IND-CCA security of the transform in the QROM is affected negligibly (in a statistical sense). Similarly, we also modify the decapsulation procedure used in the actual Kyber scheme such that (i) the IND-CCA security of the original and modified schemes are statistically equivalent, and (ii) the IND-CCA security of the modified scheme can be reduced to the IND-CCA security of the modified $\mathsf{FO}_m^{\not\perp}$ transform wherein we can now simulate the "modified-Kyber-decapsulation" oracle using the "modified-$\mathsf{FO}_m^{\not\perp}$-decapsulation" oracle perfectly in the corresponding reduction. It is then not hard to see that this *indirectly* allows us to base IND-CCA security of the actual Kyber scheme on that of the unmodified $\mathsf{FO}_m^{\not\perp}$ transform, with a negligible loss in tightness; full details of our security proof can be found in Section 4.

But one thing we would like to stress is that our current IND-CCA security proof for Kyber in Section 4 is non-tight in the sense that we still incur a square-root advantage loss w.r.t. passive security of the underlying PKE scheme mentioned above. This is because we are currently basing the IND-CCA security of Kyber on the (non-tight) IND-CCA security of $\mathsf{FO}_m^{\not\perp}$ proven in [30,37] in the QROM, which incurs a similar square-root loss. The reason we are not relying on the tighter proof of IND-CCA security for $\mathsf{FO}_m^{\not\perp}$ shown in [33] – which avoids such a loss – is that their tight proof makes an additional assumption on the underlying PKE scheme: namely, that the scheme satisfies a property called *injectivity* (as defined in [10]). However a detailed analysis of Kyber's injectivity is lacking, particularly in the context of NIST's PQC standardization process, and we also consider it out of the scope of our work. At the same time, this showcases an advantage of our "wrapper-based" approach w.r.t. the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$ in that, if the injectivity of Kyber is well established in the future, then one can simply "plug in" [33]'s tight IND-CCA security result for $\mathsf{FO}_m^{\not\perp}$ in our analysis in Section 4 as a *drop-in replacement* to essentially obtain a tight proof of IND-CCA security for Kyber in the QROM.

ANO-CCA Security of Kyber. Now when it comes to the main focus of this paper, i.e., the anonymity of Kyber in the QROM, we follow the framework of [44]. Namely, we instead show that Kyber satisfies a stronger security notion called *strong pseudorandomness (or, SPR-CCA security)*. A KEM is said to be SPR-CCA secure if, roughly speaking, an adversary cannot distinguish a *real* ciphertext/encapsulated-key pair (c^*, k^*) from a *random* pair (c', k') where c' is a random ciphertext and k' is a random key (see Subsection 2.2 for a formal definition of SPR-CCA security where we also need to consider a *simulator* to specify what we mean by a "random" ciphertext c').

It was shown in [44] that SPR-CCA security straightforwardly implies ANO-CCA security. The key insight used in [44] is that since SPR-CCA security is a "single key-pair notion" like IND-CCA security (i.e., the corresponding security game involves a single KEM key-pair), it is easier to extend the IND-CCA security analysis of a KEM to also show its SPR-CCA security than trying to directly prove its ANO-CCA security; note that ANO-CCA security is a "double key-pair notion" and hence would involve simulating *two* different decapsulation oracles in the security analysis.

Following our above discussion on IND-CCA security of Kyber in the QROM, it is straightforward to show its SPR-CCA security by relying on the same strong pseudorandomness of $\mathsf{FO}_m^{\perp}$-based KEMs by adopting the "wrapper-based" approach. But as noted above, since proving IND-CCA security of $\mathsf{FO}_m^{\perp}$ has been a complicated affair, one can expect the same when it comes to proving "beyond IND-CCA" security properties (e.g., SPR-CCA) of the explicitly-rejecting transform. In fact, we consider extending the IND-CCA security analysis of $\mathsf{FO}_m^{\perp}$ in [20,29] to other important properties, such as SPR-CCA security, in the QROM beyond the scope of this paper, and leave it as an open problem.

In contrast, SPR-CCA security of the implicitly-rejecting $\mathsf{FO}_m^{\not\perp}$ in the QROM was already shown in [44], further indicating the simplicity of analyzing $\mathsf{FO}_m^{\not\perp}$ in the QROM – when compared to its explicitly-rejecting counterpart – even w.r.t. security properties beyond IND-CCA. Hence, our above "wrapper-based" approach w.r.t. the underlying $\mathsf{FO}_m^{\not\perp}$ transform can be used to also show SPR-CCA security – and hence, ANO-CCA security – of Kyber in the QROM; in such an approach (which is presented in detail in Section 5), we need to introduce additional hybrids to replace the real ciphertext c^* with a random ciphertext c'. This showcases yet another advantage of using our approach: *quantitatively*, not only does Kyber inherit existing tight (IND-CCA) security bounds for $\mathsf{FO}_m^{\not\perp}$ in the QROM as seen above, but also *qualitatively*, Kyber inherits "beyond IND-CCA" security properties (such as SPR-CCA) of $\mathsf{FO}_m^{\not\perp}$ in the post-quantum setting.

2 Preliminaries

Notations. We denote $\lambda \in \mathbb{N}$ to be the security parameter. We sometimes omit writing λ when describing cryptosystems if it is clear from the context. PPT and QPT stand for probabilistic polynomial time and quantum polynomial time respectively. We use the standard O-notations. A function $f(\lambda)$ is said to be *negligible* if $f(\lambda) = \lambda^{-\omega(1)}$. For a finite set S, we write "$x \leftarrow_{\$} S$" to denote that x is sampled uniformly at random from S. The value $[x = y]$ is defined to be 1 if $x = y$ and 0 otherwise. 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 use "$\mathcal{A}^O$" to denote that the algorithm $\mathcal{A}$ has access to the oracle O; we'll also make it clear whether $\mathcal{A}$ has *classical* or *quantum* access to O in the description of our setting.

2.1 Quantum Random Oracle Model

Roughly speaking, the quantum random oracle model (QROM) is an idealized model where a hash function is modeled as a publicly and quantumly accessible random oracle. In this paper, we model a quantum oracle $O\colon \{0,1\}^n \to \{0,1\}^m$ as a mapping $|x\rangle |y\rangle \mapsto |x\rangle |y \oplus O(x)\rangle$, where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$. Refer to [11] for a more detailed description of the model.

We now review some useful lemmas in the QROM. The first lemma describes the collision resistance of quantum random oracles.

Lemma 1 ([45, Theorem 3.1]). *There is a universal constant* C (< 648) *such that the following holds: Let* $\mathcal{X}$ *and* $\mathcal{Y}$ *be finite sets. Let* $H\colon \mathcal{X} \to \mathcal{Y}$ *be a random oracle. If an unbounded-time quantum adversary* $\mathcal{A}$ *makes a query to* H *at most* q *times, then we have* $\Pr[H(x_0) = H(x_1) \wedge x_0 \neq x_1 : (x_0, x_1) \leftarrow \mathcal{A}^H] \leq \frac{C(q+1)^3}{|\mathcal{Y}|}$*, where all oracle accesses of* $\mathcal{A}$ *can be quantum.*

The second lemma intuitively states that a quantum random oracle can be used as a *quantum-accessible* pseudorandom function, even if the distinguisher is given full access to the quantum random oracle in addition to the PRF oracle.

Lemma 2 ([30, Lemma 4]). *Let* $H\colon \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ *and* $R\colon \mathcal{X} \to \mathcal{Y}$ *be two independent quantum random oracles. Define the oracles* $F_0 = H(k, \cdot)$*, where we have the "PRF key"* $k \leftarrow_{\$} \mathcal{K}$*, and* $F_1 = R(\cdot)$*. Consider an oracle algorithm/distinguisher* A^{H,F_i} $(i \in \{0,1\})$ *that makes at most* q *queries to* H*. Then we have* $|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]| \leq \frac{2q}{\sqrt{|\mathcal{K}|}}$.

The lemmas below provide a generic reduction from a hiding-style property (indistinguishability) to a one-wayness-style property (unpredictability) in the QROM. It is also popularly known as the *One-Way To Hiding (OW2H) lemma* in the literature, originally appearing in [42]. We first state the original OW2H lemma of [42] and later state a generalized version of the OW2H lemma from [3]. As will be seen in Section 4, different parts of our security analysis of Kyber use different versions of the OW2H lemma for the sake of convenience.

Lemma 3 (Original OW2H [42]). *Let $H\colon \mathcal{X} \to \mathcal{Y}$ be a quantum random oracle. Consider an oracle algorithm A^H that makes at most q queries to H. Let B^H be an oracle algorithm that on input x does the following: picks $i \leftarrow_{\$} \{1, \ldots, q\}$ and $y \leftarrow_{\$} \mathcal{Y}$, runs $A^H(x, y)$ until (just before) the i-th query, measures the argument of the query in the computational basis and outputs the measurement outcome (if A makes less than i queries, B outputs $\perp \notin \mathcal{X}$). Let*

$$P_A^1 = \Pr[1 \leftarrow A^H(x, H(x)) : x \leftarrow_{\$} \mathcal{X}]$$
$$P_A^2 = \Pr[1 \leftarrow A^H(x, y) : x \leftarrow_{\$} \mathcal{X},\ y \leftarrow_{\$} \mathcal{Y}]$$
$$P_B = \Pr[x \leftarrow B^H(x) : x \leftarrow_{\$} \mathcal{X}].$$

Then, we have $|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$.

Lemma 4 (Generalized OW2H [3, Theorem 3]). *Let $\mathcal{S} \subseteq \mathcal{X}$ be random. Let $G, H\colon \mathcal{X} \to \mathcal{Y}$ be random functions 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 q queries to its corresponding oracle (either G or H).*[3] *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. Let*

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

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_B.

2.2 Cryptographic Primitives

We review the primitives of PKE and KEM here. Data Encapsulation Mechanism (or, DEM) is described in Appendix A.1.

Public Key Encryption (PKE): The model for PKE schemes is summarized as follows:

Definition 1. *A PKE scheme* PKE *consists of the following triple of PPT algorithms* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*:*

[3] Strictly speaking, the generalized OW2H lemma of [3] takes into account the *parallel* oracle queries made by A by having q to be the so-called *query depth* of A. In this paper, we won't consider parallel queries of A for the sake of simplicity and denote q to be the *query number* of A. But our subsequent analysis of Kyber can be modified to also consider parallel oracle queries in a straightforward way.

- $\mathsf{KGen}(1^\lambda; r_g) \to (\mathsf{pk}, \mathsf{sk})$*: a key-generation algorithm that on input* 1^λ*, where* λ *is the security parameter, and randomness* $r_g \in \mathcal{R}_{\mathsf{KGen}}$*, outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$*.* pk *and* sk *are called the public/encryption key and private/decryption key, respectively.*
- $\mathsf{Enc}(\mathsf{pk}, m; r_e) \to c$*: an encryption algorithm that takes as input encryption key* pk*, message* $m \in \mathcal{M}$*, and randomness* $r_e \in \mathcal{R}_{\mathsf{Enc}}$*, and outputs ciphertext* $c \in \mathcal{C}$*.*
- $\mathsf{Dec}(\mathsf{sk}, c) \to m/\bot$*: a decryption algorithm that takes as input decryption key* sk *and ciphertext* c *and outputs message* $m \in \mathcal{M}$ *or a rejection symbol* $\bot \notin \mathcal{M}$*.*

Definition 2 (PKE Correctness [27]). *We say that* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ-correct *if*

$$\underset{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KGen}(1^\lambda)}{\mathrm{Exp}}[\max_{m\in\mathcal{M}} \Pr[\mathsf{Dec}(\mathsf{sk}, c) \neq m : c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)]] \leq \delta.$$

If $\delta = 0$*, then we just say that* PKE *is* perfectly correct*.*

Definition 3 (PKE Security). *Let* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme. For any adversary* $\mathcal{A}$ *and* $\mathrm{GOAL} \in \{\mathrm{IND}, \mathrm{SPR}, \mathrm{ANO}\}$*, we define* $\mathcal{A}$*'s* GOAL-CCA *advantage against* PKE *(w.r.t. a simulator* $\mathcal{S}$ *when* $\mathrm{GOAL} = \mathrm{SPR}$*) as follows:*

$$\mathbf{Adv}^{\text{GOAL-CCA}}_{\mathsf{PKE}[,\mathcal{S}]}(\mathcal{A}) := \left|\Pr[\mathbf{Expt}^{\text{GOAL-CCA}}_{\mathsf{PKE}[,\mathcal{S}],\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|,$$

where $\mathbf{Expt}^{\text{GOAL-CCA}}_{\mathsf{PKE}[,\mathcal{S}],\mathcal{A}}(\lambda)$ *is an experiment described in Figure 2. For* $\mathrm{GOAL} \in \{\mathrm{IND}, \mathrm{SPR}, \mathrm{ANO}\}$*, we say that* PKE *is* GOAL-CCA-secure *if (there exists a QPT simulator* $\mathcal{S}$ *when* $\mathrm{GOAL} = \mathrm{SPR}$ *such that)* $\mathbf{Adv}^{\text{GOAL-CCA}}_{\mathsf{PKE}[,\mathcal{S}]}(\mathcal{A})$ *is negligible (in* λ*) for any QPT adversary* $\mathcal{A}$*. We say that* PKE *is* GOAL-CPA-secure *if it is* GOAL-CCA-secure without *giving* $\mathcal{A}$ *access to decryption oracle.*

Definition 4 (Strong Disjoint Simulatablity [37,36,44]). *Let* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme and* $\mathcal{S}$ *be a QPT algorithm/simulator. For any adversary* $\mathcal{A}$*, we define* $\mathcal{A}$*'s* SDS-IND *advantage against* PKE*, w.r.t.* $\mathcal{S}$*, as follows:*

$$\mathbf{Adv}^{\text{SDS-IND}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A}) := \left|\Pr[\mathbf{Expt}^{\text{SDS-IND}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}}(\lambda) = 1] - \frac{1}{2}\right|,$$

where $\mathbf{Expt}^{\text{SDS-IND}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}}(\lambda)$ *is an experiment described in Figure 2. In addition, we define disjointness as*

$$\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}} = \Pr[c \in \mathsf{Enc}(\mathsf{pk}, \mathcal{M}) : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}, c \leftarrow \mathcal{S}(1^\lambda)].$$

We say that PKE *is* strongly disjoint-simulatable *if there exists a QPT simulator* $\mathcal{S}$ *such that* $\mathbf{Adv}^{\text{SDS-IND}}_{\mathsf{PKE},\mathcal{S}}(\mathcal{A})$ *is negligible for any QPT adversary* $\mathcal{A}$ *and* $\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}$ *is negligible in* λ*.*

```
$\mathbf{Expt}^{\text{IND-CCA}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$(m_0, m_1, \text{state}) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot)}(\mathsf{pk})$
$b \leftarrow_\$ \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(c^*, \text{state})$
$\mathbf{return}\ [b' = b]$
```

```
$\mathbf{Expt}^{\text{SPR-CCA}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$(m, \text{state}) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot)}(\mathsf{pk})$
$b \leftarrow_\$ \{0,1\}$
$c_0^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
$c_1^* \leftarrow \mathcal{S}(1^\lambda)$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}(\cdot)}(c_b^*, \text{state})$
$\mathbf{return}\ [b' = b]$
```

```
$\mathbf{Expt}^{\text{SDS-IND}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$m \leftarrow_\$ \mathcal{M};\ c_0^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$
$c_1^* \leftarrow \mathcal{S}(1^\lambda)$
$b' \leftarrow \mathcal{A}(\mathsf{pk}, c_b^*)$
$\mathbf{return}\ [b' = b]$
```

```
$\textsc{Dec}_a(c)$
$\mathbf{if}\ c = a\ \mathbf{then\ return}\ \perp$
$m \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$\mathbf{return}\ m$
```

```
$\textsc{Dec}_a(\beta, c)$
$\mathbf{if}\ c = a\ \mathbf{then\ return}\ \perp$
$m \leftarrow \mathsf{Dec}(\mathsf{sk}_\beta, c)$
$\mathbf{return}\ m$
```

```
$\mathbf{Expt}^{\text{ANO-CCA}}_{\mathsf{PKE},\mathcal{A}}(\lambda)$
$(\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KGen}(1^\lambda)$
$(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$
$(m, \text{state}) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$b \leftarrow_\$ \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m)$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot,\cdot)}(c^*, \text{state})$
$\mathbf{return}\ [b' = b]$
```

Fig. 2. Games for PKE schemes

Key Encapsulation Mechanism (KEM): The model for KEM schemes is summarized as follows:

Definition 5. *A KEM scheme* KEM *consists of the following triple of polynomial-time algorithms* $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$*:*

- $\mathsf{KGen}(1^\lambda; r_g) \rightarrow (\mathsf{pk}, \mathsf{sk})$*: a key-generation algorithm that on input* 1^λ*, where* λ *is the security parameter, and randomness* $r_g \in \mathcal{R}_{\mathsf{KGen}}$*, outputs a pair of keys* $(\mathsf{pk}, \mathsf{sk})$*.* pk *and* sk *are called the public/encapsulation key and private/decapsulation key, respectively.*
- $\mathsf{Encap}(\mathsf{pk}; r_e) \rightarrow (c, k)$*: an encapsulation algorithm that takes as input encapsulation key* pk*, and randomness* $r_e \in \mathcal{R}_{\mathsf{Encap}}$*, and outputs ciphertext* $c \in \mathcal{C}$ *and encapsulated key* $k \in \mathcal{K}$*.*
- $\mathsf{Decap}(\mathsf{sk}, c) \rightarrow k/\perp$*: a decapsulation algorithm that takes as input decapsulation key* sk *and ciphertext* c *and outputs key* $k \in \mathcal{K}$ *or a rejection symbol* $\perp \notin \mathcal{K}$*.*

Definition 6 (KEM Correctness). *We say that* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *is* δ-correct *if*

$$\Pr[\mathsf{Decap}(\mathsf{sk}, c) \neq k : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda), (c, k) \leftarrow \mathsf{Encap}(\mathsf{pk})] \leq \delta.$$

In particular, we say that KEM *is* perfectly correct *if* $\delta = 0$*.*

```
$\mathbf{Expt}^{\text{IND-CCA}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$(c^*, k_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$k_1^* \leftarrow_\$ \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}(\cdot)}(\mathsf{pk}, c^*, k_b^*)$
return $[b' = b]$
```

```
$\mathbf{Expt}^{\text{SPR-CCA}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$(c_0^*, k_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$(c_1^*, k_1^*) \leftarrow_\$ \mathcal{S}(1^\lambda) \times \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c_b^*}(\cdot)}(\mathsf{pk}, c_b^*, k_b^*)$
return $[b' = b]$
```

```
$\mathbf{Expt}^{\text{SSMT-CCA}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\lambda)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$(c^*, k_0^*) \leftarrow \mathcal{S}(1^\lambda) \times \mathcal{K}$
$k_1^* \leftarrow \mathsf{Decap}(\mathsf{sk}, c^*)$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}(\cdot)}(\mathsf{pk}, c^*, k_b^*)$
return $[b' = b]$
```

```
$\textsc{Decaps}_a(c)$
if $c = a$ then return $\perp$
$k \leftarrow \mathsf{Decap}(\mathsf{sk}, c)$
return $k$
```

```
$\textsc{Decaps}_a(\beta, c)$
if $c = a$ then return $\perp$
$k \leftarrow \mathsf{Decap}(\mathsf{sk}_\beta, c)$
return $k$
```

```
$\mathbf{Expt}^{\text{ANO-CCA}}_{\mathsf{KEM},\mathcal{A}}(\lambda)$
$(\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KGen}(1^\lambda)$
$(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$
$b \leftarrow_\$ \{0,1\}$
$(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$
$b' \leftarrow \mathcal{A}^{\textsc{Decaps}_{c^*}(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1, c^*, k^*)$
return $[b' = b]$
```

Fig. 3. Games for KEM schemes

Definition 7 (KEM Security). *Let* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *be a KEM scheme. For any adversary* $\mathcal{A}$ *and* $\text{GOAL} \in \{\text{IND}, \text{SPR}, \text{ANO}, \text{SSMT}\}$*, we define* $\mathcal{A}$*'s* GOAL-CCA *advantage against* KEM *(w.r.t. a simulator* $\mathcal{S}$ *when* $\text{GOAL} \in \{\text{SPR}, \text{SSMT}\}$*) as follows:*

$$\mathbf{Adv}^{\text{GOAL-CCA}}_{\mathsf{KEM}[,\mathcal{S}]}(\mathcal{A}) := \left| \Pr[\mathbf{Expt}^{\text{GOAL-CCA}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|,$$

where $\mathbf{Expt}^{\text{GOAL-CCA}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\lambda)$ *is an experiment described in Fig. 3. For* $\text{GOAL} \in \{\text{IND}, \text{SPR}, \text{ANO}, \text{SSMT}\}$*, we say* KEM *is* GOAL-CCA-secure *if (there exists a QPT simulator* $\mathcal{S}$ *when* $\text{GOAL} \in \{\text{SPR}, \text{SSMT}\}$ *such that)* $\mathbf{Adv}^{\text{GOAL-CCA}}_{\mathsf{KEM}[,\mathcal{S}]}(\mathcal{A})$ *is negligible for any QPT adversary* $\mathcal{A}$*.*

We also define the above security properties for PKE schemes (in Definition 3) and KEMs (in Definition 7) in the QROM where the corresponding schemes have classical access and the adversary $\mathcal{A}$ has *quantum* access to a random oracle O. Following [27,30], we make the convention that the number q_O of queries made by $\mathcal{A}$ to O counts the total number of times O is executed in the corresponding security game/experiment; i.e., the number of $\mathcal{A}$'s explicit queries to O plus the number of implicit queries to O made by the experiment.

3 Specification of Kyber

As described in [4], Kyber is a KEM whose claimed IND-CCA security relies on hardness of the module learning-with-error problem (MLWE problem [34]).

Kyber–or more formally, Kyber.KEM–is constructed by first starting with a *base* PKE scheme Kyber.PKE and then applying a tweaked Fujisaki-Okamoto (FO) transform to it in order to obtain the final KEM. The tweaked FO transform is described in detail in Figure 4; we also refer the reader to [4, Section 1.2] for a detailed specification of Kyber.PKE.

```
$\mathsf{KGen}'$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
$s \leftarrow_{\$} \{0,1\}^{256}$
$\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$
$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$
return $(\mathsf{pk}, \mathsf{sk}')$
```

```
$\mathsf{Encap}(\mathsf{pk})$
$m \leftarrow_{\$} \{0,1\}^{256}$
$m \leftarrow H(m)$
$h \leftarrow H(pk)$
$(\overline{k}, r) \leftarrow G(m, h)$
$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
$k \leftarrow H'(\overline{k}, H(c))$
return $(c, k)$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $H'(\overline{k}', H(c))$
else return $H'(s, H(c))$
```

Fig. 4. The tweaked FO transform, namely $\mathsf{FO}^{\not\perp'}$ (as described in [26,44]), used in Kyber. Here $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is the base PKE scheme and $(\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ is the final KEM. Also $H, H' \colon \{0,1\}^* \to \{0,1\}^{256}$ and $G \colon \{0,1\}^* \to \{0,1\}^{512}$ are hash functions. Technically, Kyber instantiates H' with the extendable-output function SHAKE-256 which can return outputs of arbitrary length. In this paper, we have H' to only return outputs of bit-length 256 for the sake of simplicity. But our subsequent analysis of Kyber can be modified in a straightforward manner to account for encapsulated keys (derived from H') with arbitrary length.

3.1 Security properties of Kyber.PKE

In our IND-CCA security analysis of Kyber.KEM in Section 4, we rely on the IND-CPA security of Kyber.PKE. Similarly, in our ANO-CCA security analysis (cf. Section 5) of Kyber.KEM and the hybrid PKE schemes derived from it, we rely on the *strong disjoint simulatability* (i.e., *SDS-IND security* plus *statistical disjointness*) [37,36,44] of the base Kyber.PKE scheme.

It was argued in [4, Theorem 1] that (in the (quantum) random oracle model) Kyber.PKE is tightly IND-CPA secure under the MLWE hardness assumption, since under the MLWE assumption, the public-key and ciphertexts of Kyber.PKE are pseudorandom. Hence, we have:

Lemma 5 (informal). Kyber.PKE *is tightly* IND-CPA *secure under the MLWE hardness assumption, in the QROM.*

Regarding the strong disjoint simulatability of Kyber.PKE, we have:

Lemma 6 (informal). $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is tightly* strong disjoint simulatable *under the MLWE hardness assumption, in the QROM.*

Proof (Sketch). Let $\mathcal{S}$ be a QPT simulator algorithm which simply outputs a uniformly random value from the ciphertext space $\mathcal{C}$ of $\mathsf{Kyber.PKE}$. (Note that $\mathcal{C}$ is a set of bit strings with a *fixed pre-specified* length [4, Section 1.2], and hence, is *efficiently samplable.*) The above observation of $\mathsf{Kyber.PKE}$'s public-keys and ciphertexts being pseudorandom under the MLWE assumption can be used in a straightforward manner to show that $\mathsf{Kyber.PKE}$ is tightly SDS-IND secure w.r.t. $\mathcal{S}$ (cf. Definition 4) under the MLWE hardness assumption – as also noted in [4, Section 4.3.2].

Coming to the statistical disjointness of $\mathsf{Kyber.PKE}$ w.r.t. $\mathcal{S}$ (cf. Definition 4), we have $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}} \leq \frac{|\mathsf{Enc}(\mathsf{pk},\mathcal{M})|}{|\mathcal{C}|} \leq \frac{\leq |\mathcal{M}||\mathcal{R}_{\mathsf{Enc}}|}{|\mathcal{C}|}$. Note that across all parameter sets of Kyber [4, Section 1], we have $|\mathcal{C}| \geq 2^{6144}$ and $|\mathcal{M} \times \mathcal{R}_{\mathsf{Enc}}| = 2^{512}$. Hence, for all intents and purposes, $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$ can be considered to be negligible.

Finally, our IND-CCA and ANO-CCA security analyses of $\mathsf{Kyber.KEM}$ accounts for the δ-correctness of $\mathsf{Kyber.PKE}$ (cf. Definition 2). This particular correctness property of the base $\mathsf{Kyber.PKE}$ scheme has been rigorously analyzed in [4, Section 1.4].

4 IND-CCA Security of Kyber in the QROM

In this section, we prove the IND-CCA security of Kyber in the QROM with *concrete* bounds, before proceeding to show the scheme's anonymity (i.e., ANO-CCA security) later in Section 5.

Theorem 1 (IND-CCA security of $\mathsf{Kyber.KEM}$). *Given the base PKE scheme* $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, for any* IND-CCA *adversary $\mathcal{A}$ against* $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ queries to the quantum random oracles G, H, and H', respectively, there exists an* IND-CPA *adversary $\mathcal{B}$ against* $\mathsf{Kyber.PKE}$ *such that*

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{9q_{H'} + 2q_H}{2^{128}}$$

$$+ 4q_G\sqrt{\delta} + \frac{C(q_H+1)^3}{2^{256}},$$

where C (< 648) is the constant from Lemma 1, and the running time of $\mathcal{B}$ is about the same as that of $\mathcal{A}$.

The proof essentially follows the "wrapper-based" approach described in Subsection 1.2 above but with respect to the *implicitly-rejecting* $\mathsf{FO}^{\not\perp}_{m}$ transform. Formal details follow.

Proof. Towards proving the *concrete* IND-CCA security of Kyber in the QROM, we first consider an intermediate $\mathsf{PKE} \rightarrow \mathsf{KEM}$ transform $\mathsf{FO}^{\not\perp'}_{\text{pre}}$, described in Figure 5. Let $\overline{\mathsf{Kyber}}\mathsf{.KEM}$ be the KEM obtained by applying the $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ transform

```
$\mathsf{KGen}'$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$
    $s \leftarrow_{\$} \{0,1\}^{256}$
    $\mathsf{pk}' \leftarrow (\mathsf{pk}, H(\mathsf{pk}))$
    $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}', s)$
    return $(\mathsf{pk}, \mathsf{sk}')$

$\mathsf{Encap}(\mathsf{pk})$
    $m \leftarrow_{\$} \{0,1\}^{256}$
    $h \leftarrow H(\mathsf{pk})$
    $(\overline{k}, r) \leftarrow G(m, h)$
    $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$
    return $(c, \overline{k})$

$\mathsf{Decap}(\mathsf{sk}', c)$
    Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $(\overline{k}', r') \leftarrow G(m', h)$
    $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
    if $c' = c$ then
        return $\overline{k}'$
    else return $H'(s, c)$
```

Fig. 5. The $\mathsf{PKE} \rightarrow \mathsf{KEM}$ transform $\mathsf{FO}^{\not\perp'}_{\mathrm{pre}}$.

on $\mathsf{Kyber.PKE}$, i.e., $\overline{\mathsf{Kyber}}.\mathsf{KEM} = \mathsf{FO}^{\not\perp'}_{\mathrm{pre}}[\mathsf{Kyber.PKE}, G, H, H']$. We now consider the IND-CCA security of $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ in the QROM.

Let $\overline{\mathcal{A}}$ be an IND-CCA adversary against $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ issuing at most q'_D classical queries to the decapsulation oracles, and q'_H and $q'_{H'}$ queries to the quantum random oracles H and H' respectively. Consider the sequence of games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ described in Figure 6 which only differ in the way their corresponding decapsulation oracles $\mathsf{Decap}(\mathsf{sk}', \cdot)$ reject invalid ciphertexts.

```
Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
    $(c^*, \overline{k}^*_0) \leftarrow \mathsf{Encap}(\mathsf{pk})$
    $\overline{k}^*_1 \leftarrow_{\$} \{0,1\}^{256}$
    $b \leftarrow_{\$} \{0,1\}$
    $b' \leftarrow \overline{\mathcal{A}}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, \overline{k}^*_b)$
    return $[b' = b]$

$\mathsf{Decap}(\mathsf{sk}', c)$
    Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
    $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
    $(\overline{k}', r') \leftarrow G(m', h)$
    $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
    if $c' = c$ then
        return $\overline{k}'$
    else return $H'(s, c)$ // $\overline{\mathsf{G}}_0$
    else return $H''(c)$ // $\overline{\mathsf{G}}_1$
    else return $\overline{H}(H(c))$ // $\overline{\mathsf{G}}_2$
```

Fig. 6. Games $\overline{\mathsf{G}}_0$ – $\overline{\mathsf{G}}_2$. Here $H'' \colon \{0,1\}^* \rightarrow \{0,1\}^{256}$ and $\overline{H} \colon \{0,1\}^{256} \rightarrow \{0,1\}^{256}$ are fresh *internal* random oracles, i.e., not directly accessible to $\overline{\mathcal{A}}$.

Game $\overline{\mathsf{G}}_0$: This game is exactly the IND-CCA game for $\overline{\mathsf{Kyber}}.\mathsf{KEM}$. Hence,

$$\left| \Pr[\overline{\mathsf{G}}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}). \tag{1}$$

Game $\overline{\mathsf{G}}_1$: In this game, the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle is modified such that $H''(c)$ is returned instead of $H'(s, c)$ for an invalid ciphertext c, where H'' is a fresh *internal* random oracle not directly accessible to $\overline{\mathcal{A}}$. Using Lemma 2 w.r.t. the pseudorandomness of $H'(s, \cdot)$ during decapsulation, where we have the "PRF key" $s \leftarrow_{\$} \{0,1\}^{256}$, it is not hard to obtain the following via a straightforward reduction:

$$\left|\Pr[\overline{\mathsf{G}}_1 = 1] - \Pr[\overline{\mathsf{G}}_0 = 1]\right| \leq \frac{2q'_{H'}}{2^{128}}. \tag{2}$$

Game $\overline{\mathsf{G}}_2$: In this game, we again modify the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle such that $\overline{H}(H(c))$ is returned instead of $H''(c)$ for an invalid ciphertext c, where $\overline{H}$ is another fresh internal random oracle not directly accessible to $\overline{\mathcal{A}}$. Note that the oracles H'' and $\overline{H}$ are only accessible to $\overline{\mathcal{A}}$ *indirectly* via the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle. Now in the view of adversary $\overline{\mathcal{A}}$, the output distributions of the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle in games $\overline{\mathsf{G}}_1$ and $\overline{\mathsf{G}}_2$ with regards to invalid ciphertexts c are identical *unless* $\overline{\mathcal{A}}$ queries the decapsulations of two invalid ciphertexts c_1 and c_2 such that $H(c_1) = H(c_2)$ (and $c_1 \neq c_2$). Since decapsulation queries are considered to be classical in the QROM, we can bound the probability of such an event by collision-resistance of the QRO H – as described in Lemma 1 – again via a straightforward reduction. Hence, we have[4],

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \Pr[\overline{\mathsf{G}}_1 = 1]\right| \leq \frac{C(q'_H + q'_D + 1)^3}{2^{256}}, \tag{3}$$

where C (< 648) is the constant from Lemma 1.

Hence by collecting the above bounds (1) – (3), we obtain

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right| \leq \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\overline{\mathcal{A}}) + \frac{2q'_{H'}}{2^{128}} + \frac{C(q'_H + q'_D + 1)^3}{2^{256}}, \tag{4}$$

which will be useful shortly when we now focus on proving concrete IND-CCA security of the *actual* scheme of Kyber.

Let $\mathcal{A}$ be an IND-CCA adversary against $\mathsf{Kyber.KEM}$ issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ queries to the quantum random oracles G, H and H' respectively. Consider the sequence of games G_0 – G_8 described in Figure 7.

Game G_0: This game is basically the IND-CCA game for $\mathsf{Kyber.KEM}$ where the adversary $\mathcal{A}$ gets the "real" encapsulated key k^*, i.e., $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$.

Game G_1: Here we essentially do not execute the "$m \leftarrow H(m)$" step during encapsulation (Line 2 in "$\mathsf{Encap}(\mathsf{pk})$", Fig. 4) in this game's setup. We now use the original OW2H lemma (Lemma 3) to bound the difference in $\mathcal{A}$'s "behavior" in games G_0 and G_1. In the context of applying Lemma 3, let $x := m_0^* \leftarrow_{\$} \{0,1\}^{256}$ and $y := m_1^* \leftarrow_{\$} \{0,1\}^{256}$, and consider an oracle algorithm A^H making at-most q_H queries to H such that $A^H(m_0^*, H(m_0^*))$ simulates the game G_0 towards $\mathcal{A}$

[4] Recall from our convention (described in Subsection 2.2) that q'_H counts the total number of times H is invoked in the game $\overline{\mathsf{G}}_0$. However in $\overline{\mathsf{G}}_2$, H is *additionally* invoked when $\overline{\mathcal{A}}$ queries the decapsulation of an invalid ciphertext. Hence, H is queried at most $(q'_H + q'_D)$ many times in $\overline{\mathsf{G}}_2$ in the context of applying Lemma 1.

```
Games $\mathsf{G}_0 - \mathsf{G}_8$

1 :  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
2 :  $m^* \leftarrow_{\$} \{0,1\}^{256}$
3 :  $m^* \leftarrow H(m^*)$ // $\mathsf{G}_0, \mathsf{G}_8$
4 :  $(\overline{k}^*_0, r^*) \leftarrow_{\$} G(m^*, H(\mathsf{pk}))$
5 :  $\overline{k}^*_1 \leftarrow_{\$} \{0,1\}^{256}$
6 :  $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$
7 :  $k^* \leftarrow H'(\overline{k}^*_0, H(c^*))$ // $\mathsf{G}_0$ – $\mathsf{G}_3$
8 :  $k^* \leftarrow H'(\overline{k}^*_1, H(c^*))$ // $\mathsf{G}_4$
9 :  $k^* \leftarrow_{\$} \{0,1\}^{256}$ // $\mathsf{G}_5$ – $\mathsf{G}_8$
10 : $b' \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, k^*)$
11 : return $b'$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$

1 :  Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
2 :  $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :  $(\overline{k}', r') \leftarrow G(m', h)$
4 :  $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
5 :  if $c' = c$ then
6 :    return $H'(\overline{k}', H(c))$
7 :  else
8 :    return $H'(s, H(c))$ // $\mathsf{G}_0$–$\mathsf{G}_1$, $\mathsf{G}_7$–$\mathsf{G}_8$
9 :    return $H''(H(c))$ // $\mathsf{G}_2, \mathsf{G}_6$
10 :   return $H'(\overline{H}(H(c)), H(c))$ // $\mathsf{G}_3$–$\mathsf{G}_5$
```

Fig. 7. Games G_0 – G_8. Here $H'' \colon \{0,1\}^* \to \{0,1\}^{256}$ and $\overline{H} \colon \{0,1\}^{256} \to \{0,1\}^{256}$ are fresh *internal* random oracles, i.e., not directly accessible to $\mathcal{A}$.

and $A^H(m^*_0, m^*_1)$ simulates G_1 towards $\mathcal{A}$. To be more specific, A^H sets "m^*" in Line 4, Fig. 7, to be its second input (either $H(m^*_0)$ or m^*_1) when simulating the appropriate game (G_0 or G_1, respectively) towards $\mathcal{A}$.

Again in the context of Lemma 3, it is not hard to see that $\Pr[\mathsf{G}_0 = 1] = P^1_A$ and $\Pr[\mathsf{G}_1 = 1] = P^2_A$. Regarding the probability P_B, note that during $A^H(m^*_0, m^*_1)$'s simulation of game G_1 towards $\mathcal{A}$, the view of $\mathcal{A}$ is completely independent of the value m^*_0 $(= x) \leftarrow_{\$} \{0,1\}^{256}$. Hence, we have $P_B = \frac{1}{2^{256}}$ which leads to

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{2q_H}{2^{128}} \ (= 2q_H\sqrt{P_B}). \tag{5}$$

Game G_2: In this game, the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle is modified such that $H''(H(c))$ is returned instead of $H'(s, H(c))$ for an invalid ciphertext c, where H'' is a fresh *internal* random oracle not directly accessible to $\mathcal{A}$. Similar to the $\mathsf{G}_0 \to \mathsf{G}_1$ "hop" above, by using Lemma 2 w.r.t. the pseudorandomness of $H'(s, \cdot)$–this time on inputs of the form "$H(c)$"–during decapsulation, it is not hard to obtain:

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \tag{6}$$

Game G_3: In this game, we again modify the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle such that $H'(\overline{H}(H(c)), H(c))$ is returned instead of $H''(H(c))$ for an invalid ciphertext c, where $\overline{H}$ is another fresh internal random oracle not directly accessible to $\mathcal{A}$. Here we use the generalized OW2H lemma (Lemma 4) to bound the difference in $\mathcal{A}$'s behavior in games G_2 and G_3.

In the context of Lemma 4, note that the oracle algorithm needs to distinguish the pair of random functions $(H''(\cdot), H')$ in G_2 from the pair $(H'(\overline{H}(\cdot), \cdot), H')$ in

G_3. But it is not hard to see that this is the same as distinguishing (H'', H') in G_2 from (H'', G') in G_3, where the oracle G' is obtained by *reprogramming* H' on inputs of the form "$(\overline{H}(x), x)$" with $x \in \{0,1\}^{256}$; namely, we have

$$G'(y) = \begin{cases} H''(x) & \text{if } y \text{ is of the form } (\overline{H}(x), x) \text{ with } x \in \{0,1\}^{256} \\ H'(y) & \text{otherwise.} \end{cases}$$

So again in the context of applying Lemma 4, consider an oracle algorithm A which has quantum access to either (H'', H') or (H'', G') such that $A^{H'',H'}$ and $A^{H'',G'}$ simulate G_2 and G_3 respectively towards $\mathcal{A}$, while making $q_{H'}$ oracle queries.[5] Note that the set of differences between the H' and G' oracles is $\mathcal{S} = \{(\overline{H}(x), x) \mid x \in \{0,1\}^{256}\}$. If we then set $\Pr[\mathsf{G}_2 = 1] = P_{\text{left}}$ and $\Pr[\mathsf{G}_3 = 1] = P_{\text{right}}$, from Lemma 4 we have $|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq 2q_{H'}\sqrt{P_{\text{guess}}}$. Regarding P_{guess}, note that during $A^{H'',H'}$'s simulation of G_2 towards the adversary $\mathcal{A}$, the view of $\mathcal{A}$ is completely independent of the (internal) random oracle $\overline{H}$. Hence the probability that measurement of a random H'-oracle query in G_2 will be of the form $(\overline{H}(x), x)$ (with $x \in \{0,1\}^{256}$) is at-most $\frac{1}{2^{256}}$, i.e., $P_{\text{guess}} \leq \frac{1}{2^{256}}$, since $\overline{H}(x)$ will be a fresh uniformly random value in $\{0,1\}^{256}$. Therefore,

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \tag{7}$$

Game G_4: In this game, we generate the encapsulated key k^* in the setup as "$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$" instead of "$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$" where we have $(\overline{k}_0^*, r^*) \leftarrow_\$ G(m^*, H(pk))$ and $\overline{k}_1^* \leftarrow_\$ \{0,1\}^{256}$. Here we make use of our analysis of the $\mathsf{FO}_{\text{pre}}^{\not\perp'}$ transform above.

Consider the game $\overline{\mathsf{G}}_2$ "played" by adversary $\overline{\mathcal{A}}$ in Fig. 6 w.r.t. $\overline{\mathsf{Kyber.KEM}}$. Depending on whether $\overline{\mathcal{A}}$ gets the "real *pre-key*" $\overline{k}_0^*$ or the "random *pre-key*" $\overline{k}_1^*$ from its challenger, it can simulate the game G_3 or G_4 respectively towards $\mathcal{A}$. Namely, $\overline{\mathcal{A}}^{H,H'}(c^*, \overline{k}_b^*)$ computes the encapsulated key k^* as $k^* \leftarrow H'(\overline{k}_b^*, H(c^*))$ (where b is the bit sampled by $\overline{\mathcal{A}}$'s challenger in Fig. 6) and sends it to $\mathcal{A}$ during the games' setup. $\overline{\mathcal{A}}^{H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}$ also simulates the decapsulation oracle in games G_3 and G_4 (cf. Fig. 7) as follows: given a decapsulation query c from $\mathcal{A}$, $\overline{\mathcal{A}}$ queries its *own* $\mathsf{Decap}(\mathsf{sk}',\cdot)$ oracle in $\overline{\mathsf{G}}_2$ on c to obtain a key $\overline{k}'$–which can also be the value "$\overline{H}(H(c))$" if c is invalid (cf. Line 9 in "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 6)–and returns $H'(\overline{k}', H(c))$ to $\mathcal{A}$. Hence, it is not hard to see from this reduction that

$$\begin{aligned} |\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| &= \left|\Pr[1 \leftarrow \overline{\mathcal{A}} \mid b = 1] - \Pr[1 \leftarrow \overline{\mathcal{A}} \mid b = 0]\right| \\ &= 2 \cdot \left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right|. \end{aligned}$$

[5] For example, A uses the first oracle H'' to simulate $\mathsf{Decap}(\mathsf{sk}',\cdot)$ in Figure 7 w.r.t. invalid ciphertexts c; given such a decapsulation query c from $\mathcal{A}$, the algorithm A returns $H''(H(c))$, where the oracle H is sampled independently by A at the games' setup.

By using Inequality (4) above w.r.t. our analysis of $\overline{\mathsf{Kyber.KEM}}$, we obtain[6]

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq 2\mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber.KEM}}}(\overline{\mathcal{A}}) + \frac{4q_{H'}}{2^{128}} + \frac{2C(q_H + 1)^3}{2^{256}}. \quad (8)$$

Game G_5: Here we have the encapsulated key k^* in the setup to be an independent and uniformly random value, i.e., "$k^* \leftarrow_{\$} \{0,1\}^{256}$", instead of deriving it from H' as "$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$". Similar to the $\overline{\mathsf{G}}_0 \to \overline{\mathsf{G}}_1$ hop above, by using Lemma 2 w.r.t. the pseudorandomness of $H'(\overline{k}_1^*, \cdot)$–with "PRF key" $\overline{k}_1^* \leftarrow_{\$} \{0,1\}^{256}$–during setup, it is not hard to obtain:

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \quad (9)$$

Game G_6: In this game, we modify the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle such that $H''(H(c))$ is returned instead of $H'(\overline{H}(H(c)), H(c))$ for an invalid ciphertext c. In essence, we are reverting the changes introduced in the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop. Hence, by applying a similar reasoning as that hop, we get

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \quad (10)$$

Game G_7: In this game, $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle is modified such that $H'(s, H(c))$ is returned instead of $H''(H(c))$ for an invalid ciphertext c. Again in essence, we are reverting the changes introduced in the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop. Hence, by using a similar reasoning as that hop–namely, pseudorandomness of the oracle $H'(s, \cdot)$ on inputs of the form "$H(c)$"–we obtain

$$|\Pr[\mathsf{G}_7 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \quad (11)$$

Game G_8: Here we re-introduce the "$m \leftarrow H(m)$" step during encapsulation (Line 2 in "$\mathsf{Encap}(\mathsf{pk})$", Fig. 4) in this game's setup, thereby reverting the changes introduced in the "$\mathsf{G}_0 \to \mathsf{G}_1$" hop. By applying Lemma 3 in a similar way as that hop, we get

$$|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{2q_H}{2^{128}}. \quad (12)$$

Now note that G_8 is the IND-CCA game for $\mathsf{Kyber.KEM}$ where the adversary $\mathcal{A}$ gets a "random" encapsulated key k^*, i.e., $k^* \leftarrow_{\$} \{0,1\}^{256}$ (in contrast to getting the "real" encapsulated key in G_0). Hence, we have

$$2 \cdot \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) = |\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_0 = 1]|\,.$$

[6] Here we replace the term "$q'_H + q'_D$" in Inequality (4) with "q_H". Recall from Footnote 3 that $(q'_H + q'_D)$ is the maximum number of times oracle H is queried in $\overline{\mathsf{G}}_2$. But since the decapsulation algorithm of $\mathsf{Kyber.KEM}$ involves a single invocation of $H(\cdot)$ for each input ciphertext c (see "$\mathsf{Decap}(\mathsf{sk}', c)$", Fig. 4), the quantity "q_H" *includes* the number of times H is queried by $\overline{\mathcal{A}}$ to answer decapsulation queries from $\mathcal{A}$ – following our convention w.r.t. counting the number of random oracle queries in security games (cf. Subsection 2.2).

By collecting the above bounds (5) - (12), we obtain

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}) + \frac{7q_{H'} + 2q_H}{2^{128}} + \frac{C(q_H + 1)^3}{2^{256}}. \tag{13}$$

Coming to the term "$\mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}})$", note that the $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ transform is essentially identical to the $\mathsf{FO}^{\not\perp}_{m}$ transform of [27] (also described in Fig. 1) in the context of proving IND-CCA security of the obtained KEM. That is, the existing IND-CCA security theorems w.r.t. $\mathsf{FO}^{\not\perp}_{m}$ in the QROM derived in the literature (e.g., in [30,37,10,33]) apply to $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ *as-it-is* because of the following reasons:

- Note that $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ uses a single hash function G to compute both the encapsulated key $\overline{k}$ and the random coins r for the deterministic encryption of m during encapsulation, whereas $\mathsf{FO}^{\not\perp}_{m}$ uses two separate hash functions for the same. However, these two computations are equivalent when the corresponding hash functions are modeled as independent random oracles with appropriate output lengths.
- Similarly, $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ uses the hash $H(\mathsf{pk})$ to compute $\overline{k}$ and r during encapsulation (and $H(\mathsf{pk})$ is also included in the KEM's secret key sk'), in contrast to $\mathsf{FO}^{\not\perp}_{m}$. But this change preserves the relevant IND-CCA theorems from $\mathsf{FO}^{\not\perp}_{m}$ to $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ with trivial changes to the corresponding proofs, to accommodate the inclusion of $H(\mathsf{pk})$, because the IND-CCA security notion only involves a *single* user's public-key pk (as opposed to multi-user security notions, such as ANO-CCA which involves *two* public-keys).

Hence, by applying [30, Theorem 2][7] regarding the IND-CCA security of "$\mathsf{FO}^{\not\perp}_{m}$-derived" KEMs in the QROM to $\overline{\mathsf{Kyber}}.\mathsf{KEM}$, we have that there exists an IND-CPA adversary $\mathcal{B}$ against $\mathsf{Kyber.PKE}$, with its running time about the same as that of $\overline{\mathcal{A}}$ (and hence, that of $\mathcal{A}$ as well), such that[8]

$$\mathbf{Adv}^{\text{IND-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM}}(\overline{\mathcal{A}}) \leq 2(q_G + q_{H'})\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q_{H'}}{2^{128}} + 4q_G\sqrt{\delta}. \tag{14}$$

Combining the inequalities (13) and (14) finishes the proof.

[7] As mentioned in Subsection 1.2, the reason we are not applying the *tighter* QROM IND-CCA security theorems of [10,33] w.r.t. $\mathsf{FO}^{\not\perp}_{m}$-derived KEMs is that they make an additional assumption on the base PKE scheme being *injective* [10]. However, we leave a detailed analysis of $\mathsf{Kyber.PKE}$'s injectivity as an open question.

[8] Technically, [30, Theorem 2] reduces the IND-CCA security of the KEM to the OW-CPA security of the underlying PKE scheme. But it is well-known that IND-CPA security of a PKE scheme with a sufficiently large message space also implies its OW-CPA security; namely, for any OW-CPA adversary $\mathcal{B}_{\text{ow}}$ against a PKE scheme PKE with message space $\mathcal{M}$, there exists an IND-CPA adversary $\mathcal{B}_{\text{ind}}$ against PKE with the same running time as that of $\mathcal{B}_{\text{ow}}$ such that $\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\text{ow}}) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\text{ind}}) + \frac{1}{|\mathcal{M}|}$.

Remark 1. An alternative approach to prove IND-CCA security of Kyber in the QROM was suggested in [16], involving the *compressed oracle* technique introduced in [46]. More specifically, given two random oracles $H_1 : \{0,1\}^m \to \{0,1\}^n$, $H_2 : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^n$, and a polynomial-sized stateless classical circuit C which has quantum access to H_1, H_2, it was shown in [46, Section 5] that the "domain extender" $C^{H_1,H_2}(x,y) = H_2(H_1(x), y)$ is *indifferentiable* from a quantum random oracle $H : \{0,1\}^{m+\ell} \to \{0,1\}^n$. Informally, indifferentiability guarantees that any efficient adversary cannot distinguish $\langle (H_1, H_2), C^{H_1,H_2} \rangle$ from $\langle \mathcal{S}^H, H \rangle$ where the simulator $\mathcal{S}$ queries H and simulates the oracles H_1, H_2.

Now note that in Kyber (Fig. 4, Line. 6 of "$\mathsf{Encap}(\mathsf{pk})$"), the encapsulated keys are generated as "$k \leftarrow H'(\overline{k}, H(c))$" by hashing the "*pre-key*" $\overline{k}$ and a "nested hash" of the ciphertext, i.e., $H(c)$. And as noted in [26,44], this nested hash $H(c)$ creates problems when extending prior QROM security analysis of (implicitly-rejecting) FO transforms in the literature to Kyber. However, since [46, Section 5] essentially shows that $H'(\overline{k}, H(c))$ is indifferentiable from $H''(\overline{k}, c)$, for a fresh random oracle H'', we can "ignore" the nested hash $H(c)$ in our analysis of Kyber; in fact, [26, Appendix E] already proved the IND-CCA security of a variant of the FO transform where keys are derived as "$k \leftarrow H''(\overline{k}, c)$". However, we make a couple of remarks regarding this matter:

- At a conceptual level, our IND-CCA security analysis of Kyber above (Theorem 1) relies on arguably simpler proof techniques than the ones introduced in [46]. Specifically, our analysis of Kyber in the QROM is based on that of the $\mathsf{FO}_m^{\not\perp}$ transform in the literature, which in turn is based on the well-known "One-Way To Hiding (OW2H) lemma" [42,3] proof technique. And as mentioned in Section 1, [43] provided a framework for *formally* verifying security proofs that involve applications of the OW2H lemma in the QROM. Hence, this should make our security proofs for Kyber amenable to formal verification, thereby providing further confidence in our analysis of the new NIST PQC standard.
- Quantitatively, if we rely on the above indifferentiability argument to analyze Kyber instead, then when switching from "$H'(\overline{k}, H(c))$" to "$H''(\overline{k}, c)$" we would incur an additive "indifferentiability" term $O(q^2/2^{n/2})$ (as specified in [46, Section 5]) in our IND-CCA security bounds, where q is the number of adversarial quantum random oracle queries made to H, H', and $n = 256$ for Kyber. In contrast, our concrete bounds in Theorem 1 includes an additive "collision-resistance (of H)" term $O(q^3/2^n)$. Hence, our concrete IND-CCA security theorem for Kyber allows for strictly more number of random oracle queries q when compared to the indifferentiability-based argument, especially w.r.t. higher security level parameter sets for Kyber when the "correctness" term $O(q\sqrt{\delta})$ is no longer a limiting factor on q (e.g., $\delta = 2^{-164}, 2^{-174}$).

  At the same time, there does not seem to be a straightforward *matching* attack on the IND-CCA security of Kyber that exploits finding collisions in H. Hence, we leave it as an open question to provide a concrete proof of IND-CCA security for Kyber in the QROM which does not rely on the collision-resistance of quantum random oracles, while ensuring tightness w.r.t. the passive IND-CPA security of the base PKE scheme as in the case with implicitly-rejecting FO transforms.

5 ANO-CCA Security of Kyber in the QROM

In this section, we prove the concrete ANO-CCA security of Kyber, and the hybrid PKE schemes derived from it, in the QROM. As mentioned in Subsection 1.2 above, we first prove that the aforementioned schemes are *strongly pseudorandom* (or, *SPR-CCA secure*; cf. Definitions 3, 7) in the QROM, which in turn implies their ANO-CCA security [44, Thm. 2.5 of ePrint version].

5.1 SPR-CCA Security of Kyber.KEM

Here we prove the concrete SPR-CCA security of Kyber.KEM in the QROM while relying on the *strong disjoint simulatability* (i.e., *SDS-IND security* and *statistical disjointness*; cf. Lemma 6) of the base Kyber.PKE scheme.

Theorem 2 (SPR-CCA security of Kyber.KEM). *Let the base PKE scheme* Kyber.PKE = (KGen, Enc, Dec) *be δ-correct, and $\mathcal{S}$ be a QPT simulator algorithm which simply outputs a uniformly random value from the ciphertext space of* Kyber.PKE*. Then for any* SPR-CCA *adversary $\mathcal{A}$ against* Kyber.KEM = (KGen′, Encap, Decap) *w.r.t. $\mathcal{S}$ issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ queries to the quantum random oracles G, H and H' respectively, there exists an* IND-CPA *adversary $\mathcal{B}$ and a* SDS-IND *adversary $\mathcal{D}$ against* Kyber.PKE *w.r.t. $\mathcal{S}$ such that*

$$\begin{aligned}\mathbf{Adv}^{\text{SPR-CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) \leq q_G\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{1}{2}\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda) \\ + \mathbf{Adv}^{\text{SDS-IND}}_{\mathsf{Kyber.PKE},\mathcal{S}}(\mathcal{D}) + (2 + 8(q_G + q_D + 2)^2 + 8(2q_G + 2)^2)\delta \\ + \frac{2(q_{H'} + q_D)}{2^{128}} + \frac{C(q_H + 1)^3}{2^{256}} + \frac{q_H + 7q_{H'}}{2^{128}},\end{aligned}$$

where C (< 648) is the constant from Lemma 1, and the running time of $\mathcal{B}$ and $\mathcal{D}$ is about the same as that of $\mathcal{A}$.

The proof follows quite closely to that of IND-CCA security of Kyber.KEM in the QROM above (Theorem 1). We will be focusing on the main differences in our SPR-CCA security analysis below.

Proof. Same as in our proof of IND-CCA security for Kyber.KEM (Theorem 1), we first consider SPR-CCA security of the "intermediate" scheme $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ = $\mathsf{FO}^{\not\perp'}_{\mathrm{pre}}[\mathsf{Kyber.PKE}, G, H, H']$ (see Fig. 5) in the QROM.

Let $\overline{\mathcal{A}}$ be an SPR-CCA adversary against $\overline{\mathsf{Kyber}}.\mathsf{KEM}$ w.r.t. simulator $\mathcal{S}$ (described above) issuing at most q'_D classical queries to the decapsulation oracles, and q'_H and $q'_{H'}$ queries to the quantum random oracles H and H' respectively.

```
Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
$(c_0^*, \overline{k}_0^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$
$c_1^* \leftarrow \mathcal{S}()$
$\overline{k}_1^* \leftarrow_\$ \{0,1\}^{256}$
$b \leftarrow_\$ \{0,1\}$
$b' \leftarrow \overline{\mathcal{A}}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c_b^*, \overline{k}_b^*)$
return $[b' = b]$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
  return $\overline{k}'$
else return $H'(s, c)$ // $\overline{\mathsf{G}}_0$
else return $H''(c)$ // $\overline{\mathsf{G}}_1$
else return $\overline{H}(H(c))$ // $\overline{\mathsf{G}}_2$
```

Fig. 8. Games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$. Here $H'' \colon \{0,1\}^* \to \{0,1\}^{256}$ and $\overline{H} \colon \{0,1\}^{256} \to \{0,1\}^{256}$ are fresh *internal* random oracles, i.e., not directly accessible to $\overline{\mathcal{A}}$. Also, $\mathcal{S}$ is the simulator described above which simply outputs a uniformly random Kyber.PKE ciphertext.

Consider the sequence of games $\overline{\mathsf{G}}_0 - \overline{\mathsf{G}}_2$ described in Figure 8. It is straightforward to obtain the following based on our IND-CCA security analysis of $\overline{\mathsf{Kyber}}$.KEM (Inequality (4)) in the proof of Theorem 1 above.

$$\left|\Pr[\overline{\mathsf{G}}_2 = 1] - \frac{1}{2}\right| \leq \mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{2q'_{H'}}{2^{128}} + \frac{C(q'_H + q'_D + 1)^3}{2^{256}}, \qquad (15)$$

Now, we return to proving SPR-CCA security of the *actual* Kyber.KEM. Let $\mathcal{A}$ be an SPR-CCA adversary against Kyber.KEM w.r.t. $\mathcal{S}$ issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ queries to the quantum random oracles G, H and H' respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_7$ described in Figure 9. These games are quite similar to the ones described in Figure 7 in our IND-CCA security proof.

Game G_0: This game is the SPR-CCA game for Kyber.KEM with the "real" ciphertext c^* and "real" encapsulated key k^* where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$.

Now note that the games $\mathsf{G}_0 - \mathsf{G}_3$ in Figure 9 are essentially *identical* to the games "$\mathsf{G}_0 - \mathsf{G}_3$" defined in Figure 7. Hence, from our analysis of these game hops (i.e., Inequalities (5) – (7)) in the above IND-CCA security proof, it is not hard to obtain:

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{2q_H}{2^{128}} + \frac{4q_{H'}}{2^{128}}. \qquad (16)$$

Game G_4: Relative to G_3 (and G_0), we modify how the challenge ciphertext c^* and corresponding encapsulated key k^* are generated. In this game, we generate (c^*, k^*) as $c^* \leftarrow \mathcal{S}()$ and $k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ instead, where $\mathcal{S}$ is the simulator described above and $\overline{k}_1^* \leftarrow_\$ \{0,1\}^{256}$. Here we use our SPR-CCA security analysis of the intermediate $\overline{\mathsf{Kyber}}$.KEM.

To be specific, recall that in the corresponding "$\mathsf{G}_3 \to \mathsf{G}_4$" hop (Inequality (8)) in our above IND-CCA security proof of Kyber.KEM, we showed a reduc-

```
Games $\mathsf{G}_0$ – $\mathsf{G}_7$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
$m^* \leftarrow_{\$} \{0,1\}^{256}$
$m^* \leftarrow H(m^*)$ // $\mathsf{G}_0$
$(\overline{k}_0^*, r^*) \leftarrow_{\$} G(m^*, H(\mathsf{pk}))$
$\overline{k}_1^* \leftarrow_{\$} \{0,1\}^{256}$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ // $\mathsf{G}_0$ – $\mathsf{G}_3$
$c^* \leftarrow \mathcal{S}()$ // $\mathsf{G}_4$ – $\mathsf{G}_7$
$k^* \leftarrow H'(\overline{k}_0^*, H(c^*))$ // $\mathsf{G}_0$ – $\mathsf{G}_3$
$k^* \leftarrow H'(\overline{k}_1^*, H(c^*))$ // $\mathsf{G}_4$
$k^* \leftarrow_{\$} \{0,1\}^{256}$ // $\mathsf{G}_5$ – $\mathsf{G}_7$
$b' \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, k^*)$
return $b'$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $H'(\overline{k}', H(c))$
else
    return $H'(s, H(c))$ // $\mathsf{G}_0$ – $\mathsf{G}_1$, $\mathsf{G}_7$
    return $H''(H(c))$ // $\mathsf{G}_2, \mathsf{G}_6$
    return $H'(\overline{H}(H(c)), H(c))$ // $\mathsf{G}_3$–$\mathsf{G}_5$
```

Fig. 9. Games G_0 – G_7. Here $H''\colon \{0,1\}^* \to \{0,1\}^{256}$ and $\overline{H}\colon \{0,1\}^{256} \to \{0,1\}^{256}$ are fresh *internal* random oracles, i.e., not directly accessible to $\mathcal{A}$.

tion to IND-CCA security of the underlying $\overline{\mathsf{Kyber.KEM}}$. In a similar way, it is straightforward to construct an SPR-CCA adversary $\overline{\mathcal{A}}$ against $\overline{\mathsf{Kyber.KEM}}$ w.r.t. the same $\mathcal{S}$ above such that

$$\begin{aligned}|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]| &= 2 \cdot |\Pr[\overline{\mathsf{G}}_2 = 1] - 1/2| \\ &\leq 2\mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber.KEM}},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{4q_{H'}}{2^{128}} + \frac{2C(q_H+1)^3}{2^{256}}, \quad (17)\end{aligned}$$

where we used Inequality (15) w.r.t. our analysis of $\overline{\mathsf{Kyber.KEM}}$.

Game G_5: We further modify how k^* is generated. In this game, k^* is chosen from $\{0,1\}^{256}$ uniformly at random. Similar to our analysis of the "$\mathsf{G}_4 \to \mathsf{G}_5$" hop (Inequality(9)) in the proof of Theorem 1, we obtain the following by applying Lemma 2.

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \qquad (18)$$

Game G_6: We modify the decapsulation oracle such that the oracle rejects an invalid ciphertext c by returning $H''(H(c))$. In a sense, we are reverting the changes introduced in the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop above (cf. Inequality (7) in the proof of Theorem 1). Hence, it is not hard to obtain

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \qquad (19)$$

Game G_7: We again modify the decapsulation oracle such that the oracle returns $H'(s, H(c))$ for an invalid ciphertext c. From our analysis of the "$\mathsf{G}_1 \to$

G_2" hop above (cf. Inequality (6) in the proof of Theorem 1), we have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \tag{20}$$

Note that G_7 is the SPR-CCA game for $\mathsf{Kyber.KEM}$ where $\mathcal{A}$ gets a "random" ciphertext $c^* \leftarrow \mathcal{S}()$ and "random" encapsulated key $k^* \leftarrow_{\$} \{0,1\}^{256}$. Hence, by summing up the bounds (16) - (20), we obtain

$$\begin{aligned} 2\mathbf{Adv}^{\text{SPR-CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) &= |\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_7 = 1]| \\ &\leq 2\mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}}) + \frac{2C(q_H+1)^3}{2^{256}} + \frac{2q_H + 14q_{H'}}{2^{128}}. \end{aligned} \tag{21}$$

Finally, we replace the term "$\mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}})$" with the existing SPR-CCA security bounds on the $\mathsf{FO}^{\not\perp}_m$ transform in the QROM derived in [44]. Because as previously noted in our proof of Theorem 1 above, the intermediate $\mathsf{FO}^{\not\perp'}_{\text{pre}}$ transform is essentially identical to $\mathsf{FO}^{\not\perp}_m$ in the context of "single key-pair notions" such as IND-CCA security *and* SPR-CCA security. Hence, by applying [44, Thms. D.1 and 4.1 of ePrint][9] w.r.t. the SPR-CCA security of "$\mathsf{FO}^{\not\perp}_m$-derived" KEMs in the QROM to $\overline{\mathsf{Kyber.KEM}}$, we have that there exists an IND-CPA adversary $\mathcal{B}$ and a SDS-IND adversary $\mathcal{D}$ w.r.t. $\mathcal{S}$ against $\mathsf{Kyber.PKE}$, running in about the same time as that of $\overline{\mathcal{A}}$ (and $\mathcal{A}$), such that[10]

$$\begin{aligned} \mathbf{Adv}^{\text{SPR-CCA}}_{\overline{\mathsf{Kyber}}.\mathsf{KEM},\mathcal{S}}(\overline{\mathcal{A}}) &\leq q_G \sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{Kyber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{1}{2}\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda) \\ &+ \mathbf{Adv}^{\text{SDS-IND}}_{\mathsf{Kyber.PKE},\mathcal{S}}(\mathcal{D}) + \frac{2(q_{H'} + q_D)}{2^{128}} + (2 + 8(q_G + q_D + 2)^2 + 8(2q_G + 2)^2)\delta. \end{aligned} \tag{22}$$

Combining inequalities (21) and (22) finishes the proof.

Corollary 1 (ANO-CCA security of $\mathsf{Kyber.KEM}$). *Given* $\mathsf{Kyber.PKE}$ *is* IND-CPA *secure and* strongly disjoint-simulatable*, then* $\mathsf{Kyber.KEM}$ *is* ANO-CCA *secure in the QROM.*

This follows from [44, Thm. 2.5 of ePrint] which states that the SPR-CCA security of a KEM implies its ANO-CCA security.

[9] $\mathsf{FO}^{\not\perp}_m$ is composed of two *modular* FO transforms: namely, the "T" and "$\mathsf{U}^{\not\perp}_m$" transforms defined in [27]; [44, Thm. D.1 of ePrint] considers the T transform and [44, Thm. 4.1 of ePrint] considers the $\mathsf{U}^{\not\perp}_m$ transform respectively.

[10] Technically, [44, Thm. 4.1 of ePrint] includes statistical disjointness (cf. Definition 4) of a *derandomized* version of the base PKE scheme in its SPR-CCA security bounds on the final KEM. Roughly speaking, in such a derandomized PKE, the random coins used to encrypt a message m is obtained by first hashing m. But from our proof sketch of Lemma 6, it is not hard to see that statistical disjointness of the derandomized $\mathsf{Kyber.PKE}$ is trivially upper-bounded by disjointness of the *original* $\mathsf{Kyber.PKE}$, i.e., $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}$. This is because our simulator $\mathcal{S}$ just outputs a uniformly random $\mathsf{Kyber.PKE}$ ciphertext.

5.2 SPR-CCA Security of Hybrid PKE Derived from $\mathsf{Kyber.KEM}$

We now focus on anonymity, or more specifically, SPR-CCA security of hybrid PKE schemes obtained from $\mathsf{Kyber.KEM}$ via the well-known "KEM-DEM" framework of [15]. It was shown in [44, Thm. 3.2 of ePrint] that composing a *one-time strongly pseudorandom* (or, *SPR-otCCA secure*; cf. Definition 10 in Appendix A.1) DEM with an implicitly-rejecting KEM which is both SPR-CCA secure *and strongly smooth* (or, *SSMT-CCA secure*; cf. Definition 7) results in an SPR-CCA secure hybrid PKE scheme. Hence, we establish concrete SSMT-CCA security of $\mathsf{Kyber.KEM}$ in the QROM below while relying on statistical disjointness of the base $\mathsf{Kyber.PKE}$ scheme.

Theorem 3 (SSMT-CCA security of $\mathsf{Kyber.KEM}$). *Let $\mathcal{S}$ be a QPT simulator which outputs a uniformly random value from the ciphertext space of* $\mathsf{Kyber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$. *For any* SSMT-CCA *adversary $\mathcal{A}$ against the scheme* $\mathsf{Kyber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *w.r.t. $\mathcal{S}$ issuing at most q_D classical queries to the decapsulation oracles, and at most q_G, q_H and $q_{H'}$ queries to the quantum random oracles G, H and H' respectively, we have*

$$\mathbf{Adv}^{\text{SSMT-CCA}}_{\mathsf{Kyber.KEM},\mathcal{S}}(\mathcal{A}) \leq \mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda) + \frac{2q_{H'}+1}{2^{128}} + \frac{C(q_H+1)^3}{2 \cdot 2^{256}},$$

where C (< 648) is the constant from Lemma 1.

```
Games $\mathsf{G}_0$ – $\mathsf{G}_6$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
$c^* \leftarrow \mathcal{S}()$ // $\mathsf{G}_0$, $\mathsf{G}_6$
$c^* \leftarrow \mathcal{S}() \setminus \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$ // $\mathsf{G}_1$ – $\mathsf{G}_5$
$k^* \leftarrow_\$ \{0,1\}^{256}$ // $\mathsf{G}_0$ – $\mathsf{G}_2$
$k^* \leftarrow H''(H(c^*))$ // $\mathsf{G}_3$
$k^* \leftarrow H'(s, H(c^*))$ // $\mathsf{G}_4$
$k^* \leftarrow \mathsf{Decap}(\mathsf{sk}', c^*)$ // $\mathsf{G}_5$ – $\mathsf{G}_6$
$b' \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}',\cdot)}(\mathsf{pk}, c^*, k^*)$
return $b'$
```

```
$\mathsf{Decap}(\mathsf{sk}', c)$
Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, h, s)$
if $c = c^*$ then return $\bot$
$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$(\overline{k}', r') \leftarrow G(m', h)$
$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
if $c' = c$ then
    return $H'(\overline{k}', H(c))$
else
    return $H'(s, H(c))$ // $\mathsf{G}_0$–$\mathsf{G}_1$, $\mathsf{G}_4$–$\mathsf{G}_6$
    return $H''(H(c))$ // $\mathsf{G}_2$ – $\mathsf{G}_3$
```

Fig. 10. Games G_0 – G_6. Here $H'' \colon \{0,1\}^* \to \{0,1\}^{256}$ is a fresh *internal* random oracle not directly accessible to $\mathcal{A}$. Also, $\mathcal{S}$ is the simulator described above which simply outputs a uniformly random $\mathsf{Kyber.PKE}$ ciphertext.

Proof. **Game G_0:** This game is the SSMT-CCA game for $\mathsf{Kyber.KEM}$ with the random encapsulated key $k^* \leftarrow_\$ \{0,1\}^{256}$ and simulated ciphertext $c^* \leftarrow \mathcal{S}()$.

Game G_1: We then modify how c^* is generated. In this game, c^* is generated by $\mathcal{S}()$ conditioned on that c^* is outside of $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$. More specifically, the game does a (potentially inefficient) check on whether $c^* \in \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$ and aborts if it is the case. Note that this potential inefficiency does not really matter in our analysis since we will be bounding the difference between subsequent games using *statistical* bounds anyway.

Coming to the difference between games G_0 and G_1, it is bounded by the value $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda)$, and we have

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda). \tag{23}$$

Game G_2: We next modify the "implicit rejection" of the decapsulation oracle. In this game, the oracle rejects by outputting $H''(H(c))$ instead of $H'(s, H(c))$, where H'' is an independent random oracle. From the "$\mathsf{G}_1 \rightarrow \mathsf{G}_2$" hop (Inequality (6)) in the proof of Theorem 1 above, we obtain the following via Lemma 2:

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_{H'}}{2^{128}}. \tag{24}$$

Game G_3: We next modify how k^* is generated. In this game, k^* is computed as $H''(H(c^*))$ instead of being chosen uniformly at random.

Notice that the adversary can only access H'' via the decapsulation oracle. Thus, if the adversary cannot query $c \neq c^*$ such that $H(c) = H(c^*)$, then the adversary cannot obtain any information on $H''(H(c^*))$ and this value looks completely random. Similar to the "$\overline{\mathsf{G}}_1 \rightarrow \overline{\mathsf{G}}_2$" hop (Inequality (3)) above in our IND-CCA security proof of $\mathsf{Kyber.KEM}$, we can bound the difference between G_2 and G_3 via a straightforward reduction to the collision resistance of H. Hence, we have from Lemma 1

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{C(q_H + 1)^3}{2^{256}}. \tag{25}$$

Game G_4: We next replace all invocations of $H''(H(\cdot))$ in this game – particularly, during generation of k^* and decapsulation of ciphertexts – with $H'(s, H(\cdot))$. Again from the "$\mathsf{G}_1 \rightarrow \mathsf{G}_2$" hop above (Inequality 24), we can use the pseudorandomness of H' (Lemma 2) to obtain

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2(q_{H'} + 1)}{2^{128}}. \tag{26}$$

Game G_5: In this game, we compute k^* as $k^* \leftarrow \mathsf{Decap}(\mathsf{sk}', c^*)$ instead of $k^* \leftarrow H'(s, H(c^*))$. Anyways the result of $\mathsf{Decap}(\mathsf{sk}', c^*)$ in G_5 will be equal to $H'(s, H(c^*))$ as in G_4. Because note that c^* is an invalid ciphertext since it is outside of $\mathsf{Enc}(\mathsf{pk}, \mathcal{M})$. Thus, even if the decryption of c^* yields some plaintext m', the re-encrypted ciphertext $c' = \mathsf{Enc}(\mathsf{pk}, m'; r')$ cannot be equivalent to c^*. Hence, we have $\Pr[\mathsf{G}_4 = 1] = \Pr[\mathsf{G}_5 = 1]$.

Game G_6: We finally modify how c^* is generated. In this game, c^* is generated by $\mathcal{S}()$ (and there is no check by the game on whether $c^* \in \mathsf{Enc}(\mathsf{pk}, \mathcal{M})$).

We note that this game is the SSMT-CCA game for $\mathsf{Kyber.KEM}$ with simulated ciphertext $c^* \leftarrow \mathcal{S}()$ and decapsulated key $k^* \leftarrow \mathsf{Decap}(\mathsf{sk}, c^*)$.

The difference is again bounded by $\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda)$, and we have

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]| \leq \mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda). \tag{27}$$

Summing up the above differences (23) - (27), we have

$$\begin{aligned} 2\mathbf{Adv}^{\text{SSMT-CCA}}_{\mathsf{Kyber.KEM}}(\mathcal{A}) &= |\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_6 = 1]| \\ &\leq 2\mathsf{Disj}_{\mathsf{Kyber.PKE},\mathcal{S}}(\lambda) + \frac{4q_{H'} + 2}{2^{128}} + \frac{C(q_H + 1)^3}{2^{256}}. \end{aligned}$$

Corollary 2 (ANO-CCA security of hybrid PKE from Kyber.KEM). *Given* Kyber.KEM *is* SPR-CCA *secure,* SSMT-CCA *secure, and δ-correct, and a* DEM *that is* SPR-otCCA secure*, then the hybrid PKE scheme obtained by composing* Kyber.KEM *and* DEM *is* SPR-CCA *secure, and hence,* ANO-CCA *secure.*

This follows from [44, Thm. 3.2 of ePrint].

Robustness of Kyber. The notion of "*robustness*" for PKE was defined in [1], and there it was argued that robustness is an essential conjunct of anonymous encryption. Roughly speaking, robustness guarantees that it is hard to produce a ciphertext which decrypts validly under two different private keys. Fortunately, it was shown in [26] that composing Kyber.KEM with an appropriately "robust" DEM (as defined in [21]) will result in a robust hybrid PKE scheme. In other words, composing Kyber with a one-time strongly pseudorandom and robust DEM will result in a post-quantum strongly anonymous *and* robust PKE scheme.

References

1. M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *TCC 2010*, pages 480–497, 2010.
2. G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, and D. Smith-Tone. Status report on the third round of the nist post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2022.
3. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019, Part II*, pages 269–295, 2019.
4. R. Avanzi, J. Bos, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber: NIST Round 3 Submission, Algorithm Specifications And Supporting Documentation (v3.02), 2021.
5. A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In *FC 2006*, pages 52–64, 2006.
6. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT 2001*, pages 566–582, 2001.
7. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS 93*, pages 62–73, 1993.

8. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, 2014.
9. D. J. Bernstein, B. B. Brumley, M.-S. Chen, C. Chuengsatiansup, T. Lange, A. Marotzke, B.-Y. Peng, N. Tuveri, C. van Vredendaal, and B.-Y. Yang. NTRU Prime. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.
10. N. Bindel, M. Hamburg, K. Hövelmanns, A. Hülsing, and E. Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *TCC 2019, Part II*, pages 61–90, 2019.
11. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, 2011.
12. C. Boyd, Y. Cliff, J. M. G. Nieto, and K. G. Paterson. One-round key exchange in the standard model. *Int. J. Appl. Cryptogr.*, 1(3):181–199, 2009.
13. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS 2012*, pages 309–325, 2012.
14. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, pages 93–118, 2001.
15. R. Cramer and V. 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.
16. Daniel J. Bernstein. Subject: Anonymity of KEMs in the QROM. NIST PQC Forum. `https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/8k3MhD_-5stk/m/TWGKtuL4BgAJ`.
17. J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In *AFRICACRYPT 18*, pages 282–305, 2018.
18. J.-P. D'Anvers, A. Karmakar, S. S. Roy, F. Vercauteren, J. M. B. Mera, M. V. Beirendonck, and A. Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.
19. A. W. Dent. A designer's guide to KEMs. In *9th IMA International Conference on Cryptography and Coding*, pages 133–151, 2003.
20. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Online-extractability in the quantum random-oracle model. In *EUROCRYPT 2022, Part III*, pages 677–706, 2022.
21. P. Farshim, C. Orlandi, and R. Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symm. Cryptol.*, 2017(1):449–473, 2017.
22. A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama. Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In *ASIACCS 13*, pages 83–94, 2013.
23. A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. *Des. Codes Cryptogr.*, 76(3):469–504, 2015.
24. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, pages 537–554, 1999.
25. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, 2013.
26. P. Grubbs, V. Maram, and K. G. Paterson. Anonymous, robust post-quantum public key encryption. In *EUROCRYPT 2022, Part III*, pages 402–432, 2022.

27. D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017, Part I*, pages 341–371, 2017.
28. K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. In *PKC 2020, Part II*, pages 389–422, 2020.
29. K. Hövelmanns, A. Hülsing, and C. Majenz. Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform. In *ASIACRYPT 2022 (to appear)*, 2022.
30. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018, Part III*, pages 96–125, 2018.
31. H. Jiang, Z. Zhang, and Z. Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *PKC 2019, Part II*, pages 618–645, 2019.
32. S. Katsumata, K. Kwiatkowski, F. Pintore, and T. Prest. Scalable ciphertext compression techniques for post-quantum KEMs and their applications. In *ASIACRYPT 2020, Part I*, pages 289–320, 2020.
33. V. Kuchta, A. Sakzad, D. Stehlé, R. Steinfeld, and S. Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *EUROCRYPT 2020, Part III*, pages 703–728, 2020.
34. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.
35. B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *PKC 2012*, pages 206–224, 2012.
36. X. Liu and M. Wang. QCCA-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In *PKC 2021, Part I*, pages 3–26, 2021.
37. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018, Part III*, pages 520–551, 2018.
38. K. Sako. An auction protocol which hides bids of losers. In *PKC 2000*, pages 422–432, 2000.
39. P. Schwabe, R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, G. Seiler, and D. Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.
40. P. Schwabe, D. Stebila, and T. Wiggers. Post-quantum TLS without handshake signatures. In *ACM CCS 2020*, pages 1461–1480, 2020.
41. E. E. Targhi and D. Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *TCC 2016-B, Part II*, pages 192–216, 2016.
42. D. Unruh. Revocable quantum timed-release encryption. In *EUROCRYPT 2014*, pages 129–146, 2014.
43. D. Unruh. Post-quantum verification of Fujisaki-Okamoto. In *ASIACRYPT 2020, Part I*, pages 321–352, 2020.
44. K. Xagawa. Anonymity of NIST PQC round 3 KEMs. In *EUROCRYPT 2022, Part III*, pages 551–581, 2022.
45. M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7–8), 2015.
46. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *CRYPTO 2019, Part II*, pages 239–268, 2019.

```
$\mathbf{Expt}^{\text{SPR-otCCA}}_{\mathsf{DEM},\mathcal{A}}(\lambda)$
    $k \leftarrow_\$ \mathcal{K}$
    $b \leftarrow_\$ \{0,1\}$
    $(m, \text{state}) \leftarrow \mathcal{A}(1^\lambda)$
    $c_0^* \leftarrow \mathsf{E}(k, m)$
    $c_1^* \leftarrow_\$ \mathcal{C}_{|m|}$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}(\cdot)}(c_b^*, \text{state})$
    return $[b' = b]$
```

```
$\textsc{Dec}_a(c)$
    if $c = a$ then return $\perp$
    $m \leftarrow \mathsf{D}(k, c)$
    return $m$
```

Fig. 11. SPR-otCCA game for DEM schemes.

Supplemental Materials

A Missing Definitions

A.1 Data Encapsulation Mechanism (DEM)

The model for DEM schemes is summarized as follows:

Definition 8. *A DEM scheme* DEM *consists of the following pair of polynomial-time algorithms* (E, D)*:*

- $\mathsf{E}(k, m) \to c$*: an encapsulation algorithm that takes as input key* $k \in \mathcal{K}$ *and data* $m \in \mathcal{M}$*, and outputs ciphertext* $c \in \mathcal{C}$*.*
- $\mathsf{D}(k, c) \to m/\perp$*: a decapsulation algorithm that takes as input key* k *and ciphertext* c*, and outputs data* $m \in \mathcal{M}$ *or a rejection symbol* $\perp \notin \mathcal{M}$*.*

Definition 9 (DEM Correctness). *We say* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *has* perfect correctness *if for any* $k \in \mathcal{K}$ *and any* $m \in \mathcal{M}$*, we have*

$$\Pr[\mathsf{D}(k, c) = m : c \leftarrow \mathsf{E}(k, m)] = 1.$$

Definition 10 (One-time Strong Pseudorandomness of DEM). *Let the scheme* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *be a DEM. For* $m \in \mathcal{M}$*, let* $\mathcal{C}_{|m|} (\subseteq \mathcal{C})$ *be the ciphertext space defined by the length of data* m*. For any adversary* $\mathcal{A}$*, we define* $\mathcal{A}$*'s* SPR-otCCA *advantage against* DEM *as follows:*

$$\mathbf{Adv}^{\text{SPR-otCCA}}_{\mathsf{DEM}}(\mathcal{A}) := \left| \Pr[\mathbf{Expt}^{\text{SPR-otCCA}}_{\mathsf{DEM},\mathcal{A}}(\lambda) = 1] - \frac{1}{2} \right|,$$

where $\mathbf{Expt}^{\text{SPR-otCCA}}_{\mathsf{DEM},\mathcal{A}}(\lambda)$ *is an experiment described in Fig. 11. We say that* DEM *is* strongly pseudorandom under one-time chosen-ciphertext attack *(*SPR-otCCA secure*) if* $\mathbf{Adv}^{\text{SPR-otCCA}}_{\mathsf{DEM}}(\mathcal{A})$ *is negligible for any QPT adversary* $\mathcal{A}$*.*