Tighter proofs of CCA security in the quantum random oracle model

Nina Bindel[1], Mike Hamburg[2], Andreas Hülsing[3], and Edoardo Persichetti[4]

[1] Technische Universität Darmstadt, nbindel@cdc.informatik.tu-darmstadt.de
[2] Rambus, mhamburg@rambus.com
[3] Eindhoven University of Technology, andreas@huelsing.net
[4] Florida Atlantic University, epersichetti@fau.edu

Abstract. We revisit the construction of IND-CCA secure key encapsulation mechanisms (KEM) from public-key encryption schemes (PKE). We give new, tighter security reductions for several constructions. Our main result is a tight reduction for the security of the $U^{\not\perp}$-transform of Hofheinz, Hövelmanns, and Kiltz (TCC'17) which turns OW-CPA secure deterministic PKEs into IND-CCA secure KEMs. This result is enabled by a new one-way to hiding (O2H) lemma which gives a tighter bound than previous O2H lemmas in certain settings and might be of independent interest. We extend this result also to the case of PKEs with non-zero decryption failure probability, partially non-injective PKEs, and non-deterministic PKEs.

In addition, we analyze the impact of different variations of the $U^{\not\perp}$-transform discussed in the literature on the security of the final scheme. We consider the difference between explicit ($U^{\perp}$) and implicit ($U^{\not\perp}$) rejection, proving that security of the former implies security of the latter. We show that the opposite direction holds if the scheme with explicit rejection also uses key confirmation. Finally, we prove that (at least from a theoretic point of view) security is independent of whether the session keys are derived from message and ciphertext ($U^{\not\perp}$) or just from the message ($U^{\not\perp}_m$).

1 Introduction

If a general-purpose quantum computer can be built, it will break most widely-deployed public-key cryptography. The cryptographic community is busily designing new cryptographic systems to prepare for this risk. These systems typically consist of an algebraic structure with cryptographic hardness properties, plus a symmetric cryptography layer which transforms the algebraic structure into a higher level primitive like a public-key encryption (PKE) scheme, a key encapsulation mechanism (KEM), or a signature scheme. The algebraic structures underlying these so-called "post-quantum" systems have new properties, and the quantum threat model requires changes in the way security is analyzed. Therefore the transformations turning the algebraic structures into cryptosystems have to be freshly examined.

In this work we focus on the construction of secure KEMs. In this setting the algebraic structures usually provide a PKE from which a KEM is derived via a generic transform. A new property of the algebraic structures used in many post-quantum PKEs and KEMs gives them malleable ciphertexts, so they are at risk from chosen-ciphertext attacks (CCA) [HNP+03]. The standard defenses against CCA are variants of the Fujisaki-Okamoto (FO) transform [FO99]. Known security proofs for the FO transform use the random oracle model (ROM) [BR93]. This is for two reasons. First, the FO transform has a circular structure–it chooses coins for encryption according to the message being encrypted. This leads to obstacles which we do not know how to overcome when proving security in the standard model. In the ROM, we circumvent this by re-programming. Second, in the ROM a reduction learns all the adversary's queries to the random oracle. This allows us to formalize the intuition that an adversary must have known a challenge plaintext and to extract said plaintext.

Since we are concerned with security against quantum attackers, we need to extend these proofs to the quantum-accessible random oracle model (QROM) [BDF+11]. This comes with two challenges for our setting. On the one hand, in the QROM the adversary can query all inputs in superposition. Hence, it is no longer trivial to break the circular dependency by re-programming, which results in security bounds that do not tightly match known attacks. On the other hand, a reduction cannot learn the adversarial queries by simple observation anymore. The reason is that observation of a quantum state requires a measurement which disturbs the state. Hence, more advanced techniques are required.

1.1 Our Contribution

QROM analysis of KEMs has advanced rapidly over the past several years. The initial solutions were loose by a factor of up to q^6 [TU16,HHK17], where q is the number of times the adversary queries the random oracle. This has improved to q^2 [SXY18,JZC+17] and finally to q [JZM19a,JZM19c]. Some work provides tight proofs under stronger assumptions [SXY18,XY18]. Our work provides a tight proof of KEM security for KEMs constructed from deterministic PKEs (Theorem 2), except for a square-root loss which might be impossible to avoid [JZM19b]. For KEMs constructed from randomized PKEs our bound is still loose by a factor of up to q (Theorem 1).

At the heart of our bound is a new one-way to hiding (O2H) lemma which gives a tighter bound than previous O2H lemmas (Lemma 5). This comes at the cost of limited applicability. O2H lemmas allow to bound the difference in the success probability of an adversary when replacing its oracle function by a similar function. Previous lemmas lost a factor of roughly the number of the adversary's queries to this oracle or its square-root. Our lemma does not incur any such loss. On the downside, our lemma only applies if the reduction has access to both oracle functions and if the functions only differ in one position. See Table 1 for a comparison.

Some post-quantum schemes feature an inherent probability of decryption failure, say $\delta > 0$. Such failures can be used in attacks, but they also complicate security proofs. As a result, previous bounds typically contain a term $q\sqrt{\delta}$ which is not known to be tight. However, most of the obstacles that arise in our CCA security proof can be avoided by assuming that encryption with a particular public key is injective. This is generally the case, even for imperfectly-correct systems; see Appendix D for a rough analysis of LWE schemes. In that case, the adversary's advantage is limited to the probability that it actually finds and submits a valid message that fails to decrypt. This means that our bounds apply to deterministic but failure-prone systems like one of the BIKE [ABB$^+$19] variants (BIKE-2).

Until today several variants of the FO-transform were proposed. We consider the four basic transforms $U^{\perp}, U_m^{\perp}, U^{\not\perp}, U_m^{\not\perp}$ and, in addition, we study $U_m^{\perp}$ in the presence of key confirmation. The two most notable differences reside in the use of implicit rejection ($U^{\not\perp}, U_m^{\not\perp}$) versus explicit rejection ($U^{\perp}, U_m^{\perp}$), and whether the derivation of the session key should depend on the ciphertext ($U_m^{\perp}, U_m^{\not\perp}$) or not ($U^{\perp}, U^{\not\perp}$). Another important decision is the use of key confirmation which we also partially analyze. We come to the following results. Security with implicit rejection implies security with explicit rejection (Theorem 3). The opposite holds if the scheme with explicit rejection also employs key confirmation (Theorem 4). Moreover, security is independent of the decision if the session key derivation depends on the ciphertext (Theorem 5).

Notation. We will use the following notation throughout the paper.

- For two sets X, Y, we write Y^X to denote the set of functions from X to Y.
- Let $H : X \to Y$ be a (classical or quantum-accessible) random oracle. Then we denote the programming of H at $x \in X$ to some $y \in Y$ as $H[x \to y]$.
- Let $\mathcal{A}$ be an algorithm. If $\mathcal{A}$ has access to a classical (resp., quantum-accessible) oracle H, we write $\mathcal{A}^H$ and call $\mathcal{A}$ an oracle (resp., quantum oracle) algorithm.

2 One-way to Hiding

ROM reductions typically simulate the random oracle in order to learn the adversary's queries. In the classical ROM, the adversary cannot learn any information about $H(x)$ without the simulator learning both x and $H(x)$. In the QROM things are not so simple, because measuring or otherwise recording the queries might collapse the adversary's quantum state and change its behavior. However, under certain conditions the simulator can learn the queries using "One-way to Hiding" (O2H) techniques going back to [Unr15]. We will use the O2H techniques from [AHU18], and introduce a novel variant that allows for tighter results.

Consider two quantum-accessible oracles $G, H : X \to Y$. The oracles do not need to be random. Suppose that G and H differ only on some small set $S \subset X$, meaning that $\forall x \notin S, G(x) = H(x)$. Let $\mathcal{A}$ be an oracle algorithm that takes an input z and makes at most q queries to G or H. Possibly $\mathcal{A}$ makes them in parallel. Therefore, suppose that the query depth, i.e., the maximum number of sequential invocations of the oracle [AHU18], is at most $d \leq q$. If $\mathcal{A}^G(z)$ behaves differently from $\mathcal{A}^H(z)$, then the O2H techniques give a way for the simulator to find some $x \in S$ with probability dependent on d and q.

We will use the following three O2H lemmas.

- Lemma 1 (original O2H) is the most general: the simulator needs to provide only G or H but it has the least probability of success.
- Lemma 3 (semiclassical O2H) has a greater probability of success, but requires more from the simulator: for each query x, the simulator must be able to recognize whether $x \in S$, and if not it must return $G(x) = H(x)$.
- Lemma 5 (our new "double-sided" O2H) gives the best probability of success, but it requires the simulator to evaluate both G and H in superposition. It also can only extract $x \in S$ if S has a single element. If S has many elements, but the simulator knows a function f such that $\{f(x) : x \in S\}$ has a single element, then it can instead extract that element $f(x)$.

We summarize the three variants of O2H as shown in Table 1. In all cases, there are two oracles H and G that differ in some set S, and the simulator outputs $x \in S$ with some probability ϵ. The lemma then shows an upper bound on the difference between $\mathcal{A}^H$ and $\mathcal{A}^G$ as a function of ϵ.

Variant	Lemma	Ref	Oracles differ	Sim. must know	Bound
Original	Lem. 1	[AHU18]	Arbitrary	H or G	$2d\sqrt{\epsilon}$
Semi-classical	Lem. 3	[AHU18]	Arbitrary	S and $(H\backslash S$ or $G\backslash S)$	$2\sqrt{d\epsilon}$
Double-sided	Lem. 5	this work	One place	H and G	$2\sqrt{\epsilon}$

Table 1. Comparison of O2H variants

Arbitrary joint distribution The O2H lemmas allow (G, H, S, z) to be random with arbitrary joint distribution. This is stronger than (G, H, S, z) being arbitrary fixed objects, because the probabilities in the lemma include the choice of (G, H, S, z) in addition to $\mathcal{A}$'s coins and measurements. Also, the lemmas are still true if the adversary consults other oracles which are also drawn from a joint distribution with (G, H, S, z).

2.1 Original O2H

We begin with the original O2H which first appeared in [Unr15]. We use the phrasing from [AHU18] as it is more general and more consistent with our other lemmata.

Lemma 1 (One-way to hiding; [AHU18] Theorem 3). *Let $G, H : X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $\forall x \notin S, G(x) = H(x)$. (G, H, S, z) may have arbitrary joint distribution. Furthermore, let $\mathcal{A}^H$ be a quantum oracle algorithm which queries H with depth at most d. Let Ev be an arbitrary classical event. Define an oracle algorithm $\mathcal{B}^H(z)$ as follows: Pick $i \xleftarrow{\$} \{1, \dots, d\}$. Run $\mathcal{A}^H(z)$ until just before its ith round* *of queries to H. Measure all query input registers in the computational basis, and output the set T of measurement outcomes. Let*

$$P_{\text{left}} := \Pr[\mathsf{Ev} : \mathcal{A}^H(z)], \quad P_{\text{right}} := \Pr[\mathsf{Ev} : \mathcal{A}^G(z)],$$
$$P_{\text{guess}} := \Pr[S \cap T \neq \varnothing : T \leftarrow \mathcal{B}^H(z)].$$

Then

$$|P_{\text{left}} - P_{\text{right}}| \leq 2d\sqrt{P_{\text{guess}}} \qquad \text{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2d\sqrt{P_{\text{guess}}}.$$

The same result holds with $\mathcal{B}^G(z)$ instead of $B^H(z)$ in the definition of P_{guess}.

From this lemma we conclude the following result for pseudo-random functions (PRFs, see Definition 10). It intuitively states that a random oracle makes a good PRF, even if the distinguisher is given full access to the random oracle in addition to the PRF oracle.

Corollary 1 (PRF based on random oracle). *Let $H : (K \times X) \to Y$ be a quantum-accessible random oracle. This function may be used as a quantum-accessible PRF $F_k(x) := H(k, x)$ with a key $k \xleftarrow{\$} K$. Suppose a PRF-adversary $\mathcal{A}$ makes q queries to H at depth d, and any number of queries to F_k at any depth. Then*

$$\text{Adv}^{\mathsf{PRF}}_{F_k}(\mathcal{A}) \leq 2\sqrt{dq/|K|}.$$

Proof. The adversary's goal is to distinguish (F_k, H) from (F, H), where F is an unrelated uniformly random function. This is the same as distinguishing $(F, H[(k, x) \to F(x)])$ from (F, H), and the set of differences between these two H-oracles is $S := \{k\} \times X$. By Lemma 1, the distinguishing advantage is at most $2d\sqrt{P_{\text{guess}}}$, where $P_{\text{guess}} = \Pr[\exists (k', x) \in Q : k' = k]$, for a random round Q of parallel queries made by $\mathcal{A}^{F,H}$.

Since $\mathcal{A}^{F,H}$ has no information about k, and in expectation Q contains q/d parallel queries, we have $P_{\text{guess}} \leq q/(d \cdot |K|)$, so

$$\text{Adv}^{\mathsf{PRF}}_{F_k}(\mathcal{A}) \leq 2d\sqrt{q/(d \cdot |K|)} = 2\sqrt{dq/|K|}$$

as claimed. □

2.2 Semi-classical O2H

We now move on to semi-classical O2H. Here $\mathcal{B}$ is defined in terms of *punctured oracles* [AHU18], which measure whether the input is in a set S as defined next.

Definition 1 (Punctured oracle). *Let $H : X \to Y$ be any function, and $S \subset X$ be a set. The oracle $H \backslash S$ ("H punctured by S") takes as input a value x. It first computes whether $x \in S$ into an auxilliary qubit p, and measures p. Then it runs $H(x)$ and returns the result. Let* Find *be the event that any of the measurements of p returns 1.*

The event is called Find because if the simulator chooses to, it can immediately terminate the simulation and measure the value $x \in S$ which caused the event. The oracle is called "punctured" because if Find does not occur, $H\backslash S$ returns a result independent of H's outputs on S, as shown by the following lemma.

Lemma 2 (Puncturing is effective; [AHU18] Lemma 1). *Let $G, H : X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $\forall x \notin S, G(x) = H(x)$. (G, H, S, z) may have arbitrary joint distribution. Let $\mathcal{A}^H$ be a quantum oracle algorithm. Let Ev be an arbitrary classical event. Then*

$$\Pr[\mathsf{Ev} \wedge \neg\mathsf{Find} : x \leftarrow \mathcal{A}^{H\backslash S}(z)] = \Pr[\mathsf{Ev} \wedge \neg\mathsf{Find} : x \leftarrow \mathcal{A}^{G\backslash S}(z)].$$

Also, puncturing only disturbs the adversary's state when it is likely to Find.

Lemma 3 (Semi-classical O2H; [AHU18] Theorem 1). *Let $G, H : X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $\forall x \notin S, G(x) = H(x)$. (G, H, S, z) may have arbitrary joint distribution.*

Let $\mathcal{A}^H$ be a quantum oracle algorithm which queries H with depth at most d. Let Ev be an arbitrary classical event and let

$$\begin{aligned} P_{\text{left}} &:= \Pr[\mathsf{Ev} : \mathcal{A}^H(z)], \; P_{\text{right}} := \Pr[\mathsf{Ev} : \mathcal{A}^G(z)], \\ P_{\text{find}} &:= \Pr[\mathsf{Find} : \mathcal{A}^{G\backslash S}(z)] \quad \overset{\text{Lem. 2}}{=} \quad \Pr[\mathsf{Find} : \mathcal{A}^{H\backslash S}(z)]. \end{aligned}$$

Then

$$|P_{\text{left}} - P_{\text{right}}| \leq 2\sqrt{dP_{\text{find}}} \qquad \textit{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2\sqrt{dP_{\text{find}}}.$$

The theorem also holds with bound $\sqrt{(d+1)P_{\text{find}}}$ for the following alternative definitions of P_{right}:

$$\begin{aligned} P_{\text{right}} &:= \Pr[\mathsf{Ev} : \mathcal{A}^{H\backslash S}(z)] \\ P_{\text{right}} &:= \Pr[\mathsf{Ev} \wedge \neg\mathsf{Find} : \mathcal{A}^{H\backslash S}(z)] \quad \overset{\text{Lem. 2}}{=} \quad \Pr[\mathsf{Ev} \wedge \neg\mathsf{Find} : \mathcal{A}^{G\backslash S}(z)] \\ P_{\text{right}} &:= \Pr[\mathsf{Ev} \vee \;\; \mathsf{Find} : \mathcal{A}^{H\backslash S}(z)] \quad \overset{\text{Lem. 2}}{=} \quad \Pr[\mathsf{Ev} \vee \;\; \mathsf{Find} : \mathcal{A}^{G\backslash S}(z)] \end{aligned}$$

We might expect that if the adversary has no information about S, then P_{find} would be at most $q|S|/|X|$. But this is not quite true: the disturbance caused by puncturing gives the adversary information about S. This increases $\mathcal{A}$'s chances, but only by a factor of 4, as explained next.

Lemma 4 (Search in semi-classical oracle; [AHU18] Theorem 2). *Let $H : X \to Y$ be a random function, let z be a random value, and let $S \subset X$ be a random set. (H, S, z) may have arbitrary joint distribution. Let $\mathcal{A}^H$ be a* *quantum oracle algorithm which queries H at most q times with depth at most d.*

Let $\mathcal{B}^H(z)$ and P_{guess} be defined as in Lemma 1. Then

$$\Pr[\mathsf{Find} : \mathcal{A}^{H \setminus S}(z)] \leq 4dP_{\text{guess}}.$$

In particular, if for each $x \in X$, $\Pr[x \in S] \leq \epsilon$ (conditioned on z, on other oracles $\mathcal{A}$ has access to, and on other outputs of H) then

$$\Pr[\mathsf{Find} : \mathcal{A}^{H \setminus S}(z)] \leq 4q\epsilon.$$

2.3 Double-sided O2H

We augment these lemmas with a new O2H lemma which achieves a tighter bound focusing on a special case. This focus comes at the price of limited applicability. Our lemma applies when the simulator can simulate both G and H.

Lemma 5 (Double-sided O2H). *Let $G, H : X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $\forall x \notin S, G(x) = H(x)$. (G, H, S, z) may have arbitrary joint distribution. Let $f : X \to W \subseteq \{0,1\}^n$ be a function such that*

$$\{f(x) : x \in S\} = \{w^*\}$$

is a single element. Let $\mathcal{A}^H$ be a quantum oracle algorithm. Let Ev be an arbitrary classical event. We will define another quantum oracle algorithm $\mathcal{B}^{G,H}(z)$ which returns either w^* or a failure symbol $\perp$. $\mathcal{B}$ runs in about the same amount of time as $\mathcal{A}$, but when $\mathcal{A}$ queries H, $\mathcal{B}$ queries both G and H, and also runs f twice. Let

$$P_{\text{left}} := \Pr[\mathsf{Ev} : \mathcal{A}^H(z)],\ P_{\text{right}} := \Pr[\mathsf{Ev} : \mathcal{A}^G(z)],\ P_{\text{extract}} := \Pr[\mathcal{B}^{G,H}(z) \to w^*].$$

Then

$$|P_{\text{left}} - P_{\text{right}}| \leq 2\sqrt{P_{\text{extract}}} \qquad \textit{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2\sqrt{P_{\text{extract}}}.$$

Proof. See Appendix B.

Note that if $S = \{x^*\}$ is already a single element, then we may take f as the identity. In this case $\mathcal{B}$ will return either $\perp$ or x^*.

3 KEM and PKE Security Proofs

We are now ready to get to the core of our work. All the relevant security notions are given in Appendix A. The implications are summarized in Figure 1.

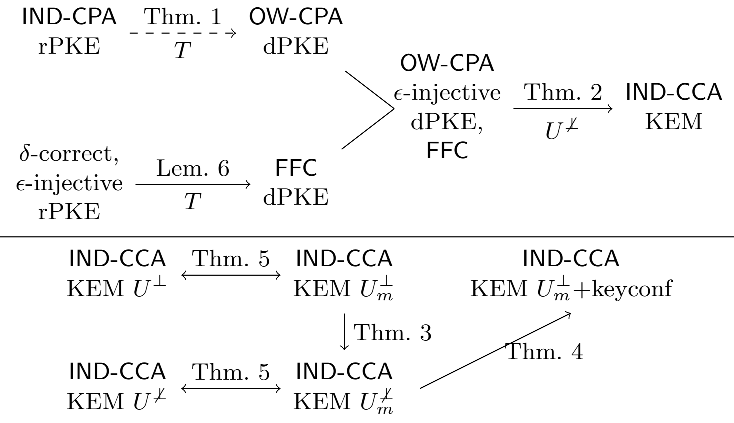

Fig. 1. Relations of our security notions using transforms T and $U^{\not\perp}$ (above) and relations between the security of different types of U-constructions (below). The solid lines show implications which are tight with respect to powers of q and/or d, and the dashed line shows a non-tight implication.

3.1 Derandomization: IND-CPA P $\overset{\text{QROM}}{\Rightarrow}$ OW-CPA $T(\mathsf{P}, G)$

The T transform [HHK17] converts a rPKE $\mathsf{P} = (\text{Keygen}, \text{Encr}, \text{Decr})$ to a dPKE $T(\mathsf{P}, G) = (\text{Keygen}, \text{Encr}_1, \text{Decr})$ by using a hash function $G : \mathcal{M} \to \mathcal{R}$, modeled as random oracle, to choose encryption coins, where

$$\text{Encr}_1(\text{pk}, m) := \text{Encr}(\text{pk}, m;\ G(m)).$$

The following theorem shows that if a PKE P is IND-CPA secure[5], then $\mathrm{T}(\mathsf{P}, G)$ is one-way secure in the quantum-accessible random oracle model.

Theorem 1. *Let* P *be an rPKE with messages in* $\mathcal{M}$ *and random coins in* $\mathcal{R}$. *Let* $G : \mathcal{M} \to \mathcal{R}$ *be a quantum-accessible random oracle. Let* $\mathcal{A}$ *be an* OW-CPA *adversary against* $\mathsf{P}' := T(\mathsf{P}, G)$. *Suppose that* $\mathcal{A}$ *queries* G *at most* q *times with depth at most* d.

Then we can construct an IND-CPA *adversary* $\mathcal{B}$ *against* P, *running in about the same time and resources as* $\mathcal{A}$, *such that*

$$\text{Adv}_{\mathsf{P}'}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) \leq (d+2) \cdot \left(\text{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}) + \frac{8(q+1)}{|\mathcal{M}|} \right).$$

Proof. See Appendix C.

Second preimages. In the traditional definition of one-way functions, the adversary wins by finding any m' where $\text{Encr}(\text{pk}, \mathrm{m}') = c^*$, whereas in our definition

[5] The theorem actually only requires a weaker notion, IND-KPA-security, in which the challenge messages are chosen at random instead of adversarially.

of $\mathsf{OW\text{-}CPA}$ the adversary must find m^* itself. This only matters if there is a second preimage, and thus a decryption failure. If P is δ-correct and ϵ-injective, it is easily shown that a definition allowing second preimages adds at most $\min(\delta, \epsilon)$ to the adversary's $\mathsf{OW\text{-}CPA}$-advantage.

Hashing the public key. Many KEMs use a variant of T which sets the coins to $G(\mathrm{pk}, m)$. This is a countermeasure against multi-key attacks. In this paper we only model single-key security, so we omit pk from the hashes for brevity. The same also applies to the other transforms later in this paper, such as $U^{\not\perp}$.

3.2 Deterministic P: OW-CPA $\mathsf{P} \overset{\mathbf{QROM}}{\Rightarrow}$ IND-CCA $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$

Our $\mathsf{OW\text{-}CPA}$ to $\mathsf{IND\text{-}CCA}$ conversion is in the style of [JZM19c]. However, that bound is based on the failure probability δ of a randomized encryption algorithm, whereas ours is based on the difficulty of finding a failure without access to the private key. This means our theorem applies to deterministic but imperfectly-correct algorithms, such as one of the three BIKE variants, BIKE-2 [ABB+19]. So instead we use injectivity and a game where the adversary tries to find ciphertexts which are valid but do not decrypt correctly.

Definition 2 (Valid ciphertext). *Let* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *be a dPKE. Call a ciphertext* c *"valid" for a public key* pk *of* P *if there exists* m *such that* $c = \mathrm{Encr}(\mathrm{pk}, \mathrm{m})$.

We introduce a new failure-finding experiment[6], to capture the probability that the adversary can find valid ciphertexts that cause a decryption failure.

Definition 3 (Finding Failing Ciphertext). *The find-failing-ciphertexts experiment* (FFC) *is shown in Figure 2. The* FFC*-advantage of an adversary* $\mathcal{A}$ *is defined by*

$$\mathrm{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{A}) := \Pr[\mathrm{Expt}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{A}) \to 1].$$

The $U^{\not\perp}$ transform converts a dPKE $\mathsf{P} = (\mathrm{Keygen}_{\mathsf{P}}, \mathrm{Encr}, \mathrm{Decr})$ into a KEM $\mathsf{K} = (\mathrm{Keygen}, \mathrm{Encaps}, \mathrm{Decaps})$ using a PRF $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times \mathcal{C} \to \mathcal{K}$ and a hash function $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$, modeled as random oracle [JZM19c,HHK17]. The PRF is used for implicit rejection, returning $\mathsf{F}(\mathrm{prfk}, c)$ in case of an invalid ciphertext using a secret prfk. The $U^{\not\perp}$ transform is defined in Figure 3. We also define variants $U_m^{\not\perp}, U^{\perp}, U_m^{\perp}$ of this transform, which make the following changes:

- On Encaps line 3 resp. Decaps line 7, the transformations $U_m^{\not\perp}$ and $U_m^{\perp}$ compute $H(m)$ resp. $H(m')$ instead of $H(m, c)$ resp. $H(m', c)$.
- On Decaps lines 4 and 6, the transformations $U^{\perp}$ and $U_m^{\perp}$ return $\perp$ instead of $\mathsf{F}(\mathrm{prfk}, c)$. These variants also don't need a prfk as part of the private key.

The transforms $U^{\perp}$ and $U_m^{\perp}$ are said to use *explicit rejection* because they return an explicit failure symbol $\perp$. $U^{\not\perp}$ and $U_m^{\not\perp}$ are said to use *implicit rejection.*

[6] It is a stretch to even call this an "experiment", because it may not be possible to efficiently determine whether the adversary succeeded. In future work we hope to force the adversary to find failing *message*, but this version is simpler to integrate into our proof.

```
$\mathrm{Expt}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{A})$:

$H \xleftarrow{\$} \mathcal{H}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()$
$L \leftarrow \mathcal{A}^{H}(\mathrm{pk})$
return $[\exists \mathrm{m} \in \mathcal{M}, c \in L : \ \mathrm{Encr}(\mathrm{pk}, \mathrm{m}) = c \ \wedge \ \mathrm{Decr}(\mathrm{sk}, c) \neq \mathrm{m}]$
```

Fig. 2. FFC experiment on a PKE P. The instantiation of H generalizes to any number of random oracles, including zero.

```
Keygen():

$(\mathrm{pk}, \mathrm{sk}_{\mathsf{P}}) \leftarrow \mathrm{Keygen}_{\mathsf{P}}()$
$\mathrm{prfk} \xleftarrow{\$} \mathcal{K}_{\mathsf{F}}$
$\mathrm{sk} \leftarrow (\mathrm{sk}_{\mathsf{P}}, \mathrm{prfk})$
return $(\mathrm{pk}, \mathrm{sk})$
```

```
Encaps(pk):

$m \xleftarrow{\$} \mathcal{M}$
$c \leftarrow \mathrm{Encr}(\mathrm{pk}, m)$
$K \leftarrow H(m, c)$
return $(K, c)$
```

```
Decaps(sk, $c$):

parse $\mathrm{sk} = (\mathrm{sk}_{\mathsf{P}}, \mathrm{prfk})$
$m' \leftarrow \mathrm{Decr}(\mathrm{sk}_{\mathsf{P}}, c)$
if $m' = \perp$:
    return $\mathsf{F}(\mathrm{prfk}, c)$
else if $\mathrm{Encr}(\mathrm{pk}, m') \neq c$):
    return $\mathsf{F}(\mathrm{prfk}, c)$
else: return $H(m', c)$
```

Fig. 3. Transform $U^{\not\perp}(\mathsf{P}, \mathsf{F}) := (\mathrm{Keygen}, \mathrm{Encaps}, \mathrm{Decaps})$.

The next theorem states that breaking the $\mathsf{IND\text{-}CCA}$ security of $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ requires either breaking the $\mathsf{OW\text{-}CPA}$ security of P, causing a decapsulation failure, or breaking the PRF used for implicit rejection. In particular, we need P to be an ϵ-injective dPKE as in Definition 6.

Theorem 2. *Let $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ be a quantum-accessible random oracle, $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times \mathcal{C} \to \mathcal{K}$ be a PRF. Let P be an ϵ-injective dPKE which is independent of H. Let $\mathcal{A}$ be an $\mathsf{IND\text{-}CCA}$ adversary against the KEM $U^{\not\perp}(\mathsf{P}, \mathsf{F})$, and suppose that $\mathcal{A}$ makes at most q_{dec} decryption queries. Then we can construct three adversaries running in about the same time and resources as $\mathcal{A}$:*

- *an $\mathsf{OW\text{-}CPA}$-adversary $\mathcal{B}_1$ against P*
- *a FFC-adversary $\mathcal{B}_2$ against P, returning a list of at most q_{dec} ciphertexts*
- *a PRF-adversary $\mathcal{B}_3$ against F*

such that

$$\mathrm{Adv}_{U^{\not\perp}(\mathsf{P})}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) \leq 2\sqrt{\mathrm{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)} + \mathrm{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + 2 \cdot \mathrm{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3) + \epsilon.$$

In the common case that $\mathsf{F}(\mathrm{prfk}, c)$ is implemented as $H(\mathrm{prfk}, c)$ it holds that if $\mathcal{A}$ makes q queries at depth d, then

$$\mathrm{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3) \overset{\text{cor. 1}}{\leq} 2\sqrt{dq/|M|}.$$

Proof. Our proof is by a series of games. In some later games, we will define an outcome "draw" which is distinct from a win or loss. A draw counts as halfway between a win and a loss, as described by the adversary's score w_i:

$$\begin{aligned} w_i &:= \Pr[\mathcal{A} \text{ wins: Game } i] + \frac{1}{2}\Pr[\text{Draw: Game } i] \\ &= \frac{1}{2}\left(\Pr[\mathcal{A} \text{ wins: Game } i] - \Pr[\mathcal{A} \text{ loses: Game } i]\right). \end{aligned}$$

Game 0 ($\mathsf{IND\text{-}CCA}$). *This is the original* $\mathsf{IND\text{-}CCA}$ *game against the KEM* $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$*, cf. Definition 12.*

Game 1 (PRF is random). *Game 1 is the same as Game 0, except the simulator replaces* $\mathsf{F}(\text{prfk}, \cdot)$ *with a random function* $R \xleftarrow{\$} \mathcal{K}^{\mathcal{C}}$ *.*

We construct a PRF-adversary $\mathcal{B}_3$ (cf. Definition 10) which replaces its calls to $\mathsf{F}(\text{prfk}, \cdot)$ by calls to its oracle, runs $\mathcal{A}$, and outputs 1 if $\mathcal{A}$ wins and 0 otherwise. Now, by construction $\Pr_{k \xleftarrow{\$} \mathcal{K}}\left[\mathcal{B}^{\mathsf{F}(k,\cdot)} = 1\right] = \Pr[\mathcal{A} \text{ wins: Game } 0]$ and $\Pr_{R \xleftarrow{\$} \mathcal{K}^{\mathcal{C}}}\left[\mathcal{A}^{R(\cdot)} = 1\right] = \Pr[\mathcal{A} \text{ wins: Game } 1]$. Hence,

$$|\Pr[\mathcal{A} \text{ wins: Game } 1] - \Pr[\mathcal{A} \text{ wins: Game } 0]| = \mathrm{Adv}_F^{\mathrm{PRF}}(\mathcal{A}).$$

Game 2 (Draw on fail or non-injective pk**).** *Let* Fail *be the event that one or more of* $\mathcal{A}$*'s decapsulation queries* $D(c)$ *fails to decrypt, meaning that* $c = \mathrm{Encr}(\mathrm{pk}, \mathrm{m})$ *for some* m*, but* $\mathrm{Decr}(\mathrm{sk}, c) \neq \mathrm{m}$*. Let* NonInj *be the event that* $\mathrm{Encr}(\mathrm{pk}, \cdot)$ *is not injective, and let* $\mathsf{Draw} := \mathsf{Fail} \vee \mathsf{NonInj}$*. In Game 2 and onward, if* Draw *occurs then the game continues, but at the end it is a draw instead of the adversary winning or losing.*

Let $d_i := \Pr[\mathsf{Draw} : \text{Game } i]$. Then $|w_2 - w_1| \leq \frac{1}{2}d_2$. Note that the event Draw is a well-defined classical event and does not depend on H, even though the simulator might not be able to determine efficiently whether it occurred.

Game 3 (Reprogram $H(m, c)$ to $R(c)$). *Game 3 is the same as Game 2, but the simulator reprograms* $H(m, c)$ *where* $c = \mathrm{Encr}(\mathrm{pk}, m)$ *to return* $R(c)$*.*

This produces the same win and draw probabilities as Game 2 as explained next. For each m, the value $H(m, \mathrm{Encr}(\mathrm{pk}, m))$ is changed to a uniformly, independently random value, except when the game is already a draw:

- It is uniformly random because R is uniformly random.
- It is independent of $H(m', c)$ for $m' \neq m$ because $\mathrm{Encr}(\mathrm{pk}, \cdot)$ is injective or else the game is a draw.
- H calls $R(c)$ only for valid ciphertexts $c = \mathrm{Encr}(\mathrm{pk}, m')$. On the other hand, the decapsulation oracle only calls $R(c')$ for rejected ciphertexts c', i.e. ones where $c' \neq \mathrm{Encr}(\mathrm{pk}, \mathrm{Decr}(\mathrm{sk}, c'))$. If a valid ciphertext has been rejected and passed to R in this way, then Draw has occurred and the return value of R does not affect w_i or d_i.

Therefore $w_3 = w_2$ and $d_3 = d_2$.

Game 4 (Decapsulation oracle returns $R(c)$). *Game 4 is the same as Game 3, but the simulated decapsulation oracle simply returns $R(c)$ for all ciphertexts other than the challenge (for which it still returns $\perp$).*

In fact, the decapsulation oracle was already doing this in Game 3: The original decapsulation returns either $H(m, c)$ with $c = \mathrm{Encr}(\mathrm{pk}, m)$ or $\mathsf{F}(\mathrm{prfk}, c)$, but both of those have been reprogrammed to return $R(c)$. Therefore $w_4 = w_3$ and $d_4 = d_3$. As of this game, the simulator doesn't use the private key anymore.

Bound draw. We now want to upper bound the draw probability. Let $\mathcal{B}_2$ be the algorithm which, given a public key pk, simulates Game 4 for $\mathcal{A}$ and outputs a list L of all of $\mathcal{A}$'s decapsulation queries. Then $\mathcal{B}_2$ is a FFC-adversary against P which runs in about the same time as $\mathcal{A}$ and succeeds whenever a draw occured during the game. Consequently,

$$d_2 = d_3 = d_4 \leq \mathrm{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + \epsilon.$$

Game 5 (Change shared secret). *In Game 5, the shared secret is changed to a uniformly random value r. If $b = 1$, then for all m such that* $\mathrm{Encr}(\mathrm{pk}, m) = c^*$, *the oracle $H(m)$ is reprogrammed to return r. If $b = 0$, then H is not reprogrammed.*

If $\mathrm{Encr}(\mathrm{pk}, \cdot)$ is injective, then this is the same distribution as Game 4, and otherwise the game is a draw. Therefore $w_5 = w_4$.

It remains to bound $\mathcal{A}$'s advantage in Game 5. The simulation still runs in about the same time as $\mathcal{A}$. Furthermore, the $b = 0$ and $b = 1$ cases are now distinguished by a single return value from the H oracle. Hence, we can consider two oracles H and $H' := H[m^* \to r]$ as required by Lemma 5. Then Lemma 5, states that there is an algorithm $\mathcal{B}_1$, running in about the same time as $\mathcal{A}$, such that for all H:

$$\begin{aligned}
|\Pr[\mathrm{Win} : b = 0] - \Pr[\mathrm{Lose} : b = 1]| &= \left| \begin{array}{r} \Pr[\mathcal{A} \to 0 \land \neg\mathsf{Draw} : b = 0] \\ - \Pr[\mathcal{A} \to 0 \land \neg\mathsf{Draw} : b = 1] \end{array} \right| \\
&= \left| \begin{array}{r} \Pr[\mathcal{A}^{H} \to 0 \land \neg\mathsf{Draw}] \\ - \Pr[\mathcal{A}^{H'} \to 0 \land \neg\mathsf{Draw}] \end{array} \right| \\
&\leq 2\sqrt{\Pr[\mathcal{B}_1(\mathrm{pk}, c) \to m^*]}.
\end{aligned}$$

The same is true in expectation over H by Jensen's inequality:

$$\begin{aligned}
&\mathrm{E}\left[2\sqrt{\Pr[\mathcal{B}_1(\mathrm{pk}, c) \to m^*]} : H \xleftarrow{\$} \mathcal{K}^{(\mathcal{M}\times\mathcal{C})}\right] \\
&\quad \leq 2\sqrt{\mathrm{E}\left[\Pr[\mathcal{B}_1(\mathrm{pk}, c) \to m^*] : H \xleftarrow{\$} \mathcal{K}^{(\mathcal{M}\times\mathcal{C})}\right]} \\
&\quad = 2\sqrt{\mathrm{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)}
\end{aligned}$$

so that

$$|\Pr[\text{Win} : b = 0] - \Pr[\text{Lose} : b = 1]| \le 2\sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)}.$$

Likewise, for the same adversary $\mathcal{B}_1$,

$$|\Pr[\text{Win} : b = 1] - \Pr[\text{Lose} : b = 0]| \le 2\sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)}.$$

Since b is either 0 or 1 each with probability $\frac{1}{2}$, we have by the triangle inequality:

$$|\Pr[\text{Win}] - \Pr[\text{Lose}]| \le 2\sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)}$$

so that $\left|w_5 - \frac{1}{2}\right| \le \sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)}$.

Summing up the differences in the previous games, we have

$$\left|w_0 - \frac{1}{2}\right| \le \sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)} + \frac{1}{2}\text{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + \frac{\epsilon}{2} + \text{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3)$$

and finally

$$\text{Adv}_{U^{\not\perp}(\mathsf{P})}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) \le 2\sqrt{\text{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1)} + 2 \cdot \text{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3) + \text{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + \epsilon.$$

This completes the proof of Theorem 2. □

Tightness This bound is essentially tight, since breaking the one-wayness of P and finding decryption failures are both known to result in attacks. Breaking the PRF harms security if and only if implicit rejection is more secure than explicit rejection. For a correct P the bound boils down to the first two terms of the sum. The square-root loss arises from OW being a weaker security notion than IND [MW18], i.e., harder to break, and recent results [JZM19b] suggest that the square-root loss might be unavoidable in the quantum setting.

3.3 Decryption failures

When the dPKE is constructed by derandomizing an rPKE, we can also bound the FFC advantage.

Lemma 6. *Let* $\mathsf{P} = (\text{Keygen}, \text{Encr}, \text{Decr})$ *be a* δ*-correct,* ϵ*-injective rPKE with messages in* $\mathcal{M}$ *and randomness in* $\mathcal{R}$*. Let* $G : \mathcal{M} \to \mathcal{R}$ *be a random oracle, so that* $T(\mathsf{P}, G) := (\text{Keygen}, \text{Encr}_1, \text{Decr})$ *is a derandomized version of* P*. Let* $\mathcal{A}$ *be a* FFC *adversary against* $T(\mathsf{P}, G)$ *which makes at most* q *queries at depth* d *to* G *and returns a list of at most* q_{dec} *ciphertexts. Then*

$$\text{Adv}_{T(\mathsf{P},G)}^{\mathsf{FFC}}(\mathcal{A}) \le ((4d + 1)\delta + \sqrt{3\epsilon}) \cdot (q + q_{\text{dec}}) + \epsilon.$$

Proof. See Appendix E.

Note that if ϵ is negligible, and if the adversary can recognize which ciphertexts will fail, then this is a Grover bound.

4 Explicit rejection and key confirmation

We now turn to systems with explicit rejection or key confirmation. The next theorem shows that the transform $U^{\perp}$ (with explicit rejection) never yields KEMs that are more secure than KEMs constructed via $U^{\not\perp}$ (with implicit rejection).

Theorem 3 (Explicit → implicit). *Let* P *be a dPKE. Let* $\mathcal{A}$ *be an* $\mathsf{IND\text{-}CCA}$ *adversary against* $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$*. Then there is an* $\mathsf{IND\text{-}CCA}$ *adversary* $\mathcal{B}$ *against* $U^{\perp}(\mathsf{P}, H)$*, running in about the same time and resources as* $\mathcal{B}$*, such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{U^{\not\perp}(\mathsf{P},\mathsf{F},H)}(\mathcal{A}) = \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{U^{\perp}(\mathsf{P},H)}(\mathcal{B}).$$

Proof. The only difference between $U^{\perp}(\mathsf{P}, H)$ and $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ is that where the former would reject a ciphertext c by returning $\perp$, the latter instead returns $\mathsf{F}(\mathrm{prfk}, c)$. So the adversary $\mathcal{B}$ can simply choose a random PRF key prfk, run $\mathcal{A}$, and output $\mathcal{A}$'s result. $\mathcal{B}$ forwards all of $\mathcal{A}$'s queries to its oracles and returns the responses with the only difference that in case the decapsulation oracle returns $\perp$, $\mathcal{B}$ returns $\mathsf{F}(\mathrm{prfk}, c)$. The algorithm $\mathcal{B}$ perfectly simulates the $\mathsf{IND\text{-}CCA}$ game for $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ and hence $\mathcal{A}$ succeeds with the same success probability as in the original game. □

On the other hand, explicit rejection is secure if key confirmation is used. Key confirmation refers to adding a hash of the message to the cipher text. Let τ be the number of bits desired for the key-confirmation tag. For a PKE $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ define the transform $C(\mathsf{P}, H_t, \tau) := (\mathrm{Keygen}, \mathrm{Encr}_1, \mathrm{Decr}_1)$ using a random oracle $H_t : \mathcal{M} \to \{0, 1\}^{\tau}$ as in Figure 4.

```
$\mathrm{Encr}_1(\mathrm{pk}, m)$:
$c \leftarrow \mathrm{Encr}(\mathrm{pk}, m)$
$t \leftarrow H_t(m)$
return $(c, t)$

$\mathrm{Decr}_1(\mathrm{sk}, (c, t))$:
$m' \leftarrow \mathrm{Decr}(\mathrm{sk}_{\mathsf{P}}, c)$
if $H_t(m') \neq t$:
    return $\perp$
return $m'$
```

Fig. 4. Transform $C(\mathsf{P}, t) := (\mathrm{Keygen}, \mathrm{Encr}_1, \mathrm{Decr}_1)$.

Theorem 4 (Implicit → explicit with key confirmation). *Let* P *be an* ϵ*-injective dPKE. Consider the KEM* $\mathsf{K}_1 := U^{\perp}_{m}(C(\mathsf{P}, H_t, \tau), H_s)$ *obtained from* P *applying the* C*-transform with random oracle* $H_t : \mathcal{M} \to \{0, 1\}^{\tau}$ *and the* $U^{\perp}_{m}$*-transform with independent random oracle* $H_s : \mathcal{M} \to \{0, 1\}^{\varsigma}$*. Let* $\mathsf{K}_2 := U^{\not\perp}_{m}(\mathsf{P}, \mathsf{F}, H)$ *be the KEM obtained from* P *applying the* $U^{\not\perp}_{m}$*-transform with random oracle* $H : \mathcal{M} \to \{0, 1\}^{\varsigma+\tau}$*.*

If $\mathcal{A}$ *is an* $\mathsf{IND\text{-}CCA}$*-adversary against* K_1 *which makes* q_{dec} *decapsulation queries, then it is also an* $\mathsf{IND\text{-}CCA}$*-adversary against* P_2 *and there is a* PRF*-adversary* $\mathcal{B}$ *against* F *which uses about the same time and resources as* $\mathcal{A}$*, such* *that:*

$$\mathrm{Adv}_{\mathsf{K}_1}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}_{\mathsf{K}_2}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) + \frac{q_{\mathrm{dec}}}{2^{\tau-1}} + 2 \cdot \mathrm{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}) + 2\epsilon.$$

Proof. The gist of the proof is that the tag and the shared secret are both computed by hashing m. This means we can emulate K_1, where the ς-bit shared secret and τ-bit tag are computed by splitting the $(\varsigma+\tau)$-bit shared secret of K_2. In the case that K_2 implicitly rejects a ciphertext, then it returns a pseudorandom key. The tag parsed from this key will not match the adversary's tag except with probability $2^{-\tau}$, so the emulated K_1 will almost always reject as well.

We defer the details to Appendix F.

Finally, we can show that hashing m is equivalent to hashing (m, c) in the next theorem.

Theorem 5 ($U_m \leftrightarrow U$). *Let* P *be a dPKE. Let* $\mathsf{K}_1 = U^{\perp}(\mathsf{P}, H_1)$ *and* $\mathsf{K}_2 = U_m^{\perp}(\mathsf{P}, H_2)$*. Then* K_1 *is* $\mathsf{IND\text{-}CCA}$ *secure if and only if* K_2 *is* $\mathsf{IND\text{-}CCA}$ *secure. In otherwords, if there is an adversary* $\mathcal{A}$ *against one, then there is an adversary* $\mathcal{B}$ *against the other, running in about the same time and with the same advantage.*

The same is true for $U^{\not\perp}$ *and* $U_m^{\not\perp}$*.*

Proof. This is a simple indifferentiability argument. In both the encapsulation and decapsulation functions, the $\mathsf{IND\text{-}CCA}$ experiment against K_1 only calls $H_1(m, c)$ when $c = \mathrm{Encr}(\mathrm{pk}, m)$. So to simulate the K_1-experiment playing in an $\mathsf{IND\text{-}CCA}$ experiment against K_2 (with oracle $H_2 : \mathcal{M} \to \mathcal{K}$), sample fresh random oracle $H \xleftarrow{\$} \mathcal{K}^{(\mathcal{M},\mathcal{C})}$ and set

$$H_1(m, c) := \begin{cases} H_2(m) & \text{if } c = \mathrm{Encr}(\mathrm{pk}, m), \\ H(m, c) & \text{otherwise.} \end{cases}$$

This exactly simulates the $\mathsf{IND\text{-}CCA}$ experiment against K_1. In the other direction, to simulate the $\mathsf{IND\text{-}CCA}$ experiment against K_2 it suffices to redirect $H_2(m)$ to $H_1(m, \mathrm{Encr}(\mathrm{pk}, m))$.

The same technique works for $U^{\not\perp}$ and $U_m^{\not\perp}$. It also works for security notions other than $\mathsf{IND\text{-}CCA}$, such as $\mathsf{OW\text{-}CCA}$, $\mathsf{OW\text{-}qPVCA}$, etc. □

5 Acknowledgements

Special thanks to Daniel J. Bernstein, Edward Eaton and Mark Zhandry for helpful discussions.

References

ABB$^+$19. Nicolas Aragon, Paulo S. L. M. Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Shay Gueron, Tim Güneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Valentin Vasseur, and Gilles Zémor. BIKE: Bit flipping key encapsulation, 2019. `https://bikesuite.org`.

AHU18. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. Cryptology ePrint Archive, Report 2018/904, 2018. https://eprint.iacr.org/2018/904.

BDF⁺11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_3.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. doi:10.1145/168588.168596.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, August 1999. doi:10.1007/3-540-48405-1_34.

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017. doi:10.1007/978-3-319-70500-2_12.

HNP⁺03. Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The impact of decryption failures on the security of NTRU encryption. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 226–246. Springer, Heidelberg, August 2003. doi:10.1007/978-3-540-45146-4_14.

JZC⁺17. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. Post-quantum IND-CCA-secure KEM without additional hash. Cryptology ePrint Archive, Report 2017/1096, 2017. https://eprint.iacr.org/2017/1096.

JZM19a. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/052, 2019. https://eprint.iacr.org/2019/052.

JZM19b. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. On the non-tightness of measurement-based reductions for key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/494, 2019. https://eprint.iacr.org/2019/494.

JZM19c. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/134, 2019. https://eprint.iacr.org/2019/134.

MW18. Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 3–28. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78381-9_1.

NC00. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

SXY18. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018,* *Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_17.

TU16. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016. doi:10.1007/978-3-662-53644-5_8.

Unr15. Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, December 2015. URL: http://doi.acm.org/10.1145/2817206, doi:10.1145/2817206.

XY18. Keita Xagawa and Takashi Yamakawa. (tightly) qcca-secure key-encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2018/838, 2018. https://eprint.iacr.org/2018/838.

Zha18. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. Cryptology ePrint Archive, Report 2018/276, 2018. https://eprint.iacr.org/2018/276.

A Security Notions and Definitions

In this section we recall the definition of KEMs and PKEs. Additionally, we recall the respective security notions that are needed in this paper.

```
Classical random oracle $\mathcal{O}_H(x)$:
  1. $q_H \leftarrow q_H + 1$
  2. return $H(x)$
Quantum random oracle $\mathcal{O}_H(\sum_{x,t,z} \psi_{x,t,z} |x, t, z\rangle)$:
  1. $q_H \leftarrow q_H + 1$
  2. return $\sum_{x,t,z} \psi_{x,t,z} |x, t \oplus H(x), z\rangle$
```

Fig. 5. Definition of the classical and quantum random oracle

Definition 4 (Public-Key Encryption Schemes). *A* randomized *public-key encryption scheme (rPKE) is defined over a finite message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, a secret key space $\mathcal{SK}$ and a public key space $\mathcal{PK}$. It consists of a triple of algorithms* $\mathsf{P} = (\text{Keygen}, \text{Encr}, \text{Decr})$ *defined as follows.*

- $\text{Keygen}() \rightarrow (\text{pk}, \text{sk})$ *is a randomized algorithm that returns a secret key* $\text{sk} \in \mathcal{SK}$ *and a public key* $\text{pk} \in \mathcal{PK}$*.*
- $\text{Encr}(\text{pk}, \text{m}) \rightarrow c$ *is a randomized algorithm that takes as input a public key* pk *and a message* $\text{m} \in \mathcal{M}$*, and outputs a ciphertext* $c \in \mathcal{C}$*.*
- $\text{Decr}(\text{sk}, c) \rightarrow \{\text{m}', \bot\}$ *is a deterministic algorithm that takes as input a secret key* $\text{sk} \in \mathcal{SK}$ *and a ciphertext* $c \in \mathcal{C}$ *and returns either a message* $\text{m}' \in \mathcal{M}$ *or a failure symbol* $\bot \notin \mathcal{M}$*.*

A deterministic *public-key encryption scheme (dPKE) is defined the same way, except that* Encr *is a deterministic algorithm.*

Definition 5 (Correctness and failure probability of PKEs). *A PKE* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *is δ-correct if*

$$\mathrm{E}\left[\max_{\mathrm{m}\in\mathcal{M}} \Pr[\mathrm{Decr}(\mathrm{sk}, \mathrm{Encr}(\mathrm{pk}, \mathrm{m})) \neq \mathrm{m}]] : (\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()\right] \leq \delta.$$

We call δ the decryption failure probability of P*. We say* P *is correct if $\delta = 0$.*

Note that this definition works for a deterministic or randomized PKE, but for a deterministic PKE the term $\max_{\mathrm{m}\in\mathcal{M}} \Pr[\mathrm{Decr}(\mathrm{sk}, \mathrm{Encr}(\mathrm{pk}, \mathrm{m}))$ is either 0 or 1 for each keypair.

Definition 6 (Injectivity of PKEs). *A dPKE* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *is ϵ-injective if*

$$\Pr\left[\mathrm{Encr}(\mathrm{pk}, m) \textit{ is not injective} : (\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}(), H \xleftarrow{\$} \mathcal{H}\right] \leq \epsilon.$$

We say P *is injective if $\epsilon = 0$.*

Definition 7 (OW-CPA Advantage). *Let* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *be a dPKE or rPKE. The one-way under chosen-plaintext attacks* (OW-CPA) *experiment is shown in Figure 6. The* OW-CPA*-advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathrm{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) := \Pr[\mathrm{Expt}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) \rightarrow 1].$$

Note that some papers, e.g., [JZM19c], define OW-CPA-advantage this way, and some, e.g., [HHK17], instead use the looser condition that $\mathrm{Encr}(\mathrm{pk}, \mathrm{m}') = c^*$, particularly if Encr is deterministic. We use the definition in Figure 6 because it is more convenient for our proofs of Theorems 1 and 2.

Definition 8 (IND-CPA Advantage). *Let* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *be an rPKE. The indistinguishability under chosen-plaintext attacks* (IND-CPA) *experiment is shown in Figure 6. The* IND-CPA*-advantage of an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is defined as*

$$\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) := 2\left|\Pr[\mathrm{Expt}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) \rightarrow 1] - \frac{1}{2}\right|.$$

Note that IND-CPA is unachievable for dPKEs, because $\mathcal{A}$ can just test which message encrypts to c^*.

A weakening of IND-CPA is IND-KPA where the challenge messages are chosen by the experiment.

Definition 9 (IND-KPA Advantage). *Let* $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ *be an rPKE. The indistinguishability under known-plaintext attack* (IND-KPA) *experiment is shown in Figure 6. The* IND-KPA*-advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}KPA}}(\mathcal{A}) := 2\left|\Pr[\mathrm{Expt}_{\mathsf{P}}^{\mathsf{IND\text{-}KPA}}(\mathcal{A}) \rightarrow 1] - \frac{1}{2}\right|.$$

```
$\mathrm{Expt}^{\mathsf{OW\text{-}CPA}}_{\mathsf{P}}(\mathcal{A})$:

$H \xleftarrow{\$} \mathcal{H}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()$
$\mathrm{m}^* \xleftarrow{\$} \mathcal{M}$
$c^* \leftarrow \mathrm{Encr}(\mathrm{pk}, \mathrm{m}^*)$
$\mathrm{m}' \leftarrow \mathcal{A}^H(\mathrm{pk}, c^*)$
return $[\mathrm{m}^* = \mathrm{m}']$
```

```
$\mathrm{Expt}^{\mathsf{IND\text{-}KPA}}_{\mathsf{P}}(\mathcal{A})$:

$H \xleftarrow{\$} \mathcal{H}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()$
$\mathrm{m}_0, \mathrm{m}_1 \leftarrow \mathcal{M}$
$b \xleftarrow{\$} \{0, 1\}$
$c^* \leftarrow \mathrm{Encr}(\mathrm{pk}, \mathrm{m}^*_b)$
$b' \leftarrow \mathcal{A}^H(\mathrm{pk}, \mathrm{m}_0, \mathrm{m}_1, c^*)$
return $[b = b']$
```

```
$\mathrm{Expt}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P}}(\mathcal{A})$:

$H \xleftarrow{\$} \mathcal{H}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()$
$(st, \mathrm{m}_0, \mathrm{m}_1) \leftarrow \mathcal{A}^H_1(\mathrm{pk})$
$b \xleftarrow{\$} \{0, 1\}$
$c^* \leftarrow \mathrm{Encr}(\mathrm{pk}, \mathrm{m}^*_b)$
$b' \leftarrow \mathcal{A}^H_2(\mathrm{pk}, \mathrm{m}_0, \mathrm{m}_1, c^*, st)$
return $[b = b']$
```

Fig. 6. OW-CPA, IND-CPA, and IND-KPA of a PKE P. The instantiation of H generalizes to any number of random oracles, including zero.

Clearly IND-CPA ⇒ IND-KPA as any IND-KPA adversary can be used to break IND-CPA using it as $\mathcal{A}_2$ and simulating $\mathcal{A}_1$ by just sampling random messages.

Definition 10 (PRF Advantage). *Let* $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times X \to Y$ *be a pseudorandom function (PRF). We define the* PRF*-advantage of an adversary* $\mathcal{A}$ *as*

$$\mathrm{Adv}^{\mathrm{PRF}}_{F}(\mathcal{A}) = \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} \left[\mathcal{A}^{\mathsf{F}(k,\cdot)} = 1 \right] - \Pr_{R \xleftarrow{\$} Y^X} \left[\mathcal{A}^{R(\cdot)} = 1 \right] \right| .$$

Definition 11 (Key Encapsulation Mechanism). *A KEM* K *defined over the message space* $\mathcal{M}$*, the public key space* $\mathcal{PK}$*, the secret key space* $\mathcal{SK}$*, and the key space* $\mathcal{K}$*, is a triple of algorithms* $\mathsf{K} = (\mathrm{Keygen}, \mathrm{Encaps}, \mathrm{Decaps})$ *defined as follows.*

- $\mathrm{Keygen}() \to (\mathrm{pk}, \mathrm{sk})$ *is a randomized algorithm that returns a public key* $\mathrm{pk} \in \mathcal{PK}$ *and a secret key* $\mathrm{sk} \in \mathcal{SK}$*.*
- $\mathrm{Encaps}(\mathrm{pk}) \to (c, \kappa)$ *is a randomized algorithm that takes as input a public key* pk *and outputs a ciphertext* c *as well as a key* $\kappa \in K$*.*
- $\mathrm{Decaps}(\mathrm{sk}, c) \to \kappa$ *or* $\perp$ *is a deterministic algorithm that takes as input a secret key* $\mathrm{sk} \in \mathcal{SK}$ *and a ciphertext* c *and returns a key* $\kappa \in K$ *or a failure symbol* $\perp \notin K$*.*

As before, we use $\mathcal{H}$ to denote the space of functions from which the random hash function is randomly sampled if a proof for K is being given in the ROM.

Definition 12 (IND-CCA Advantage). *Let K be a KEM. The security experiment $\mathrm{Expt}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$ is defined in Figure 7 for an adversary $\mathcal{A}$ against K, given access to a (quantum-accessible) random oracle H and a classical decapsulation oracle D.*

We define the advantage of a classical (resp., quantum) adversary $\mathcal{A}$ against a KEM K in the classical (resp., quantum-accessible) random oracle model as

$$\mathrm{Adv}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) = \left|\Pr\left[\mathrm{Expt}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) = 1\right] - \frac{1}{2}\right|.$$

```
$\mathrm{Expt}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$:

$H \xleftarrow{\$} \mathcal{H}$
$(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}()$
$(c^*, k_0^*) \leftarrow \mathrm{Encaps}(\mathrm{pk})$
$k_1^* \xleftarrow{\$} K$
$b \xleftarrow{\$} \{0,1\}$
$b' \leftarrow \mathcal{A}^{H,D}(\mathrm{pk}, c^*, k_b^*)$
return $[b = b']$
```

```
Classical decapsulation oracle $D(c)$:

if $c = c^*$: return $\perp$
return $\mathrm{Decaps}(\mathrm{sk}, c)$
```

Fig. 7. IND-CCA security experiment in the (Q)ROM against an adversary $\mathcal{A}$

B Proof of Lemma 5

Lemma 5 (Double-sided O2H). *Let $G, H : X \to Y$ be random functions, let z be a random value, and let $S \subset X$ be a random set such that $\forall x \notin S, G(x) = H(x)$. (G, H, S, z) may have arbitrary joint distribution. Let $f : X \to W \subseteq \{0,1\}^n$ be a function such that*

$$\{f(x) : x \in S\} = \{w^*\}$$

is a single element. Let $\mathcal{A}^H$ be a quantum oracle algorithm. Let Ev be an arbitrary classical event. We will define another quantum oracle algorithm $\mathcal{B}^{G,H}(z)$ which returns either w^* or a failure symbol $\perp$. $\mathcal{B}$ runs in about the same amount of time as $\mathcal{A}$, but when $\mathcal{A}$ queries H, $\mathcal{B}$ queries both G and H, and also runs f twice. Let

$$P_{\text{left}} := \Pr[\mathsf{Ev} : \mathcal{A}^H(z)],\ P_{\text{right}} := \Pr[\mathsf{Ev} : \mathcal{A}^G(z)],\ P_{\text{extract}} := \Pr[\mathcal{B}^{G,H}(z) \to w^*].$$

Then

$$|P_{\text{left}} - P_{\text{right}}| \le 2\sqrt{P_{\text{extract}}} \qquad \textit{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \le 2\sqrt{P_{\text{extract}}}.$$

Proof. The outline of our proof is to use the compressed oracle framework from [Zha18] to instantiate a new random oracle $B : W \to \{0,1\}$. On query x, the simulator returns $H(x)$ if $B(f(x)) = 0$, or $G(x)$ if $B(f(x)) = 1$. The only value of $B(z)$ that affects the result is $b := B(w^*)$, so at the end of the computation the compressed oracle table for B must be either the empty table or $\{w^* \to b\}$. Our proof quantizes and simplifies this outline.

To begin, suppose that G and H are fixed and $\mathcal{A}$ is unitary. Consider an algorithm $\mathcal{B}_0^{H,G}$ which runs $\mathcal{A}^H$ and $\mathcal{A}^G$ in superposition, with an additional bit b signifying which oracle is being used. Then if $\mathcal{A}^G$ behaves differently from $\mathcal{A}^H$, the state of $\mathcal{A}$ will become entangled with b. We will use $|b\rangle = |{+}\rangle := (|0\rangle + |1\rangle)/\sqrt{2}$ to signify that $\mathcal{A}$ is using H, and $|b\rangle = |{-}\rangle := (|0\rangle - |1\rangle)/\sqrt{2}$ to signify that $\mathcal{A}$ is using G. That is:

$$\mathcal{B}_0^{H,G} := \frac{\mathcal{A}^H \otimes |{+}\rangle + \mathcal{A}^G \otimes |{-}\rangle}{\sqrt{2}}.$$

This $\mathcal{B}_0$ can be implemented as $\mathcal{A}$ with only the oracle queries changed. To do this, let b start in the state $(|{+}\rangle + |{-}\rangle)/\sqrt{2} = |0\rangle$. When $\mathcal{A}$ queries the oracle, $\mathcal{B}_0$ implements the following map on $|x, y, b\rangle$:

$$\begin{aligned} U(|x, y, {+}\rangle) &:= |x, y \oplus H(x), {+}\rangle\,, \\ U(|x, y, {-}\rangle) &:= |x, y \oplus G(x), {-}\rangle\,. \end{aligned}$$

This is the same as a conditional evaluation map which queries H if $b = 0$ and G if $b = 1$, with a Hadamard transform before and after.

Let ψ_H resp. ψ_G be the final states of $\mathcal{A}^H$ resp. $\mathcal{A}^G$. The final state of $\mathcal{B}_0^{H,G}$ is

$$\frac{\psi_H \otimes |{+}\rangle + \psi_G \otimes |{-}\rangle}{\sqrt{2}} = \frac{1}{2} \cdot \begin{pmatrix} (|\psi_H\rangle + |\psi_G\rangle) \otimes |0\rangle \\ +(|\psi_H\rangle - |\psi_G\rangle) \otimes |1\rangle \end{pmatrix}.$$

Suppose we measure b in the computational basis. This commutes with the final measurement of $\mathcal{A}$'s state. Then, we will measure 1 with probability $\epsilon := \| |\psi_H\rangle - |\psi_G\rangle \|^2/4$, and hence,

$$\| |\psi_H\rangle - |\psi_G\rangle \| = 2\sqrt{\epsilon}.$$

This $2\sqrt{\epsilon}$ is the claimed probability bound, but we still need a way to extract w^*. The full algorithm $\mathcal{B}^{H,G}$ is the same as $\mathcal{B}_0$, but with a different final measurement and another auxiliary register $w \in \{0,1\}^n$ (i.e. a register that can represent elements of w). The w register is initialized to 0.

We will ensure that except during queries, (b, w) will always be in the state $(0, 0)$ or $(1, w^*)$. More formally, let T_w operate on the b and w registers by $T_w(|b, w\rangle) := |b, w \oplus (b \cdot w^*)\rangle$. Therefore T_w swaps $(1, 0)$ with $(1, w^*)$. We will ensure that if at some step in the computation the state of $\mathcal{B}_0^{H,G}$ is ψ, then during the same step the state of $\mathcal{B}^{H,G}$ is $T_w(\psi \otimes |0\rangle)$.

Since $\mathcal{B}_0^{H,G}$ replaces the oracle queries with U, $\mathcal{B}^{H,G}$ should replace them with $U_w := T_w \circ U \circ T_w^\dagger$. (This gives the desired result because T_w commutes with all the steps of $\mathcal{A}$ except for the oracle queries.) To do this, let

$$T_f(|x, y, b, w\rangle) := |x, y, b, w \oplus (b \cdot f(x))\rangle .$$

Then $\mathcal{B}^{H,G}$ replaces $\mathcal{A}$'s oracle queries with

$$U_f := T_f \circ U \circ T_f^\dagger .$$

In fact, $U_f = U_w$. On the subspace where $x \in S$, we have $f(x) = w^*$ by assumption. Therefore $T_f = T_w$ and $U_f = U_w$. On the orthogonal subspace where $x \notin S$, we have $G(x) = H(x)$, so the operation U does not depend on b or w. Therefore on that subspace, U commutes with T_f and T_w, so that $U_f = U = U_w$. In sum, $U_f = U_w$ is an efficient implementation of the oracle by $\mathcal{B}^{H,G}$.

When $\mathcal{A}$ completes, $\mathcal{B}^{H,G}$ measures (b, w) in the computational basis. With probability ϵ it measures $(1, w^*)$, in which case it outputs w^*. Otherwise it measures $(0, 0)$, in which case it outputs $\perp$.

By [AHU18] Lemmas 3 and 4,

$$\left|\Pr[\mathcal{A}^H \to 1] - \Pr[\mathcal{A}^G \to 1]\right| \le \| |\psi_0\rangle - |\psi_1\rangle \| \le 2\sqrt{\Pr[\mathcal{B}^{H,G} \to w^*]}$$

and likewise

$$\left|\sqrt{\Pr[\mathcal{A}^H \to 1]} - \sqrt{\Pr[\mathcal{A}^G \to 1]}\right| \le \| |\psi_0\rangle - |\psi_1\rangle \| \le 2\sqrt{\Pr[\mathcal{B}^{H,G} \to w^*]}.$$

This completes the proof for unitary adversaries $\mathcal{A}$ with a fixed H and G.

For non-unitary adversaries and for random distributions of H, G, we instead end in a mixture of states Ψ_0 resp. Ψ_1, for which Euclidean distance is not appropriate but the Bures distance [NC00] is. By monotonicity and joint concavity of fidelity (exactly as in [AHU18] Lemma 6 and 9), the same bound holds for the Bures distance:

$$\left|\Pr[\mathcal{A}^H \to 1] - \Pr[\mathcal{A}^G \to 1]\right| \le B(\Psi_0, \Psi_1) \le 2\sqrt{\Pr[\mathcal{B}^{H,G} \to w^*]}$$

and likewise

$$\left|\sqrt{\Pr[\mathcal{A}^H \to 1]} - \sqrt{\Pr[\mathcal{A}^G \to 1]}\right| \le B(\Psi_0, \Psi_1) \le 2\sqrt{\Pr[\mathcal{B}^{H,G} \to w^*]}.$$

This completes the proof in the general case. □

C Proof of Theorem 1

Theorem 1. *Let* P *be an rPKE with messages in* $\mathcal{M}$ *and random coins in* $\mathcal{R}$. *Let* $G : \mathcal{M} \to \mathcal{R}$ *be a quantum-accessible random oracle. Let* $\mathcal{A}$ *be an* OW-CPA *adversary against* $\mathsf{P}' := T(\mathsf{P}, G)$. *Suppose that* $\mathcal{A}$ *queries* G *at most* q *times with depth at most* d.

Then we can construct an IND-CPA *adversary* $\mathcal{B}$ *against* P, *running in about the same time and resources as* $\mathcal{A}$, *such that*

$$\mathrm{Adv}_{\mathsf{P}'}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) \le (d+2) \cdot \left(\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}) + \frac{8(q+1)}{|\mathcal{M}|}\right).$$

Proof. Let $\mathcal{A}_1$ be the same as $\mathcal{A}$, except that at the end after choosing an output m, it computes and discards $G(m)$. Therefore it makes at most $q+1$ queries at depth at most $d+1$. This is a formality so that returning the correct m will count as a Find (see Definition 1]) later in the proof. Clearly the two algorithms $\mathcal{A}$ and $\mathcal{A}_1$ have the same $\mathsf{OW\text{-}CPA}$-advantage against P'.

We actually show a slightly stronger result, constructing an $\mathsf{IND\text{-}KPA}$ adversary $\mathcal{B}$. The $\mathsf{IND\text{-}KPA}$ adversary $\mathcal{B}$ (cf. Definition 9) is given the tuple

$$(\mathrm{pk}, m_0, m_1, c) \;\text{ where }\; c = \mathrm{Encr}(\mathrm{pk}, m_b; r).$$

It wants to determine whether $b=0$ or $b=1$. The algorithm $\mathcal{B}$ creates a fresh random oracle G and runs

$$\mathcal{A}_1^{G\setminus\{m_0,m_1\}}(\mathrm{pk}, c).$$

Suppose Find occurs, i.e., a query $x \in \{m_0, m_1\}$ was asked by $\mathcal{A}$ to its oracle G. Then $\mathcal{B}$ measures whether the query x was m_0 or m_1, and returns the corresponding b. If Find does not occur, or if Find occurs but both m_0 and m_1 were queried, then $\mathcal{B}$ guesses b at random.

Let G' be the oracle such that $G'(m_b) = r$ gives the encryption coins used to encrypt m_b, but $G'(m) = G(m)$ for all other messages m. G' is unknown to $\mathcal{B}$, but we can still analyze $\mathcal{A}$'s behavior when run with G' instead of G.

By construction, $\mathcal{A}_1^{G'\setminus\{m_0,m_1\}}$ cannot return m_b without causing Find. Hence,

$$\begin{aligned}
\sqrt{\mathrm{Adv}_{\mathsf{P}'}^{\mathsf{OW\text{-}CPA}}(\mathcal{A})} &= \sqrt{\Pr[\mathcal{A}^{G'} \to m_b]} \\
&= \left|\sqrt{\Pr\left[\mathcal{A}_1^{G'} \to m_b\right]}\right| - \underbrace{\sqrt{\Pr\left[\mathcal{A}_1^{G'\setminus\{m_0,m_1\}} \to m_b \wedge \neg\mathsf{Find}\right]}}_{=\,0} \\
&\overset{\text{Lem. 3}}{\leq} \sqrt{(d+2)\cdot \Pr\left[\mathsf{Find} : \mathcal{A}_1^{G'\setminus\{m_0,m_1\}}\right]}.
\end{aligned}$$

Squaring both sides,

$$\begin{aligned}
\mathrm{Adv}_{\mathsf{P}'}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) &\leq (d+2)\cdot \Pr\left[\mathsf{Find} : \mathcal{A}_1^{G'\setminus\{m_0,m_1\}}\right] \\
&\overset{\text{Lem. 2}}{=} (d+2)\cdot \Pr\left[\mathsf{Find} : \mathcal{A}_1^{G\setminus\{m_0,m_1\}}\right] \\
&= (d+2)\cdot \Pr\left[\mathsf{Find} : \mathcal{B}\right].
\end{aligned}$$

Now decompose Find as $\mathsf{Find}_b \vee \mathsf{Find}_{\neg b}$, where the former event means that Find occurs and m_b is measured, and latter means that Find occurs and $m_{\neg b}$ is measured. They can both occur if $\mathcal{A}$ makes multiple queries simultaneously, but $\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}KPA}}(\mathcal{B}) = |\Pr[\mathsf{Find}_b] - \Pr[\mathsf{Find}_{\neg b}]|$ regardless.

Moreover, since $\mathcal{B}$ measures m whenever Find occurs, we can view $G\setminus\{m_0, m_1\}$ as $G''\setminus\{m_{\neg b}\} := (G\setminus\{m_b\})\setminus\{m_{\neg b}\}$. Since $\mathcal{A}$ has no information about $m_{\neg b}$ except from puncturing, it holds for any m that

$$\Pr\left[m \in \{m_{\neg b}\} \;: \mathcal{A}^{G''}\right] = 1/|\mathcal{M}| =: \epsilon.$$

By (the second statement in) Lemma 4, we have

$$\Pr[\mathsf{Find}_{\neg b} : \mathcal{B}] \leq 4(q+1)\epsilon = \frac{4(q+1)}{|\mathcal{M}|}.$$

Hence,

$$\begin{aligned}\mathrm{Adv}_{\mathsf{P}'}^{\mathsf{IND\text{-}KPA}}(\mathcal{B}) &= \left|\Pr\left[\mathsf{Find}_b : \mathcal{B}\right] - \Pr\left[\mathsf{Find}_{\neg b:\mathcal{B}}\right]\right| \\ &\geq \Pr\left[\mathsf{Find} : \mathcal{B}\right] - 2\Pr\left[\mathsf{Find}_{\neg b} : \mathcal{B}\right] \\ &\geq \Pr\left[\mathsf{Find} : \mathcal{B}\right] - 8(q+1)/|\mathcal{M}|.\end{aligned}$$

Taking into account that $\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}KPA}}(\mathcal{B}) \leq \mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B})$ and combining these results gives

$$\mathrm{Adv}_{\mathsf{P}'}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) \leq (d+2) \cdot \left(\mathrm{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}) + \frac{8(q+1)}{|\mathcal{M}|}\right)$$

as claimed. □

IND vs. OW. Our $\mathcal{B}$ is a distinguishing adversary, not a one-way adversary. The reason is that $\mathcal{A}$ can check whether a given m is the challenge message, but if P is semantically secure then $\mathcal{B}$ cannot check this. Instead $\mathcal{B}$ would have to pick a random query to measure, which still works using Lemma 1, but with an additional factor of q tightness loss. That is, the one-way problem is potentially harder for a randomized encryption scheme than for a deterministic one. The authors discussed using a new "one-way with confirmation oracle" security game to more tightly capture the OW vs. IND tradeoff, but decided that it is simpler to just reduce to IND-CPA.

We also note that ordinarily distinguishing adversaries are much harder to amplify than one-way adversaries, but $\mathcal{B}$ is constructed to either output with relative certainty if Find, or to fail and guess at random. This means that its advantage will still be high in the Micciancio-Walter notion of cryptographic advantage [MW18]. It is likely that the $8(q+1)/|\mathcal{M}|$ could be reduced to a $4(q+1)/|\mathcal{M}|$ without this requirement.

D Why encryption is usually injective for LWE

Here we outline why we expect $\mathrm{Encr}(\mathrm{pk}, \cdot)$ to be an injective function for the overwhelming majority of public keys pk in a PKE based on Learning with Errors (LWE). Consider a typical LWE PKE, where the public key has the form $(A, S = sA + e)$ where s and e are small of dimension n, and ciphertexts have the form $(As' + e', \lceil Xs' + e' + \mathrm{encode}(m)\rfloor)$. Encryption will fail to be injective for some G if there are $(s_0', e_0') = G(m_0)$ and $(s_1', e_1') = G(m_1)$ such that

$$As_0' + e_0' = As_1' + e_1' \quad \text{and} \quad \lceil Xs_0' + e_0' + \mathrm{encode}(m_0)\rfloor = \lceil Xs_1' + e_1' + \mathrm{encode}(m_1)\rfloor .$$

For correctness, the rounded component is always larger than the message space, and is generally larger than $|M|^2$. The unrounded component has size at least

q^n which is larger still. The function $(s_0', e_0') \to As_0' + e_0'$ is almost a universal hash unless s_0' has large nullity, which is highly unlikely for any secure PKE. So the probability of collision with fewer than $|M|^2$ message pairs is not much bigger than q^{-n}, which is negligible.

E Proof of Lemma 6

To prove Lemma 6, we first show a result about Bernoulli variables.

Lemma 7. *Let $\{e_i : 1 \leq i \leq n\}$ be a collection of n independent Bernoulli variables. Let $\delta := \max \ \Pr[e_i]$, and for each integer j let $p_j := \Pr\left[\sum_{i=1}^n e_i = j\right]$. Then $p_1 \leq \sqrt{3p_0p_2 + \delta^2} \leq \sqrt{3p_2} + \delta$.*

Proof. Let $\epsilon_i := \Pr[e_i = 1]$, and without loss of generality let

$$\delta = \epsilon_1 \geq \epsilon_2 \geq \ldots \geq \epsilon_n$$

be given in descending order. Then

$$p_0 = \prod_{i=1}^n (1-\epsilon_i), \quad p_1 = p_0 \cdot \left(\sum_{i=1}^n \frac{\epsilon_i}{1-\epsilon_i}\right), \quad p_2 = p_0 \cdot \left(\sum_{i>j} \frac{\epsilon_i\epsilon_j}{(1-\epsilon_i)(1-\epsilon_j)}\right)$$

so that

$$\begin{aligned}
p_1^2 - 3p_0p_2 &= p_0^2 \cdot \left(\sum_{i,j=1}^n \frac{\epsilon_i\epsilon_j}{(1-\epsilon_i)(1-\epsilon_j)} - 3\sum_{i>j} \frac{\epsilon_i\epsilon_j}{(1-\epsilon_i)(1-\epsilon_j)}\right) \\
&= p_0^2 \cdot \left(\sum_{i=1}^n \frac{\epsilon_i^2}{(1-\epsilon_i)^2} - \sum_{i>j} \frac{\epsilon_i\epsilon_j}{(1-\epsilon_i)(1-\epsilon_j)}\right) \\
&\geq p_0^2 \cdot \left(\sum_{i=1}^n \frac{\epsilon_i^2}{(1-\epsilon_i)^2} - \sum_{i=j+1}^{n} \frac{\epsilon_i\epsilon_j}{(1-\epsilon_i)(1-\epsilon_j)}\right) \\
&\geq p_0^2 \cdot \left(\sum_{i=1}^n \frac{\epsilon_i^2}{(1-\epsilon_i)^2} - \sum_{i=2}^n \frac{\epsilon_i^2}{(1-\epsilon_i)^2}\right) \\
&= p_0^2 \cdot \frac{\epsilon_1^2}{(1-\epsilon_1)^2} \leq \delta^2.
\end{aligned}$$

Hence, $p_1^2 - 3p_0p_2 \leq \delta^2$ and $p_1 \leq \sqrt{3p_0p_2 + \delta^2}$ as claimed. □

We are now ready to prove Lemma 6.

Lemma 6. *Let $\mathsf{P} = (\mathrm{Keygen}, \mathrm{Encr}, \mathrm{Decr})$ be a δ-correct, ϵ-injective rPKE with messages in $\mathcal{M}$ and randomness in $\mathcal{R}$. Let $G : \mathcal{M} \to \mathcal{R}$ be a random oracle, so that $T(\mathsf{P}, G) := (\mathrm{Keygen}, \mathrm{Encr}_1, \mathrm{Decr})$ is a derandomized version of P. Let $\mathcal{A}$ be* *a* FFC *adversary against* $T(\mathsf{P}, G)$ *which makes at most* q *queries at depth* d *to* G *and returns a list of at most* q_{dec} *ciphertexts. Then*

$$\text{Adv}^{\mathsf{FFC}}_{T(\mathsf{P},G)}(\mathcal{A}) \leq ((4d+1)\delta + \sqrt{3\epsilon}) \cdot (q + q_{\text{dec}}) + \epsilon.$$

Proof. Essentially, the idea is at follows: the adversary gets an advantage of about $4dq\delta$ from querying G in search of failing ciphertexts, and at most $q_{\text{dec}}(\delta + 3\sqrt{\epsilon})$ from guessing blindly. The latter term comes from considering ways that some blind guess could be a failing ciphertext: if it is the encryption of one message, then δ is large, and if it is possibly the encryption of more than one message (e.g., as a general "encryption failed" output), then ϵ is large. We will formalize this in what follows.

Generate a keypair $(\text{pk}, \text{sk}) \leftarrow \text{Keygen}()$ and oracle $G \overset{\$}{\leftarrow} \mathcal{R}^{\mathcal{M}}$. Let

$$Y_{\text{m}} := \{r : \text{Decr}(\text{sk}, \text{Encr}(\text{pk}, \text{m}, r)) = \text{m}\}$$

be the set of coins such that decryption of m will succeed. Let $G'(\text{m}) := G(\text{m})$ if $G(\text{m}) \in Y_{\text{m}}$, $G'(\text{m}) \overset{\$}{\leftarrow} \mathcal{R}$ if $Y_{\text{m}} = \emptyset$, and $G'(\text{m}) \overset{\$}{\leftarrow} Y_{\text{m}}$ otherwise. Thus G' is uniformly random in the space $\mathcal{G}$ of oracles where decryption succeeds if possible. Moreover, G' is independent of the behavior of messages and ciphertexts for $T(\mathsf{P}, G)$ which do not decrypt correctly.

Now, fix (sk, pk) and G' and let

$$\delta' := \max_{\text{m} \in \mathcal{M}} \Pr[\text{Decr}(\text{sk}, \text{Encr}(\text{pk}, \text{m})) \neq \text{m}]$$

be the failure probability for this keypair. Let $\mathsf{DblFail}$ be the event that some ciphertext c is the encryption of two messages m_1 and m_2 such that $\text{Decr}(\text{sk}, c) \notin \{\text{m}_1, \text{m}_2\}$. We define $\epsilon' := \Pr[\mathsf{DblFail}]$. Both δ' and ϵ' are independent of G'. In addition, let Fail be the event that $\mathcal{A}$ wins the FFC game (see Definition 3), and $\mathsf{Ev} := \mathsf{Fail} \wedge \neg\mathsf{DblFail}$. By Lemma 1, it holds that

$$\left| \sqrt{\Pr[\mathsf{Ev} : \mathcal{A}^{G}(\text{pk})]} - \sqrt{\Pr[\mathsf{Ev} : \mathcal{A}^{G'}(\text{pk})]} \right| \leq 2d\sqrt{P_{\text{guess}}}.$$

Since, conditioned on G', $G(\text{m}) \neq G'(\text{m})$ at each m with probability at most δ' and there are q/d guesses (in expectation), it holds furthermore that

$$2d\sqrt{P_{\text{guess}}} \leq \sqrt{4d^2 P_{\text{guess}}} \leq \sqrt{4dq\delta'}.$$

Next we define for a ciphertext c,

$$p_1(c) := \Pr[\exists \text{ unique m} \in \mathcal{M} : c = \text{Encr}(\text{pk}, \text{m}, G(\text{m})) \wedge \text{Decr}(\text{sk}, c) \neq \text{m}].$$

(It is important to note that if m exists but is not unique, then $\mathsf{DblFail}$ occurs.) Furthermore, let $p_1 := \max_c(p_1(c))$. Since $p_1(c)$ and p_1 are independent of G', we have

$$\Pr[\mathsf{Ev} : \mathcal{A}^{G'}(\text{pk})] \leq q_{\text{dec}} \cdot p_1.$$

By Lemma 7, $p_1 \leq \delta' + \sqrt{3\epsilon'}$. Plugging this in and applying the Cauchy-Schwarz corollary $\sqrt{ab} + \sqrt{cd} \leq \sqrt{(a+c)(b+d)}$ gives

$$\begin{aligned}\sqrt{\Pr[\mathsf{Ev} : \mathcal{A}^G(\mathrm{pk})]} &\leq \sqrt{4dq\delta'} + \sqrt{q_{\mathrm{dec}} \cdot (\delta' + \sqrt{3\epsilon'})} \\ &\leq \sqrt{((4d+1)\delta' + \sqrt{3\epsilon'}) \cdot (q + q_{\mathrm{dec}})}.\end{aligned}$$

Finally, by definition of correctness and injectivity of an rPKE (see Definition 5 and 6, respectively), it holds that $\delta = \mathrm{E}\left[\delta'\right]$ and $\epsilon \leq \mathrm{E}\left[\epsilon'\right]$. By Jensen's inequality, it holds furthermore that $\sqrt{\epsilon} \leq \mathrm{E}\left[\sqrt{\epsilon'}\right]$. Hence,

$$\begin{aligned}\mathrm{Adv}^{\mathsf{FFC}}_{T(\mathsf{P},G)}(\mathcal{A}) &\leq \mathrm{E}\left[\Pr[\mathsf{Ev} : \mathcal{A}^G(\mathrm{pk})] : (\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Keygen}(); G' \xleftarrow{\$} \mathcal{G}\right] + \epsilon \\ &\leq ((4d+1)\delta + \sqrt{3\epsilon}) \cdot (q + q_{\mathrm{dec}}) + \epsilon\end{aligned}$$

as claimed. □

F Proof of Theorem 4

Theorem 4 (Implicit → explicit with key confirmation). *Let* P *be an* ϵ*-injective dPKE. Consider the KEM* $\mathsf{K}_1 := U_m^{\perp}(C(\mathsf{P}, H_t, \tau), H_s)$ *obtained from* P *applying the* C*-transform with random oracle* $H_t : \mathcal{M} \to \{0,1\}^{\tau}$ *and the* $U_m^{\perp}$*-transform with independent random oracle* $H_s : \mathcal{M} \to \{0,1\}^{\varsigma}$*. Let* $\mathsf{K}_2 := U_m^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ *be the KEM obtained from* P *applying the* $U_m^{\not\perp}$*-transform with random oracle* $H : \mathcal{M} \to \{0,1\}^{\varsigma+\tau}$*.*

If $\mathcal{A}$ *is an* $\mathsf{IND\text{-}CCA}$*-adversary against* K_1 *which makes* q_{dec} *decapsulation queries, then it is also an* $\mathsf{IND\text{-}CCA}$*-adversary against* P_2 *and there is a* PRF*-adversary* $\mathcal{B}$ *against* F *which uses about the same time and resources as* $\mathcal{A}$*, such that:*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{K}_1}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{K}_2}(\mathcal{A}) + \frac{q_{\mathrm{dec}}}{2^{\tau-1}} + 2 \cdot \mathrm{Adv}^{\mathsf{PRF}}_{\mathsf{F}}(\mathcal{B}) + 2\epsilon.$$

Proof. The proof is by a series of games. Let w_i be the probability that $\mathcal{A}$ wins Game i.

Game 0 ($\mathsf{IND\text{-}CCA}$). *This is the* $\mathsf{IND\text{-}CCA}$ *game against* K_1.

Game 1 (Modify decapsulation with random function). *In Game 1, the simulator instantiates a fresh random function* R*, and modifies the decapsulation oracle* D *to the oracle* D' *shown in Figure 8.*

We analyze the difference between D and D' as follows.

- If $c = c^*$, then D' returns $\perp$. If $\mathrm{Encr}(\mathrm{pk}, \cdot)$ is injective, then so does D.
- Otherwise, if $\mathrm{Encr}(\mathrm{pk}, m') = c$, then both D and D' return $H_s(m')$ if $H_t(m') = t$, and $\perp$ otherwise.

```
$D((c,t))$:

if $(c,t) = (c^*, t^*)$: return $\perp$
$m' \leftarrow \mathrm{Decr}(\mathrm{sk}, c)$
if $m' = \perp$:
    return $\perp$
else if $(\mathrm{Encr}(\mathrm{pk}, m'), H_t(m')) \neq (c,t)$:
    return $\perp$
else: return $H_s(m')$
```

```
$D'((c,t))$:

if $c = c^*$: return $\perp$
$m' \leftarrow \mathrm{Decr}(\mathrm{sk}, c)$
if $m' = \perp$: $(k, t') \leftarrow R(c)$
else if $\mathrm{Encr}(\mathrm{pk}, m') \neq c$: $(k, t') \leftarrow R(c)$
else: $(k, t') \leftarrow (H_s(m'), H_t(m'))$
if $t' \neq t$: return $\perp$
else: return $k$
```

Fig. 8. Decapsulation oracles for Game 0 and Game 1.

- Otherwise, D returns $\perp$, and so does D' unless t matches t'. Since R is random and is only used for this purpose, this happens with probability at most $q_{\mathrm{dec}}/2^{\tau}$.

The difference in $\mathcal{A}$'s view is bound by the probability that D' acts different than D. So overall $|w_1 - w_0| \leq q_{\mathrm{dec}}/2^{\tau} + \epsilon$.

Game 2 (Use PRF instead of R). *In Game 2, the simulator replaces $R(c)$ by $\mathsf{F}(\mathrm{prfk}, \cdot)$ with a random prf key* prfk.

The difference in probability of any adversary $\mathcal{A}$ between winning Game 1 and Game 2 is exactly the PRF-advantage of an adversary $\mathcal{B}$ that works exactly as in the anlaysis of Game 1 in the proof of Theorem 2. Hence it holds that

$$|w_2 - w_1| = \mathrm{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B})$$

Game 3 (Redirect to $U_m^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$). *Game 3 is refactored so that it simulates the* $\mathsf{IND\text{-}CCA}$ *experiment for $\mathsf{K}_2 = U_m^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ for the case $b = 0$ (correct key), as follows:*

- *Hash redirection. The oracles H_s resp. H_t used for C resp. $U^{\perp}$ are redirected to the first ς resp. last t bits of the hash function H of $U_m^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$.*
- *The simulator creates a challenge ciphertext c with shared secret (k_s, k_t). This is given to $\mathcal{A}$ as a challenge ciphertext (c, k_t) with shared secret $k = k_s$.*
- *The decapsulation oracle D' from Game 2 is changed to use the $U_m^{\not\perp}$ decapsulation oracle internally, as shown in Figure 9. It is called D''.*

All these steps do not change $\mathcal{A}$'s view compared to Game 2, so $w_3 = w_2$.

Game 4 (Redirect to $U_m^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ with random keys). *Game 4 is the same as Game 3 except that it now simulates the $b = 1$ (random key) case of the* $\mathsf{IND\text{-}CCA}$ *experiment against K_2, i.e., it always sets $k \xleftarrow{\$} \{0,1\}^{\varsigma}$.*

Distinguishing Game 3 from Game 4 is exactly the $\mathsf{IND\text{-}CCA}$ experiment for K_2. Hence,

$$|w_4 - w_3| = \mathrm{Adv}_{\mathsf{K}_2}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$$

```
$D''((c,t))$:

$r \leftarrow \mathrm{Decaps}_{\mathsf{K}_2}(c)$
if $r = \bot$: return $\bot$
parse $r$ as $(k, t')$
if $t' \neq t$: return $\bot$
else: return $k$
```

Fig. 9. Decapsulation oracle for Game 3.

Note that $\mathcal{A}$'s goal is still to learn the bit b of the $\mathsf{IND\text{-}CCA}$ game against K_1. But now the shared secret k is always random, and thereby independent of b. Hence, the adversary has no information about b and so $w_4 = \frac{1}{2}$.

Summing up the differences in winning probability from all the games we get

$$\left| w_0 - \frac{1}{2} \right| \leq \mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{K}_2}(\mathcal{A}) + \frac{q_{\mathrm{dec}}}{2^{\tau}} + \mathrm{Adv}^{\mathsf{PRF}}_{\mathsf{F}}(\mathcal{B})$$

and $\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{K}_1}(\mathcal{A})$ is at most twice this value. This completes the proof. □