# Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform

Kathrin Hövelmanns[1], Andreas Hülsing[1], and Christian Majenz[2]

[1] Eindhoven University of Technology, The Netherlands
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark
authors-fo-failure@huelsing.net

**Abstract.** In known security reductions for the Fujisaki-Okamoto transformation, decryption failures are handled via a reduction solving the rather unnatural task of finding failing plaintexts *given the private key*, resulting in a Grover search bound. Moreover, they require an implicit rejection mechanism for invalid ciphertexts to achieve a reasonable security bound in the QROM. We present a reduction that has neither of these deficiencies: We introduce two security games related to finding decryption failures, one capturing the *computationally hard* task of *using the public key* to find a decryption failure, and one capturing the *statistically hard* task of searching the random oracle for *key-independent* failures like, e.g., large randomness. As a result, our security bounds in the QROM are tighter than previous ones with respect to the generic random oracle search attacks: The attacker can only partially compute the search predicate, namely for said key-independent failures. In addition, our entire reduction works for the explicit-reject variant of the transformation and improves significantly over all of its known reductions. Besides being the more natural variant of the transformation, security of the explicit reject mechanism is also relevant for side channel attack resilience of the implicit-rejection variant. Along the way, we prove several technical results characterizing preimage extraction and certain search tasks in the QROM that might be of independent interest.

**Keywords:** Public-key encryption, post-quantum security, QROM, Fujisaki-Okamoto transformation, decryption failures, NIST

## 1 Introduction

The Fujisaki-Okamoto (FO) transform [FO99, FO13] is a well known transformation that combines a weakly secure public-key encryption scheme and a weakly secure secret-key encryption scheme into an IND-CCA secure public-key encryption scheme in the random oracle model. Dent [Den03, Table 5] gave an adoption

A.H. was supported by an NWO VIDI grant (Project No. VI.Vidi.193.066). C.M. was supported by a NWO VENI grant (Project No. VI.Veni.192.159).

for the setting of key-encapsulation. This adoption for key encapsulation mechanisms (KEM) is now the de-facto standard to build secure KEMs. In particular, it was used in virtually all KEM submissions to the NIST PQC standardisation process [NIS17]. In the context of post-quantum security, however, two novel issues surfaced: First, many of the PKE schemes being transformed into KEM are not perfectly correct, i.e., they sometimes fail to decrypt a ciphertext to its plaintext. Second, security proofs have to be done in the quantum-accessible random oracle model (QROM) to be applicable to quantum attackers.

Both problems were tackled in [HHK17] and a long sequence of follow-up works (among others [SXY18, JZC+18, BHH+19, HKSU20, KSS+20]). While these works made great progress towards achieving tighter reductions in the QROM, the treatment of decryption failures did not improve significantly. In this work, we make significant progress on the treatment of decryption failures. Along the way, we obtain several additional results relevant on their own.

An additional quirk of existing QROM reductions for the FO transform is that they require an *implicit rejection* variant, where pseudorandom session keys are returned instead of reporting decapsulation errors, to avoid extreme reduction losses. (The only known concrete bound [DFMS21] for Dent's variant is much weaker then those known for the implicit rejection variant.)

**The Fujisaki-Okamoto transformation.** We recall the FO transformation for KEM as introduced in [Den03, Table 5] and revisited by [HHK17], there called $\mathsf{FO}^{\perp}_m$. $\mathsf{FO}^{\perp}_m$ constructs a KEM from a public-key encryption scheme $\mathsf{PKE}$, and the overall transformation $\mathsf{FO}^{\perp}_m$ can be described by first modifying $\mathsf{PKE}$ to obtain a deterministic scheme $\mathsf{PKE}^{\mathsf{G}}$, and then applying a PKE-to-KEM transformation (called $\mathsf{U}^{\perp}_m$ in [HHK17]) to $\mathsf{PKE}^{\mathsf{G}}$:

Modified scheme $\mathsf{PKE}^{\mathsf{G}}$. Starting from $\mathsf{PKE}$ and a hash function $\mathsf{G}$, deterministic encryption scheme $\mathsf{PKE}^{\mathsf{G}}$ is built by letting $\mathsf{Enc}^{\mathsf{G}}$ encrypt messages $m$ according to the encryption algorithm $\mathsf{Enc}$ of $\mathsf{PKE}$, but using the hash value $\mathsf{G}(m)$ as the random coins for $\mathsf{Enc}$: $\mathsf{Enc}^{\mathsf{G}}(pk, m) := \mathsf{Enc}(pk, m; \mathsf{G}(m))$. $\mathsf{Dec}^{\mathsf{G}}$ uses the decryption algorithm $\mathsf{Dec}$ of $\mathsf{PKE}$ to decrypt a ciphertext $c$ to obtain $m'$, and rejects by returning $\perp$ if $c$ fails to decrypt or $m'$ fails to encrypt back to $c$.

PKE-to-KEM transformation $\mathsf{U}^{\perp}_m$. Starting from a deterministic encryption scheme $\mathsf{PKE}$' and a hash function $\mathsf{H}$, key encapsulation algorithm $\mathsf{KEM}^{\perp}_m := \mathsf{U}^{\perp}_m[\mathsf{PKE}', \mathsf{H}]$ is built by letting $\mathsf{Encaps}(pk) := (c := \mathsf{Enc}'(pk, m), K := \mathsf{H}(m))$, where $m$ is picked at random from the message space. Decapsulation will return $K := \mathsf{H}(m)$ unless $c$ fails to decrypt, in which case it returns failure symbol $\perp$.

Combined PKE-to-KEM transformation $\mathsf{FO}^{\perp}_m$. The 'full FO' transformation $\mathsf{FO}^{\perp}_m$ is defined by taking $\mathsf{PKE}$ and hash functions $\mathsf{G}$ and $\mathsf{H}$, and defining $\mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\perp}_m[\mathsf{PKE}^{\mathsf{G}}, \mathsf{H}]$. While there exists a plethora of variants that differ from $\mathsf{FO}^{\perp}_m$, it was proven [BHH+19] that security of these variants is either equivalent to or implied by security of $\mathsf{FO}^{\perp}_m$.

**The role of correctness errors in security proofs for FO.** Correctness errors play a role during the proof that an FO-transformed KEM is IND-CCA

secure: To tackle the CCA part, it is necessary to simulate the decapsulation oracle oDecaps without the secret key, meaning the plaintext has to be obtained via strategies different from decrypting. While different strategies for this exist in both ROM and QROM, they all have in common that the obtained plaintext is rather a plaintext that encrypts to the queried ciphertext (a "ciphertext preimage") than the decryption. Consequently, the simulation fails to recognise *failing ciphertexts*, i.e., ciphertexts for which decryption results in a plaintext different from the ciphertext preimage (or even in $\perp$), and will in this case behave differently from oDecaps. Hence, the simulations are distinguishable from oDecaps if the attacker can craft such failing ciphertexts.

The approach chosen by [HHK17] was to show that the distinguishing advantage between the two cases can be bounded by the advantage in a game COR. Game COR (defined in [HHK17]) provides an adversary with a key pair (including the secret key) and asks to return a *failing message*, i.e., a message that encrypts to a failing ciphertext, for the derandomized scheme $\mathsf{PKE}^{\mathsf{G}}$. [HHK17] further bounded the maximal advantage in game COR for $\mathsf{PKE}^{\mathsf{G}}$ in terms of a statistical worst-case quantity $\delta_{\mathrm{wc}}$ of PKE, which is the expected maximum probability for plaintexts to cause a decryption failure, with the expectation being taken over the key pair. This results in a typical search bound as the adversary can use the secret key to check if a ciphertext fails. In the QROM, the resulting bound is therefore $8q^2\delta_{\mathrm{wc}}$, $q$ being the number of queries to G.[3]

Intuitively, this notion suffers from two related unnatural features:

– First, it is unnatural to provide adversaries with the secret key, as long as the scheme has at least some basic security.[4] In particular, this observation applies to adversaries tasked with finding failing plaintexts, which is not a mere issue of aesthetics: If the secret key is given to the adversary, an analysis of this bound can't make use of computational assumptions without becoming heuristic.[5]

– Second, it is unnatural that the bound contains a Grover-like search term with regard to $\delta_{\mathrm{wc}}$: As IND-CCA adversaries don't have access to the secret key, they can only check if ciphertexts fail via their *classical* CCA oracle, which should render a Grover search impossible. Furthermore, in ROM and QROM, it should be the (usually much smaller) number of CCA queries that limits the adversary's ability to search, not the number of random oracle queries. Hence this bound seems overly conservative.

While follow-up works have used different games in place of COR to deal with decryption errors, all result in the same quantum search bound in terms of $\delta_{\mathrm{wc}}$.

[3] Some publications (e.g., [JZC+18]) use the bound $2q \cdot \sqrt{\delta_{\mathrm{wc}}}$, it is however straightforward to verify that the bound above can be achieved by using [HKSU20, Lemma 2.9] as a drop-in replacement. Note that this is indeed a quadratic improvement unless $4q \cdot \sqrt{\delta_{\mathrm{wc}}} > 1$, in which case the IND-CCA bound is meaningless, anyways.

[4] Schemes that allow for a key recovery attack serve as pathological examples why this argument does not hold in generality.

[5] An example we happen to be aware of is the analysis of the correctness error bound of Kyber [BDK+18].

**Main contribution.** Our main contribution is a new security reduction for the FO transformation that improves over existing ones in two ways.

Decryption failures. We introduce a family of new security games, the *Find Failing Plaintext* (FFP) games. These provide a much more natural framework for dealing with decryption errors in the FO transformation, and it is the novel structure of our reduction that allows their usage. Two important members of the FFP family are as follows: The first one, *Find Failing Plaintext that is Non-Generic* (FFP-NG), gives a public key to the adversary and asks it to find a message that triggers a decryption failure more likely with respect to this key pair than with respect to an independent key pair. The second one, *Find Failing Plaintext with No Key* (FFP-NK), tasks an adversary with producing a message that triggers a decryption failure with respect to an independently sampled key pair, without providing any key to the adversary. As summarised in Fig. 1, we provide a reduction from FFP-NG and passive security of PKE together with FFP-NK for $\mathsf{PKE}^{\mathsf{G}}$ to IND-CCA security of the FO-transformed of PKE. This new reduction structure avoids both unnatural features mentioned above:
– None of the two failure-related games FFP-NG and FFP-NK provide the adversary with the secret key. In particular, we show how to bound an adversary's advantage in game FFP-NK in terms of $\delta_{\mathrm{ik}}$, the worst-case decryption error rate when the message is picked *independently of the key*, and additional related statistical parameters . We give two concrete example bounds, one involving the variance based on Chebyshev's inequality and one based on a Gaussian-shaped tail bound. We expect that these "independent-key" statistical parameters can be estimated more conveniently and *without heuristics*, by exploiting the computational assumptions of the PKE scheme at hand.
– Game FFP-NK still allows for a Grover search advantage, but only when searching for messages that are more likely to cause a failure *on average over the key*. This game corresponds, e.g., to the first attempt at finding a failure in attacks like [DVV18, BS20, DRV20]. In the context of the entire security reduction for FO, the advantage in this game is multiplied with the number of decapsulation queries a CCA attacker makes, correctly reflecting the fact that the ability of *identifying* a decryption failure should depend on the CCA oracle and is thus limited. Game FFP-NG defines a property of the underlying PKE scheme, it thus allows to analyze the hardness of finding meaningful decryption failures independently from the hardness of searching a random oracle for them. FFP-NG seems thus more amenable to both security reductions and cryptanalysis.

As a consequence of these features, we expect our reduction to yield much better security bounds that provide non-trivial provable security for real-world parameters.

FO with explicit rejection. Our reduction employs a technique for generalized *preimage extraction* in the QROM that was recently introduced in [DFMS21]. As shown by [DFMS21], this technique is well-suited for proving $\mathsf{FO}^{\perp}_{m}$ secure. We furthermore generalize the one-way to hiding (OWTH) lemma [AHU19] such that it is compatible with the technique from [DFMS21]. OWTH was used to derive the state-of-the-art bounds for implicitly rejecting variants,

and combining the two techniques, we obtain a security bound for $\mathsf{FO}_m^{\perp}$ that is competitive with said state-of-the-art bounds.

QROM TOOLS. To facilitate the above-described reduction, we provide two technical tools that might be of independent interest: Firstly, we generalize the OWTH framework from [AHU19] such that it can be combined with the extractable quantum random oracle simulation from [DFMS21], rendering the two techniques compatible with being used together in the same security reduction. We make crucial use of this possibility to avoid the additional reduction losses that [DFMS21] need to accept to be able to use the plain one-way to hiding framework in juxtaposition with the extractable simulator.

Secondly, we prove query lower bounds for tasks where an algorithm has access to a QRO (or even an extractable simulator thereof) and has to output an input value $x$ which, together with the corresponding oracle output $\mathsf{RO}(x)$, achieves a large value under some figure-of-merit function. We use this technical result to provide the aforementioned bounds for the adversarial advantage in the FFP-NK game, but they might prove of independent interest.

**TL;DR for scheme designers.** Sect. 6 provides concrete bounds for the IND-CCA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. Besides having to analyze the conjectured passive security of PKE, applying the bounds to a concrete scheme PKE requires to analyze the following computational and statistical properties:
– $\gamma$, the spreadness of PKE.
– An upper bound for FFP-NG against PKE.
– Either an upper bound for FFP-NK for $\mathsf{PKE}^{\mathsf{G}}$, in our extended oracle model that allows preimage extractions, or , two statistical values: $\delta_{\mathrm{ik}}$, the worst-case decryption error rate when the message is picked *independently of the key*, and $\sigma_{\delta_{\mathrm{ik}}}$, the maximal variance of $\delta_{\mathrm{ik}}$.

**Acknowledgements.** We would like to thank Dominique Unruh for valuable discussions about the semi-classical one-way to hiding lemma and Manuel Barbosa for pointing out the use of heuristics in bounds for delta.

## 2 ROM reduction

This section substantiates the upper half of Fig. 1 in the ROM. The first step of common security reductions for the FO transformation consists of simulating the decapsulation oracle without using the secret key. This simulation allows transforming an IND-CCA-KEM-adversary $\mathcal{A}$ against $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ into an IND-CPA-KEM-adversary $\tilde{\mathcal{A}}$ against the same $\mathsf{KEM}_m^{\perp}$. The oracle simulation, however, will not accurately simulate the behaviour of Decaps for ciphertexts that trigger decryption errors. We will show that from an adversary capable of distinguishing between the real decapsulation oracle and its simulation, we can construct an adversary $\mathcal{B}$ that is able to extract failing plaintexts for the derandomised version $\mathsf{PKE}^{\mathsf{G}}$ of PKE. In more detail, we formalise extraction of failing plaintexts as the winning condition of two Find Failing Plaintext (FFP) games, which we formally define in Definition 1 (also see Fig. 2). For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$,

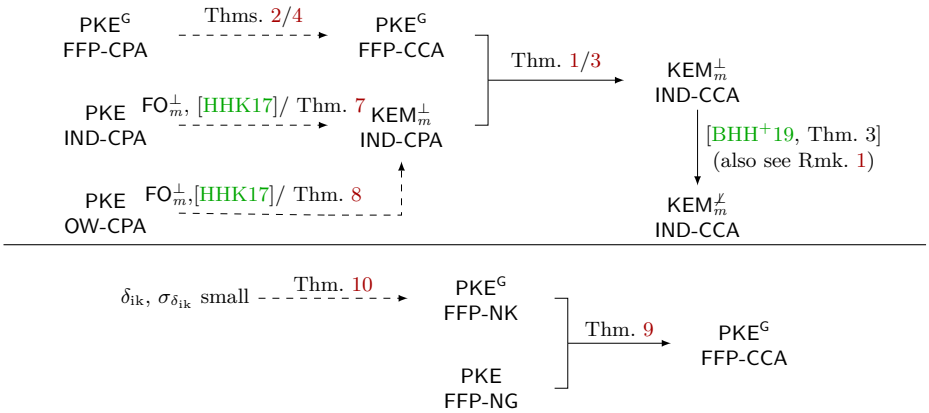


**Fig. 1.** Summary of our results. Top: "Ths. X/Y" indicates that we provide a ROM Thm. X (in Sect. 2) and a QROM Thm. Y (Sect. 4). Bottom: Breaking down FFP-CPA security of $\mathsf{PKE}^{\mathsf{G}}$ (Sect. 5). Solid (dashed) arrows indicate tight (non-tight) reductions in the QROM. Thms. 2 and 4 have comparably mild tightness loss: It is linear in the number of decryption queries. Thms. 7 and 8 are as lossy as previously known ones.

an adversary $\mathcal{B}$ playing the FFP-ATK game for a deterministic encryption scheme PKE gets access to the same oracles as in the respective IND-ATK game, outputs a message $m$, and wins if $\mathsf{Dec}(\mathsf{Enc}(m)) \neq m$. (Here, and in the following, we sometimes omit the arguments $pk$ and $sk$, respectively.) For such messages $m$ we say that $m$ is a *failing plaintext*, or shorter, that $m$ *fails*. The final bounds we obtain are essentially similar to the ones in [HHK17] except for involving a different correctness definition, see the discussion after Remark 1. Game FFP-CCA was already introduced in [BS20], there called COR-ad-CCA.

**Definition 1 (FFP-ATK).** *Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a deterministic public-key encryption scheme. For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$*, we define* FFP-ATK *games as in Fig. 2, where* $\mathsf{O}_{\mathsf{ATK}}$ *is trivial if* $\mathsf{ATK} = \mathsf{CPA}$ *and*

$$\mathsf{O}_{\mathsf{ATK}} := o\textsc{Decrypt} \quad \text{if } \mathsf{ATK} = \mathsf{CCA}.$$

*We define the* FFP-ATK *advantage function of an adversary* $\mathcal{A}$ *against* PKE *as*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}ATK}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr[\mathsf{FFP\text{-}ATK}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] \ .$$

Note that in neither FFP-ATK game, the adversary has access to the secret key. In particular, the FFP-CPA game only differs from the correctness game COR defined in [HHK17] in exactly this fact, as game COR additionally provides the secret key. We note that an adversary winning either FFP-ATK game for a deterministic scheme PKE can be used to win in game COR.

We start by introducing two simulations of the Decaps oracle, o$\textsc{Decaps}'$ and a variant o$\textsc{Decaps}''$ of o$\textsc{Decaps}'$. o$\textsc{Decaps}''$ extracts failing plaintexts from adversarial decapsulation queries, and is simulatable by FFP adversaries with

| **Game** FFP-ATK | oDECRYPT($c$) |
|---|---|
| 01 $(pk, sk) \leftarrow \mathsf{KG}$ | 06 $m := \mathsf{Dec}(sk, c)$ |
| 02 $m \leftarrow \mathcal{A}^{\mathrm{O}_{\mathrm{ATK}},\mathsf{G}}(pk)$ | 07 **return** $m$ |
| 03 $c := \mathsf{Enc}(pk, m)$ | |
| 04 $m' := \mathsf{Dec}(sk, c)$ | |
| 05 **return** $[\![m' \neq m]\!]$ | |

**Fig. 2.** Games FFP-ATK for a deterministic PKE, where $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$. $\mathrm{O}_{\mathsf{ATK}}$ is the decryption oracle present in the respective IND-ATK-KEM game (see Definition 1) and G is a random oracle, provided if it is used in the definition of PKE.

access to the decryption oracle oDECRYPT for $\mathsf{PKE}^{\mathsf{G}}$. Both simulations of the Decaps oracle make use of a list $\mathcal{L}$ of previous queries to G and their respective encryptions. For this to work, we replace G with a modification $\mathsf{G}'$ that keeps track of all issued queries and compiles $\mathcal{L}$. The original Decaps oracle and its simulations are defined in Fig. 3, using the following conventions. For a set of pairs $\mathcal{L} \subset \mathcal{X} \times \mathcal{Y}$, we assume that a total order is chosen on $\mathcal{X}$ and $\mathcal{Y}$. We denote by $\mathcal{L}^{-1}(y)$ the first preimage of $y$. Formally, we define $\mathcal{L}^{-1}(y)$ by setting

$$\mathcal{L}^{-1}(y) := \begin{cases} x & \text{if } (x,y) \in \mathcal{L} \text{ and } x \leq x' \text{ for all } x' \text{ s. th. } (x',y) \in \mathcal{L} \\ \bot & \nexists\, x \text{ s. th. } (x,y) \in \mathcal{L}. \end{cases} \tag{1}$$

The simulation oDECAPS$'$ can, however, only *reverse* encryptions that were already computed by the adversary (with a query to oracle $\mathsf{G}'$) *before* their query to oracle oDECAPS$'$, which is where the spreadness of PKE comes into play: If $\gamma$ is large, it becomes unlikely that the attacker can guess an encryption $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$ without a respective query to G. oDECAPS$'$ will furthermore answer inconsistently if the reversion (in other words, the preimage) of $c$ differs from its decryption, meaning that $c$ belongs to a failing plaintext that can be recognized by the failure-extracting variant oDECAPS$''$.

**Theorem 1.** *Let* PKE *be a (randomised)* PKE *scheme that is $\gamma$-spread, and let* $\mathsf{KEM}_m^{\bot} := \mathsf{FO}_m^{\bot}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. *Let $\mathcal{A}$ be an* IND-CCA-KEM*-adversary (in the ROM) against* $\mathsf{KEM}_m^{\bot}$*, making at most $q_{\mathsf{D}}$ many queries to its decapsulation oracle* oDECAPS. *Then there exist an* IND-CPA-KEM *adversary $\tilde{\mathcal{A}}$ and an* FFP-CCA *adversary $\mathcal{B}$ against* $\mathsf{PKE}^{\mathsf{G}}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\bot}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}_m^{\bot}}\left(\tilde{\mathcal{A}}\right) + \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + q_{\mathsf{D}} \cdot 2^{-\gamma}. \tag{2}$$

*$\tilde{\mathcal{A}}$ makes $q_{\mathsf{G}}$ queries to* G *and $q_{\mathsf{H}} + q_{\mathsf{D}}$ queries to* H*, $\mathcal{B}$ makes $q_{\mathsf{G}}$ queries to* G *and $q_{\mathsf{D}}$ decryption queries, and both adversaries run in about the time of $\mathcal{A}$.*

*Proof.* Let $\mathcal{A}$ be an adversary against $\mathsf{KEM}_m^{\bot}$. We define $\tilde{\mathcal{A}}$ as the IND-CPA-KEM adversary against $\mathsf{KEM}_m^{\bot}$ that runs $b' \leftarrow \mathcal{A}^{\mathsf{G}',\mathsf{H},\mathrm{oDECAPS}'}$ and returns $b'$. We furthermore define our FFP-CCA adversary $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$ as follows: $\mathcal{B}$ runs

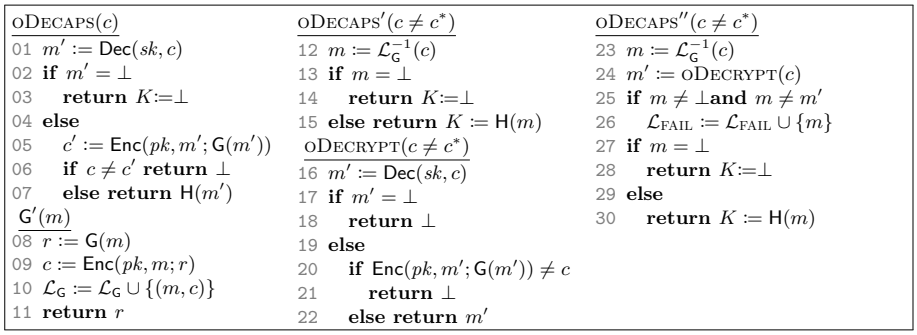


**Fig. 3.** Simulation oDecaps′ of oracle oDecaps for $\mathsf{KEM}_m^\bot$, failing-plaintext-extracting version oDecaps″ of oDecaps′, and decryption oracle oDecrypt for $\mathsf{PKE}^\mathsf{G}$. Oracles oDecaps′ and oDecaps″ use in lines 12 and 23 the notation introduced in Equation (1). $\mathsf{G}'$ only differs from $\mathsf{G}$ by compiling list $\mathcal{L}_\mathsf{G}$ (which was initialized to $\emptyset$).

$\mathcal{A}^{\mathsf{G}', \mathsf{H}, \text{oDecaps}''}$, using its own FFP-CCA oracle oDecrypt to simulate oDecaps″. As soon as oDecaps″ adds a plaintext $m$ to $\mathcal{L}_{\text{FAIL}}$, $\mathcal{B}$ aborts $\mathcal{A}$ and returns $m$. If $\mathcal{A}$ finishes and $\mathcal{L}_{\text{FAIL}}$ is still empty, $\mathcal{B}$ returns $\bot$.

First, we will relate $\mathcal{A}$'s success probability to the one of $\tilde{\mathcal{A}}$. Note that unless $\tilde{\mathcal{A}}$'s simulation oDecaps′ of the decapsulation oracle fails, $\tilde{\mathcal{A}}$ perfectly simulates the game to $\mathcal{A}$ and wins if $\mathcal{A}$ wins. Let DIFF be the event that $\mathcal{A}$ makes a decryption query $c$ such that $\mathsf{Decaps}(sk, c) \neq \text{oDecaps}'(c)$. We bound

$$\begin{aligned}
&\frac{1}{2} + \mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^\bot}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}] = \Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{DIFF}] + \Pr[\mathcal{A} \text{ wins} \wedge \mathsf{DIFF}] \\
&= \Pr\left[\tilde{\mathcal{A}} \text{ wins} \wedge \neg\mathsf{DIFF}\right] + \Pr[\mathcal{A} \text{ wins} \wedge \mathsf{DIFF}] \leq \Pr\left[\tilde{\mathcal{A}} \text{ wins}\right] + \Pr[\mathsf{DIFF}] \\
=&\frac{1}{2} + \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}_m^\bot}\left(\tilde{\mathcal{A}}\right) + \Pr[\mathsf{DIFF}].
\end{aligned}$$

To analyze the probability of event DIFF, we note that it covers several cases:

- Original oracle oDecaps$(c)$ rejects, whereas simulation oDecaps′$(c)$ does not, meaning that $c$ is an encryption belonging to a previous query $m$ to $\mathsf{G}'$, but fails the reencryption check performed by oDecaps$(c)$. Since the latter means that either $m' := \mathsf{Dec}(sk, c) = \bot$ or that $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, this case only occurs if $\mathsf{Dec}(sk, c) \neq m$, meaning $m$ fails.
- Neither oracle rejects, but the return values differ, i.e., $c$ is an encryption belonging to a previous query $m$ to $\mathsf{G}'$, but decrypts to some message $m' \neq m$.
- oDecaps′$(c)$ rejects, whereas oDecaps$(c)$ does not, i.e., while $c$ would pass the reencryption check, its decryption $m$ has not yet been queried to $\mathsf{G}'$.

In either of the former two cases, $\mathsf{G}'$ has been queried on a failing plaintext $m$ and the decapsulation oracle has been queried on its encryption $c$, meaning that the failing plaintext can be found and recognized by $\mathcal{B}$ since $\mathcal{B}$ can use its

own FFP-CCA oracle oDecrypt to simulate oDecaps″. We will denote the last case by GUESS since $\mathcal{A}$ has to find a guess for a ciphertext $c$ that passes the reencryption check, meaning it is indeed of the form $c = \mathsf{Enc}(pk, m; \mathsf{G}'(m))$ for $m := \mathsf{Dec}(sk, c)$, while not having queried $\mathsf{G}'$ on $m$ yet. Whenever DIFF occurs, $\mathcal{B}$ succeeds unless GUESS occurs. In formulae,

$$\begin{aligned}\Pr[\mathsf{DIFF}] &= \Pr[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}] + \Pr[\mathsf{DIFF} \wedge \mathsf{GUESS}] \\ &\leq \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + \Pr[\mathsf{GUESS}].\end{aligned}$$

Together with Lemma 1 below, this yields the desired bound. □

We continue by bounding the probability of event GUESS. We will also need to analyze a very similar event in Thm. 2, in which we revisit the FFP-CCA attacker $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$, and where we will simulate $\mathcal{B}$'s oracle oDecrypt via an oracle oDecrypt′ (see Fig. 4). Therefore, we generalize the definition of event GUESS accordingly. Since GUESS means that $\mathcal{A}$ computed a ciphertext $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$ *before* querying $\mathsf{G}$ on $m$, the probability can be upper bounded in terms of the maximal probability of any ciphertext being hit by $\mathsf{Enc}(pk, -; -)$. For completeness, we prove Lem. 1 in the full version.

**Lemma 1.** *Let* PKE *be $\gamma$-spread, and let $\mathcal{A}$ be an adversary expecting oracles* $\mathsf{G}$, $\mathsf{H}$ *as well as either a decapsulation oracle* oDecaps *for* $\mathsf{KEM}^{\perp}_{m}$ *or a decryption oracle* oDecrypt *for* $\mathsf{PKE}^{\mathsf{G}}$*, issuing at most $q_D$ queries to the latter. When run with $\mathsf{G}'$ and simulated oracle* oDecaps′ *(or* oDecrypt′*, respectively), there is only a small probability that original oracle* oDecaps *(*oDecrypt*) would not have rejected, but simulation* oDecaps′ *(*oDecrypt′*) does. Concretely, we have*

$$\Pr[\mathsf{GUESS}] \leq q_D \cdot 2^{-\gamma}. \tag{3}$$

So far, we have shown that whenever an IND-CCA adversary $\mathcal{A}$'s behaviour is significantly changed by being run with simulation oDecaps′ instead of the real oracle oDecaps, we can use $\mathcal{A}$ to find a failing plaintext, assuming access to the FFP-CCA decryption oracle oDecrypt for $\mathsf{PKE}^{\mathsf{G}}$. We now show that oDecrypt can be simulated via oracle oDecrypt′ (see Fig. 4) without the secret key, thereby being able to construct an FFP-CPA adversary from any FFP-CCA adversary that succeeds with the same probability up to (at most) a multiplicative factor equal to the number of decryption queries the FFP-CCA adversary makes.

**Theorem 2.** *Let* PKE *be -$\gamma$-spread, and let $\mathcal{B}$ be an* FFP-CCA *adversary against* $\mathsf{PKE}^{\mathsf{G}}$*, issuing at most $q_{\mathsf{D}}$ many decryption queries. Then there exists an* FFP-CPA *adversary $\tilde{\mathcal{B}}$ such that*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \leq (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}\left(\tilde{\mathcal{B}}\right) + q_{\mathsf{D}} \cdot 2^{-\gamma} \ . \tag{4}$$

*Adversary $\tilde{\mathcal{B}}$ makes at most the same number of queries to* $\mathsf{G}$ *as $\mathcal{B}$ and runs in about the time of $\mathcal{B}$ .*

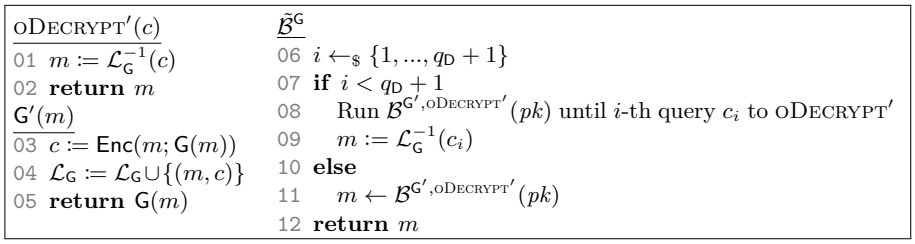


**Fig. 4.** Simulation oDecrypt′ of oracle oDecrypt for $\mathsf{PKE}^{\mathsf{G}}$, which is defined analogously to oDecaps′ (see Fig. 3), and FFP-CPA adversary $\tilde{\mathcal{B}}$. For the reader's convenience, we repeat the definition of $\mathsf{G}'$.

*Proof.* To simulate oDecrypt, we use a similar strategy as in the proof of Theorem 1. We define the events DIFF and GUESS in the same way as in the proof of Theorem 1, except now with respect to the adversary $\mathcal{B}$ and oracles oDecrypt (oDecrypt′) instead of oDecaps (oDecaps′). If our simulation does not fail, then a reduction can simulate the FFP-CCA game to $\mathcal{B}$ and use $\mathcal{B}$'s output to win its own FFP-CPA game. The simulation will fail if either GUESS happens (with probability at most $q_D \cdot 2^{-\gamma}$ due to Lem. 1), or DIFF, while GUESS does not, meaning that the failing message triggering DIFF can be extracted from $\mathcal{L}_{\mathsf{G}}$. Our reduction $\tilde{\mathcal{B}}$ combines both approaches (using $\mathcal{B}$'s output and $\mathcal{L}_{\mathsf{G}}$). Since $\tilde{\mathcal{B}}$ has no knowledge of the secret key, it cannot determine which message will let it succeed and hence has to guess.

Assume without loss of generality that $\mathcal{B}$ makes exactly $q_{\mathsf{D}}$ many queries to oracle oDecrypt. Consider the adversary $\tilde{\mathcal{B}}^{\mathsf{G}}$ in Fig. 4. $\tilde{\mathcal{B}}$ samples $i \leftarrow \{1, ..., q_{\mathsf{D}}+1\}$ and either runs $\mathcal{B}^{\mathsf{G}',\text{oDecrypt}'}$ until its $i$-th query to oDecrypt′ or until the end if $i = q_{\mathsf{D}}+1$. To implement $\mathsf{G}'$, $\tilde{\mathcal{B}}$ uses its oracle $\mathsf{G}$. Simulation oDecrypt′ is defined in Fig. 4 and works analogous to oDecaps′ in the previous proof. Finally, $\tilde{\mathcal{B}}$ outputs query preimage $\mathcal{L}_{\mathsf{G}}^{-1}(c_i)$, where $c_i$ is $\mathcal{B}$'s $i$-th query to decryption oracle oDecrypt′, unless $i = q_{\mathsf{D}} + 1$, in which case $\tilde{\mathcal{B}}$ outputs the output of $\mathcal{B}$.

Using the same chain of inequalities as in the proof of Thm. 1, and again using Lemma 1, we obtain

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \leq \Pr\left[\mathcal{B} \text{ wins } \wedge \neg\mathsf{DIFF}\right] + \Pr\left[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}\right] + q_{\mathsf{D}} \cdot 2^{-\gamma}. \quad (5)$$

Adversary $\tilde{\mathcal{B}}$ perfectly simulates game FFP-CCA unless DIFF occurs, and wins with probability $1/q_{\mathsf{D}}+1$ if $\mathcal{B}$ wins by returning a failing plaintext or if $\mathcal{B}$ issues a decryption query that triggers DIFF but not GUESS.

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}\left(\tilde{\mathcal{B}}\right) = \frac{1}{q_{\mathsf{D}}+1} \cdot \left(\Pr\left[\mathcal{B} \text{ wins } \wedge \neg\mathsf{DIFF}\right] + \Pr\left[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}\right]\right) \quad (6)$$

Combining Equations (5) and (6) yields the desired bound. □

Next, we observe that IND-CPA security of $\mathsf{KEM}^{\perp}_{m}$ can be based on passive security of PKE. This result is implicitly contained in [HHK17] since [HHK17]

proved such a result for IND-CCA security of $\mathsf{KEM}_m^{\not\perp}$. Combining Thms. 1 and 2 with the result from [HHK17], we obtain the following

**Corollary 1.** *Let* PKE *and* $\mathcal{A}$ *be as in Thm. 1. Then there exist a* OW-CPA *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *and an* IND-CPA *adversary* $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\not\perp}}(\mathcal{A}) \leq & (q_{\mathsf{RO}} + q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) \\ & + (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) + 2q_{\mathsf{D}} \cdot 2^{-\gamma}\end{aligned}$$

*and*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\not\perp}}(\mathcal{A}) \leq & \; 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{2 \cdot (q_{\mathsf{RO}} + q_{\mathsf{D}}) + 1}{|\mathcal{M}|} \\ & + (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + 2q_{\mathsf{D}} \cdot 2^{-\gamma}.\end{aligned}$$

*$\mathcal{C}$ makes $q_{\mathsf{G}}$ queries to* G*, and all adversaries run in about the time of $\mathcal{A}$.*

We remark that the factor 2 in front of the additive term $q_{\mathsf{D}} \cdot 2^{-\gamma}$ is an artefact of our modular proof (in terms of Theorems 1 and 2). It is straightforward to show that the bound of Cor. 1 can be proven without the factor of 2, when directly analyzing the composition of the reductions from Theorems 1 and 2.

When comparing our bounds with the respective bounds from [HHK17], we note that our bounds are still in the same asymptotic ball park and differ from the bounds in [HHK17] essentially by replacing the worst-case correctness term $\delta_{\mathrm{wc}}$ (there denoted by $\delta$) present in [HHK17] by $\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B})$, and having an additional term in $\gamma$ even for $\mathsf{KEM}_m^{\not\perp}$. We believe that the additional $\gamma$-term could be removed by doing a direct proof for $\mathsf{KEM}_m^{\not\perp}$, but redoing the whole proof for this variant was outside the scope of this work. We will further analyze $\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B})$ in Sect. 5.

*Remark 1 (Obtaining the results for $\mathsf{FO}_m^{\not\perp}[\mathsf{PKE}]$).* We can use the results from [BHH+19] to furthermore show that the bounds given in Cor. 1 also hold if $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is replaced with $\mathsf{KEM}_m^{\not\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$: In more detail, it follows directly from [BHH+19, Theorem 3] that for any IND-CCA-KEM attacker $\mathcal{A}$ against $\mathsf{KEM}_m^{\not\perp}$, there exists an IND-CCA-KEM attacker $\mathcal{B}$ against $\mathsf{KEM}_m^{\perp}$ such that $\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\not\perp}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{B})$ and Cor. 1 does not contain any terms relative to $\mathsf{KEM}_m^{\perp}$ itself, it only contains terms relative to the underlying schemes PKE and $\mathsf{PKE}^{\mathsf{G}}$.

## 3 Compressed oracles and extraction

We want to generalize the ROM results obtained in Sect. 2 to the QROM. To this end, we will use an extension of the compressed oracle technique [Zha19] that was introduced in [DFMS21]. It was shown in [Zha19] how a quantum-accessible random oracle $\mathsf{O} : X \to Y$ can be simulated by preparing a database $D$ with an entry $D_x$ for each input value $x$, with each $D_x$ being initialized as a

uniform superposition of all elements of $Y$, and omitting the "oracle-generating" measurements until after the algorithm accessing $\mathsf{O}$ has finished. In [DFMS21], this oracle simulation was generalized to obtain an *extractable* oracle simulator $\mathsf{eCO}$ (for extractable Compressed Oracle) that has two interfaces, the random oracle interface $\mathsf{eCO.RO}$ and an extraction interface $\mathsf{eCO.E}_f$, defined relative to a function $f : X \times Y \to T$. Whenever it is clear from context which function $f$ is used, we simply write $\mathsf{eCO.E}$ instead of $\mathsf{eCO.E}_f$.

In general, $\mathsf{eCO.E}_f$ can extract preimage entries from the "database" $D$ during the runtime of an adversary instead of only after the adversary terminated. This allows for adaptive behaviour of a reduction, based on an adversary's queries. In [DFMS21], it was already used for the same purpose we need it for – the simulation of a decapsulation oracle, by having $\mathsf{eCO.E}$ extract a preimage plaintext from the ciphertext on which the decapsulation oracle was queried. We will denote oracles modelled as extractable quantum-accessible ROs by $\mathrm{eQRO}_f$, and a proof that uses an $\mathrm{eQRO}_f$ will be called *a proof in the* $\mathrm{eQROM}_f$.

We will now make this description more formal, closely following notation and conventions from [DFMS21]. Like in [DFMS21], we describe an inefficient variant of the oracle that is not (yet) "compressed". Efficient simulation is possible via a standard sparse encoding, see [DFMS21, Appendix A]. The simulator $\mathsf{eCO}$ for a random function $\mathsf{O} : \{0,1\}^m \to \{0,1\}^n$ is a stateful oracle with a state stored in a quantum register $D = D_{0^m} \dots D_{1^m}$, where for each $x \in \{0,1\}^m$, register $D_x$ has $n+1$ qubits used to store superpositions of $n$-bit output strings $y$, encoded as $0y$, and an additional symbol $\perp$, encoded as $10^n$. We adopt the convention that an operator expecting $n$ input qubits acts on the last $n$ qubits when applied to $D_x$. The compressed oracle has the following three components.

- The initial state of the oracle, $|\phi\rangle = |\perp\rangle^{2^m}$
- A quantum query with query input register $X$ and output register $Y$ is answered using the oracle unitary $O_{XYD}$ defined by

$$O_{XYD} |x\rangle_X = |x\rangle_X \otimes \left(F_{D_x} \mathsf{CNOT}^{\otimes n}_{D_x:Y} F_{D_x}\right), \tag{7}$$

  where $F|\perp\rangle = |\phi_0\rangle$, $F|\phi_0\rangle = |\perp\rangle$ and $F|\psi\rangle = |\psi\rangle$ for all $|\psi\rangle$ such that $\langle\psi|\perp\rangle = \langle\psi|\phi_0\rangle = 0$, with $|\phi_0\rangle = |+\rangle^{\otimes n}$ being the uniform superposition. The CNOT operator here is responsible for XORing the function value (stored in $D_x$, now in superposition) into the query algorithm's output register.
- A *recovery algorithm* that recovers a standard QRO $\mathsf{O}$: apply $F^{\otimes 2^m}$ to $D$ and measure it to obtain the function table of $\mathsf{O}$.

We now make our description of the extraction interface $\mathsf{eCO.E}$ formal: Given a random oracle $\mathsf{O} : \{0,1\}^m \to \{0,1\}^n$, let $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^\ell$ be a function. We define a family of measurements $(\mathcal{M}^t)_{t\in\{0,1\}^\ell}$. The measurement $\mathcal{M}^t$ has measurement projectors $\{\Sigma^{t,x}\}_{x\in\{0,1\}^m\cup\{\emptyset\}}$ defined as follows. For $x \in \{0,1\}^m$, the projector selects the case where $D_x$ is the first (in lexicographical order) register that contains $y$ such that $f(x,y) = t$, i.e.

$$\Sigma^{t,x} = \bigotimes_{x'<x} \bar{\Pi}^{t,x'}_{D'_x} \otimes \Pi^{t,x}_{D_x}, \quad \text{with} \quad \Pi^{t,x} = \sum_{\substack{y\in\{0,1\}^n:\\ f(x,y)=t}} |y\rangle\langle y| \tag{8}$$

and $\bar{\Pi} = \mathbb{1} - \Pi$. $\Sigma^{t,\emptyset}$ covers the case where no register contains such a $y$, i.e.

$$\Sigma^{t,\emptyset} = \bigotimes_{x' \in \{0,1\}^m} \bar{\Pi}^{t,x'}_{D'_x}. \tag{9}$$

As an example, say we model a random oracle $\mathsf{H}$ as such an $\mathrm{eQRO}_f$. Using $f(x,y) := [\![\mathsf{H}(x) = y]\!]$, $\mathcal{M}^1$ allows us to extract a preimage of $y$.

$\mathsf{eCO}$ is initialized with the inital state of the compressed oracle. $\mathsf{eCO.RO}$ is quantum-accessible and applies the compressed oracle query unitary $O_{XYD}$. $\mathsf{eCO.E}$ is classically-accessible. On input $t$, it applies $\mathcal{M}^t$ to $\mathsf{eCO}$'s internal state and returns the result. $\mathsf{eCO}$ has useful properties that were characterized in [DFMS21, Theorem 3.4]. These characterisations are in terms of the quantity

$$\begin{aligned} \Gamma(f) &= \max_t \Gamma_{R_{f,t}}, \text{ with} \\ R_{f,t}(x,y) &:\Leftrightarrow f(x,y) = t \text{ and} \\ \Gamma_R &:= \max_x |\{y \mid R(x,y)\}|. \end{aligned} \tag{10}$$

For $f = \mathsf{Enc}(\cdot;\cdot)$, the encryption function of a PKE that takes as inputs a message $m$ and an encryption randomness $r$ , we have $\Gamma(f) = 2^{-\gamma}|\mathcal{R}|$ if $\mathsf{PKE}$ is $\gamma$-spread. In this case, $\mathsf{eCO.E}(c)$ outputs a plaintext $m$ such that $\mathsf{Enc}(m, \mathsf{eCO.RO}(m)) = c$, or $\perp$ if the ciphertext $c$ has not been computed using $\mathsf{eCO.RO}$ before.

## 4 QROM reduction

In this section, we generalize the reductions from Sect. 2 to the QROM. To do so, we give in Fig. 6 the quantum analogues of the simulated decapsulation oracles oDecaps′ and oDecaps″ from Fig. 3, which were (essentially) developed in [DFMS21]. We have to adapt our simulations since the ROM simulations from Fig. 3 use book-keeping techniques and therefore cannot be easily implemented in the standard QROM. Instead, we use the formalism described in Sect. 3, i.e., we use a simulation of a quantum-accessible random oracle and *make use of the additional extraction interface* $\mathsf{eCO.E}$: While the simulations in Fig. 3 had access to a list $\mathcal{L}_{\mathsf{G}}$ that could be used to extract potential ciphertext preimages, the simulations in Fig. 6 can now extract them by accessing extractor $\mathsf{eCO.E}$ (see lines 12 and 24). The rest of the simulation is exactly as before. Using the notation from Sect. 3, we denote the modelling of the ROM as extractable by $\mathrm{eQROM}_{\mathsf{Enc}}$, as we extract preimages relative to function $f = \mathsf{Enc}(pk,\cdot,\cdot)$, with the message being $f$'s first and the randomness being $f$'s second input.

We split this section as follows: Sect. 4.1 ends with $\mathsf{IND\text{-}CPA}$ security of $\mathsf{KEM}^{\perp}_m$ and $\mathsf{FFP\text{-}CPA}$ security of $\mathsf{PKE}^{\mathsf{G}}$, in the $\mathrm{eQROM}_{\mathsf{Enc}}$.We give the $\mathrm{eQROM}_f$ definition of $\mathsf{FFP\text{-}ATK}$ in Fig. 5. Sect. 4.2 develops the necessary $\mathrm{eQROM}_{\mathsf{Enc}}$ tools to further analyze $\mathsf{IND\text{-}CPA}$ security of $\mathsf{KEM}^{\perp}_m$. Concretely, Sect. 4.2 provides an $\mathrm{eQROM}_{\mathsf{Enc}}$-compatible variant of the one-way to hiding ($\mathsf{OWTH}$) lemma for semi-classical oracles as introduced in [AHU19]. Equipped with the results from Sect. 4.2, we show in Sect. 4.3 that *also in the* $\mathrm{eQROM}_{\mathsf{Enc}}$, $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}^{\perp}_m[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ can be based on passive security of $\mathsf{PKE}$.

| **Game** FFP-ATK | oDECRYPT($c$) |
|---|---|
| 01 $(pk, sk) \leftarrow \mathsf{KG}$ | 05 $m := \mathsf{Dec}(sk, c)$ |
| 02 $m \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ATK}}, \mathsf{eCO}}(pk)$ | 06 **return** $m$ |
| 03 $c := \mathsf{Enc}(pk, m)$ | |
| 04 **return** $[\![\mathsf{Dec}(sk, c) \neq m]\!]$ | |

**Fig. 5.** Games FFP-ATK for a deterministic PKE, where $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, in the eQROM$_f$. Like in its classical counterpart (see Fig. 2, page 7), $\mathsf{O}_{\mathsf{ATK}}$ is the decryption oracle present in the respective IND-ATK-KEM game . The only difference is that random oracle G is now modelled as an extractable superposition oracle eCO.

### 4.1 From IND-CPA$_{\mathsf{FO}[\mathsf{PKE}]}$ and FFP-CCA$^{\mathsf{G}}_{\mathsf{PKE}}$ to IND-CCA$_{\mathsf{FO}[\mathsf{PKE}]}$

We begin by proving a quantum analogue of Thm. 1.

**Theorem 3.** *Let* PKE *be a (randomized)* PKE *that is $\gamma$-spread, and* $\mathsf{KEM}^{\perp}_m := \mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. *Let $\mathcal{A}$ be an* IND-CCA-KEM*-adversary (in the QROM) against* $\mathsf{KEM}^{\perp}_m$*, making at most $q_{\mathsf{D}}$, $q_{\mathsf{G}}$ and $q_{\mathsf{H}}$ queries to* oDECAPS, G *and* H, *respectively. Let furthermore $d$ and $w$ be the combined query depth and query width of $\mathcal{A}$'s random oracle queries. Then there exist an* IND-CPA-KEM *adversary $\tilde{\mathcal{A}}$ and an* FFP-CCA *adversary $\mathcal{B}$ against* $\mathsf{PKE}^{\mathsf{G}}$*, both in the* eQROM$_{\mathsf{Enc}}$*, such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}(\tilde{\mathcal{A}}) + \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + 12 q_{\mathsf{D}}(q_{\mathsf{G}} + 4q_{\mathsf{D}}) \cdot 2^{-\gamma/2}.$$

*The adversary $\tilde{\mathcal{A}}$ makes $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ queries to* eCO.RO *with a combined depth of $d + q_{\mathsf{D}}$ and a combined width of $w$, and $q_{\mathsf{D}}$ queries to* eCO.E. *Here,* eCO.RO *simulates* $\mathsf{G} \times \mathsf{H}$. *The adversary $\mathcal{B}$ makes $q_{\mathsf{D}}$ many queries to* oDECRYPT *and* eCO.E *and $q_{\mathsf{G}}$ queries to* eCO.RO*, and neither $\tilde{\mathcal{A}}$ nor $\mathcal{B}$ query* eCO.E *on the challenge ciphertext. The running times of the adversaries $\tilde{\mathcal{A}}$ and $\mathcal{B}$ are bounded as* $\mathrm{Time}(\tilde{\mathcal{A}}) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$ *and* $\mathrm{Time}(B) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$.

Before proving the theorem, we point out similarities and differences to the ROM counterpart, Thm. 1. First note that the bounds look very similar. The only difference lies in the additive error term that depends on the spreadness parameter $\gamma$. In the above theorem, this additive error term $O(q_{\mathsf{D}} q_{\mathsf{G}} 2^{-\gamma/2})$ is much larger than the term $O(q_{\mathsf{D}} 2^{-\gamma})$ present in Thm. 1. It originates from dealing with the fact that the extraction technique used to simulate the Decaps oracle inflicts an error onto the simulation of the QRO. We expect that for many real-world schemes, the additive security loss of $O(q_{\mathsf{D}} q_{\mathsf{G}} 2^{-\gamma/2})$ is still small enough to be neglected. Another important difference between Thm. 3 and Thm. 1 is of course that the adversaries $\tilde{\mathcal{A}}$ and $\mathcal{B}$ are now in the non-standard eQROM$_{\mathsf{Enc}}$. Looking ahead, we provide further reductions culminating in Cor. 6 which gives a standard-QROM bound for $\mathsf{KEM}^{\perp}_m$ in terms of (standard model) security properties of PKE.

*Proof.* We prove this theorem via a number of hybrid games, drawing some inspiration from the reduction for the entire FO transformation given in [DFMS21].

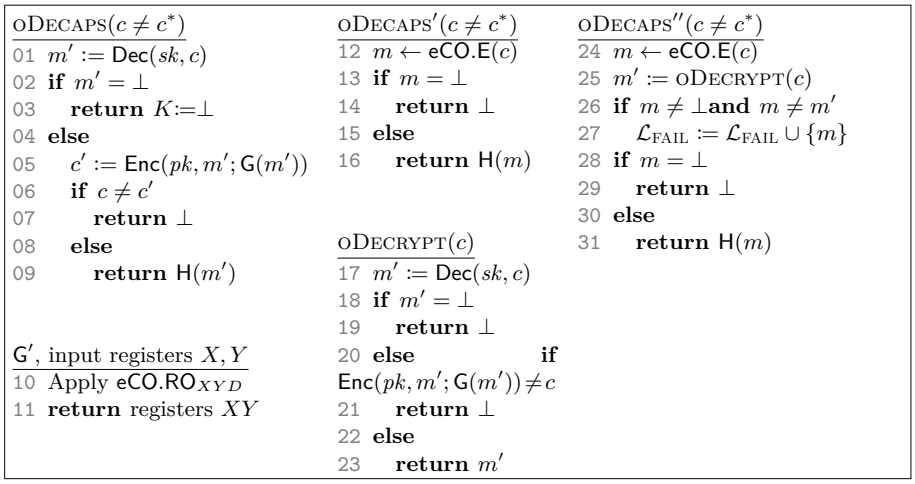


**Fig. 6.** Simulated and failing-plaintext-extracting versions of the decapsulation oracle oDecaps for $\mathsf{FO}^{\bot}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$, using the extractable QRO simulator $\mathsf{eCO}$ from [DFMS21] (see Sect. 3). The simulations of oDecaps are exactly like the ROM ones in Fig. 3 except for how they extract ciphertext preimages (lines 12, 24). $\mathsf{eCO}$ is assumed to be freshly initialized before oDecaps′ or oDecaps″ is used for the first time, and extraction interface $\mathsf{eCO.E}$ is defined with respect to function $f = \mathsf{Enc}(pk, \cdot; \cdot)$.

**Game $\mathbf{G_0}$** is $\mathsf{IND\text{-}CCA\text{-}KEM}_{\mathsf{KEM}^{\bot}_m}(\mathcal{A})$.

**Game $\mathbf{G_1}$** is like **Game $\mathbf{G_0}$**, except for two modifications: The quantum-accessible random oracle $\mathsf{G}$ is replaced by $\mathsf{G}'$ as defined in Fig. 6 , and after the adversary has finished, we compute $\hat{m}_i := \mathsf{eCO.E}(c_i)$ for all $i = 1, ..., q_{\mathsf{D}}$, where $c_i$ is the input to the adversary's $i$th decapsulation query. By property 1 in [DFMS21, Lem. 3.4], $\mathsf{G}'$ perfectly simulates $\mathsf{G}$ until the first $\mathsf{eCO.E}$-query, and since the first $\mathsf{eCO.E}$-query occurs only after $\mathcal{A}$ finishes, we have

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\bot}_m}(\mathcal{A}) = \mathrm{Adv}^{\mathbf{Game\ G_0}} = \mathrm{Adv}^{\mathbf{Game\ G_1}} \ . \tag{11}$$

**Game $\mathbf{G_2}$** is like **Game $\mathbf{G_1}$**, except that $\hat{m}_i := \mathsf{eCO.E}(c_i)$ is computed right after $\mathcal{A}$ submits $c_i$ instead of computing it in the end. Note that **Game $\mathbf{G_2}$** can be obtained from **Game $\mathbf{G_1}$** by first swapping the $\mathsf{eCO.E}$ call that produces $\hat{m}_1$ with all $\mathsf{eCO.RO}$ calls that happen after the adversary submits $c_1$, including the calls inside oDecaps, then continuing with the $\mathsf{eCO.E}$-call that produces $\hat{m}_2$, etc. By property 2.c of [DFMS21, Lem. 3.4] and since $\Gamma(\mathsf{Enc}(\cdot;\cdot)) = 2^{-\gamma}|\mathcal{R}|$ for $\gamma$-spread PKE schemes, we have that

$$\left|\mathrm{Adv}^{\mathbf{Game\ G_1}} - \mathrm{Adv}^{\mathbf{Game\ G_2}}\right| \leq 8\sqrt{2} q_{\mathsf{D}}(q_{\mathsf{G}} + q_{\mathsf{D}}) \cdot 2^{-\gamma/2} \ . \tag{12}$$

**Game $\mathbf{G_3}$** is the same as **Game $\mathbf{G_2}$**, except that $\mathcal{A}$ in run with access to the oracle oDecaps′ instead of oDecaps, meaning that upon a decapsulation query on $c_i$, $\mathcal{A}$ receives $\text{oDecaps}'(c_i) = \mathsf{H}(\hat{m}_i)$ instead of $\text{oDecaps}(c_i) =$

$\mathsf{Decaps}(sk, c_i)$ (using the convention $\mathsf{H}(\bot) := \bot$). We still let the game also compute $\textsc{oDecaps}(c_i)$, as $\textsc{oDecaps}$ makes queries to $\mathsf{eCO.RO}$ which can influence the behavior of $\mathsf{eCO.E}$ in subsequent queries. (Note that the reencryption step of $\textsc{oDecaps}$ triggers a call to $\mathsf{G}'$, which in turn uses $\mathsf{eCO.RO}$.) We define $\mathcal{B}$ exactly as in the proof of Thm. 1, except that it uses the oracles $\mathsf{G}'$ and $\textsc{oDecaps}''$ defined in Fig. 6: $\mathcal{B}$ runs $\mathcal{A}^{\mathsf{G}',\mathsf{H},\textsc{oDecaps}''}$, using its own $\mathsf{FFP\text{-}CCA}$ oracle $\textsc{oDecrypt}$ to simulate $\textsc{oDecaps}''$ and answering $\mathsf{H}$ queries by simulating a fresh compressed oracle.[6] As soon as $\textsc{oDecaps}''$ adds a plaintext $m$ to $\mathcal{L}_{\text{FAIL}}$, $\mathcal{B}$ aborts $\mathcal{A}$ and returns $m$. If $\mathcal{A}$ finishes and $\mathcal{L}_{\text{FAIL}}$ is still empty, $\mathcal{B}$ returns $\bot$.

Let $\mathsf{DIFF}$ be the event that $\mathcal{A}$ makes a decryption query $c$ in $\mathbf{Game\ G_2}$ such that $\textsc{oDecaps}(c) \neq \textsc{oDecaps}'(c)$. Like in Thm. 1, we bound

$$\begin{aligned}
&\frac{1}{2} + \mathrm{Adv}^{\mathbf{Game\ G_2}} = \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_2}\right] \\
&= \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_2} \wedge \neg\mathsf{DIFF}\right] + \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_2} \wedge \mathsf{DIFF}\right] \\
&= \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_3} \wedge \neg\mathsf{DIFF}\right] + \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_2} \wedge \mathsf{DIFF}\right] \\
&\leq \Pr\left[\mathcal{A} \text{ wins in } \mathbf{Game\ G_3}\right] + \Pr\left[\mathsf{DIFF}\right] = \frac{1}{2} + \mathrm{Adv}^{\mathbf{Game\ G_3}} + \Pr\left[\mathsf{DIFF}\right] \quad .
\end{aligned}$$

Again, event $\mathsf{DIFF}$ encompasses three cases: For some decapsulation query $c$,

- the original decapsulation oracle $\textsc{oDecaps}(c)$ rejects, but the simulation $\textsc{oDecaps}'(c) = \mathsf{H}(\hat{m})$ does not. By construction of the oracles, this implies that $\mathsf{Dec}(sk, \mathsf{Enc}(pk, \hat{m}, \mathsf{eCO.RO}(\hat{m}))) \neq \hat{m}$ if the $\mathsf{eCO.RO}$ call in the previous equation is performed right after the considered $\textsc{oDecaps}''$ call.
- Neither oracle rejects, but the return values differ, i.e., calling $\mathsf{eCO.E}(c)$ in line 12 yielded something different than $\mathsf{Dec}(sk, c)$. Like above, this implies that preimage $\hat{m} := \mathsf{eCO.E}(c)$ fails
- $\textsc{oDecaps}(c)$ does not reject, while $\textsc{oDecaps}'(c)$ does, i.e., $\hat{m} := \mathsf{eCO.E}(c)$ in line 12 yielded $\bot$, but the re-encryption check inside the $\textsc{oDecaps}$ call in line 25 checked out, meaning that $\mathsf{Enc}(pk, m, \mathsf{eCO.RO}(m)) = c$ for $m := \mathsf{Dec}(sk, c)$. (Equivalently, the latter means that $\textsc{oDecrypt}(c) = m$.)

In the above, any statements about $\mathsf{eCO}$ calls that are not actually performed by the adversary or an oracle are assumed to be made right after the query $c$ and do not cause any measurement disturbance in that case.

We will again denote the last case by $\mathsf{GUESS}$. Whenever $\mathsf{DIFF}$ occurs, $\mathcal{B}$ succeeds unless only case $\mathsf{GUESS}$ occurs: If $\mathsf{DIFF} \wedge \neg\mathsf{GUESS}$ occurs, then a failing plaintext is extractable from the ciphertext that triggered $\mathsf{DIFF} \wedge \neg\mathsf{GUESS}$ (this time due to access to $\mathsf{eCO.E}$), and the plaintext is recognisable as failing by $\mathcal{B}$ due to its $\mathsf{FFP\text{-}CCA}$ oracle $\textsc{oDecrypt}$. In formulae,

$$\Pr[\mathsf{DIFF}] = \Pr[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}] + \Pr[\mathsf{DIFF} \wedge \mathsf{GUESS}] \leq \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}[\mathcal{B}] + \Pr[\mathsf{GUESS}].$$

[6] We remark that a $t$-wise independent function for sufficiently large $t = O(q_{\mathsf{H}} + q_{\mathsf{D}})$ also suffices, which is more efficient as it doesn't require (nearly as much) quantum memory.

In summary, we can bound the difference in advantages between **Game $\mathbf{G_2}$** and **Game $\mathbf{G_3}$** as

$$\left|\mathrm{Adv}^{\textbf{Game } \mathbf{G_2}} - \mathrm{Adv}^{\textbf{Game } \mathbf{G_3}}\right| \leq \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + \Pr\left[\mathsf{GUESS}\right].$$

The following two steps are in a certain sense symmetric to the steps for **Games 0-2**: $\mathcal{A}$ playing **Game $\mathbf{G_3}$** can almost be simulated without using the oDecaps oracle, except that oDecaps is still invoked before each call to the oracle oDecaps$'$, without the result ever being used. This is an artifact from **Game $\mathbf{G_2}$**. Omitting the oDecaps invocations might introduce changes in $\mathcal{A}$'s view, as these invocations might influence the behavior of $\mathsf{eCO.E}$ in subsequent queries. We therefore define **Game $\mathbf{G_4}$** like **Game $\mathbf{G_3}$**, except that the oDecaps invocations are postponed until after $\mathcal{A}$ finishes. By a similar argument as for the transition from **Game $\mathbf{G_1}$** to **Game $\mathbf{G_2}$**, we obtain

$$\left|\mathrm{Adv}^{\textbf{Game } \mathbf{G_3}} - \mathrm{Adv}^{\textbf{Game } \mathbf{G_4}}\right| \leq 8\sqrt{2}q_{\mathsf{D}}^2 2^{-\gamma/2} \quad .$$

Finally, **Game $\mathbf{G_5}$** is like **Game $\mathbf{G_4}$**, but the computations of oDecaps$(c_i)$ are omitted entirely. In game 4, all invocations of oDecaps already happened after the execution of $\mathcal{A}$, hence this omission does not influence $\mathcal{A}$'s success probability.

Let $\tilde{\mathcal{A}}$ be an $\mathsf{IND\text{-}CPA\text{-}KEM}$ adversary against $\mathsf{KEM}^{\perp}_m$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$, simulating **Game $\mathbf{G_5}$** to $\mathcal{A}$: $\tilde{\mathcal{A}}$ has access to a single extractable oracle whose oracle interface $\mathsf{eCO.RO}$ simulates the combination of $\mathsf{G}$ and $\mathsf{H}$, i.e., $\mathsf{eCO.RO}$ simulates $\mathsf{G} \times \mathsf{H}$. (We decided to combine $\mathsf{G}$ and $\mathsf{H}$ into one oracle to simplify the subsequent analysis of the $\mathsf{IND\text{-}CPA}$ advantage against $\mathsf{KEM}^{\perp}_m$ that will be carried out in Sect. 4.3.) $\tilde{\mathcal{A}}$ runs $b' \leftarrow \mathcal{A}^{\mathsf{G}',\mathsf{H},\text{oDecaps}'}$ and returns $b'$. The simulation of $\mathcal{A}$'s oracles using $\mathsf{eCO.RO}$ is straightforward (preparing the redundant register in uniform superposition, querying the combined oracle, and uncomputing the redundant register).

We now have

$$\mathrm{Adv}^{\textbf{Game } \mathbf{G_4}} = \mathrm{Adv}^{\textbf{Game } \mathbf{G_5}} = \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}(\tilde{\mathcal{A}}). \tag{13}$$

Collecting the terms from the hybrid transitions, using Lem. 2 below, and bounding $q_{\mathsf{D}}2^{-\gamma} \leq q_{\mathsf{D}}^2 2^{-\gamma/2}$ yields the desired bound. The statements about query numbers, width and depth, as well as the runtime, are straightforward. □

Like in Sect. 2, we continue by bounding the probability of event $\mathsf{GUESS}$, and Lem. 2 below is the $\mathrm{eQROM}_{\mathsf{Enc}}$ analogue of Lem. 1. Again, we will soon revisit $\mathsf{FFP\text{-}CCA}$ attacker $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$, and we will simulate $\mathcal{B}$'s oracle oDecrypt via an oracle oDecrypt$'$ (see Fig. 7) that differs from oDecrypt if an event equivalent to $\mathsf{GUESS}$ occurs. Therefore, we again generalize the definition of event $\mathsf{GUESS}$ accordingly.

**Lemma 2.** *Let* $\mathsf{PKE}$ *and* $\mathcal{A}$ *be like in Lem. 1 (see page 9), except that* $\mathcal{A}$ *is now considered in the* $\mathrm{eQROM}_{\mathsf{Enc}}$. *Let* $\mathcal{A}$ *be run with* $\mathsf{G}'$ *and* oDecaps *or* oDecaps$'$ *(*oDecrypt *or* oDecrypt$'$*), but for each query* $c_i$, *both* $\hat{m}_i =$ oDecrypt$'(c_i)$

| $\textsc{oDecrypt}'(c)$ | $\mathsf{G}'$, input registers $X, Y$ |
|---|---|
| 01 $m \leftarrow \mathsf{eCO.E}(c)$ | 03 Apply $\mathsf{eCO.RO}_{XYD}$ |
| 02 **return** $m$ | 04 **return** registers $XY$ |

**Fig. 7.** Simulation $\textsc{oDecrypt}'$ of oracle $\textsc{oDecrypt}$ for $\mathsf{PKE}^{\mathsf{G}}$. For the reader's convenience, we repeat the definition of $\mathsf{G}'$.

*and $m_i = \textsc{oDecrypt}(c_i)$ are computed in that order, regardless of which of the two oracles $\textsc{oDecaps}$ and $\textsc{oDecaps}'$ ($\textsc{oDecrypt}$ and $\textsc{oDecrypt}'$) $\mathcal{A}$ has access to. Then $\mathsf{GUESS}$, the event that $\hat{m}_i = \perp$ while $m_i \neq \perp$, is very unlikely. Concretely,*

$$\Pr[\mathsf{GUESS}] \leq 2q_D \cdot 2^{-\gamma}. \tag{14}$$

*Proof.* We begin by bounding the probability that for some fixed $i \in \{1, ..., q_D\}$ we have $\hat{m}_i = \perp$ but $m_i \neq \perp$. From the definitions of $\textsc{oDecaps}$ and $\textsc{oDecaps}'$, as well as the definitions of the interfaces $\mathsf{eCO.RO}$ and $\mathsf{eCO.E}$, we obtain

$$\begin{aligned}\sqrt{\Pr[\hat{m}_i = \perp \wedge m_i \neq \perp]} &= \sqrt{\Pr[\hat{m}_i = \perp \wedge \mathsf{Enc}(m_i, \mathsf{eCO.RO}(m_i)) = c_i]} \\ &= \left\| \Pi_Y^{c,x} O_{XYF} \Sigma_F^{c,\emptyset} \left|m_i\right\rangle_X \left|0\right\rangle_Y \left|\psi_i\right\rangle_{FE} \right\| \end{aligned} \tag{15}$$

Here, $|\psi_i\rangle$ is the adversary-oracle state before $\mathcal{A}$ submits the query $c_i$ and the projectors $\Pi_Y^{c,x}$ and $\Sigma^{c,\emptyset}$ are with respect to $f = \mathsf{Enc}$ (see Eq. (8)). We begin by simplifying the expression on the right hand side. We have $O_{XYF} |m_i\rangle_X = F_{F_{m_i}} \mathsf{CNOT}^{\otimes n}_{F_{m_i}:Y} F_{F_{m_i}} \otimes |m_i\rangle_X$ and $\Pi_Y \mathsf{CNOT}^{\otimes n}_{F_{m_i}:Y} |0\rangle_Y = \mathsf{CNOT}^{\otimes n}_{F_{m_i}:Y} \Pi_{F_{m_i}} |0\rangle_Y$ for any projector $\Pi$ that is diagonal in the computational basis. We can thus simplify

$$\begin{aligned} &\left\| \Pi_Y^{c,x} O_{XYF} \Sigma_F^{c,\emptyset} \left|m_i\right\rangle_X \left|0\right\rangle_Y \left|\psi_i\right\rangle_{FE} \right\| = \left\| \Pi_{F_{m_i}}^{c,x} F_{F_{m_i}} \Sigma^{c,\emptyset} \left|m_i\right\rangle_X \left|0\right\rangle_Y \left|\psi_i\right\rangle_{FE} \right\| \\ \leq &\left\| F_{F_{m_i}} \Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} \left|m_i\right\rangle_X \left|0\right\rangle_Y \left|\psi_i\right\rangle_{FE} \right\| + \|[\Pi^{c,x}, F]\| \\ \leq &\left\| F_{F_{m_i}} \Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} \left|m_i\right\rangle_X \left|0\right\rangle_Y \left|\psi_i\right\rangle_{FE} \right\| + \sqrt{2} \cdot 2^{-\gamma/2} \end{aligned} \tag{16}$$

where we have applied the two observations and omitted any final unitary operators in the first equality, and the last inequality is due to Lemma 3.3 in [DFMS21]. But the remaining norm term vanishes as

$$\Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} = (\Pi^{c,x} \bar{\Pi}^{c,x})_{F_{m_i}} \otimes (\bar{\Pi}^{c,x})^{\otimes |\mathcal{M}|-1}_{F_{\mathcal{M}\setminus\{m_i\}}} = 0. \tag{17}$$

Combining Eqs. (15) to (17) and squaring the resulting inequality yields

$$\Pr[\hat{m}_i = \perp \wedge m_i \neq \perp] \leq 2 \cdot 2^{-\gamma}. \tag{18}$$

Collecting the terms and applying a union bound over the $q_D$ decapsulation queries yields the desired bound. □

So far, we have shown that whenever an IND-CCA adversary $\mathcal{A}$'s behaviour is significantly changed by being run with simulation $\text{oDecaps}'$ instead of the real oracle oDecaps, we can use $\mathcal{A}$ to find a failing plaintext, assuming access to the decryption oracle oDecrypt provided in the FFP-CCA game. We continue by proving an $\text{eQROM}_{\mathsf{Enc}}$-analogue of Thm. 2, i.e., we show that oDecrypt can be simulated via oracle $\text{oDecrypt}'$ (see Fig. 7) without the secret key, thereby being able to construct an FFP-CPA adversary from any FFP-CCA adversary (both in the $\text{eQROM}_{\mathsf{Enc}}$).

**Theorem 4.** *Let* PKE *and* $\mathcal{B}$ *be like in Thm. 2 (see page 9), except that* $\mathcal{B}$ *is now considered in the* $\text{eQROM}_{\mathsf{Enc}}$*, issuing at most* $q_{\mathsf{eCO.RO}}/q_{\mathsf{eCO.E}}$ *many queries to its respective oracle* eCO.RO/eCO.E*. Then there exist an* FFP-CPA *adversary* $\tilde{\mathcal{B}}$ *in the* $\text{eQROM}_{\mathsf{Enc}}$ *such that*

$$\text{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \leq (q_{\mathsf{D}}+1)\text{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{B}}) + 12q_{\mathsf{D}}(q_{\mathsf{G}}+4q_{\mathsf{D}})2^{-\gamma/2} \tag{19}$$

*The adversary* $\tilde{\mathcal{B}}$ *makes* $q_{\mathsf{eCO.RO}}$ *queries to* eCO.RO *and* $q_{\mathsf{eCO.E}}+q_{\mathsf{D}}$ *queries to* eCO.E*, and its runtime satisfies* $\text{Time}(\tilde{\mathcal{B}}) = \text{Time}(\mathcal{B}) + O(q_{\mathsf{D}})$.

*Proof.* On a high level, the proof works as follows. Analogous to Thm. 3, we simulate oDecrypt by $\text{oDecrypt}'$. As we wish to remove the usage of oDecrypt entirely, however, we cannot use it to determine at which $\text{oDecrypt}'$ query a failure occurs. We thus resort to guessing that information. On a technical level this proof follows the proof of Thm. 3 with deviations similar as in the proof of Thm. 2. Let $\text{oDecrypt}'$ be the simulation defined in Fig. 7. Let **Game $\mathbf{G_0}$** be the FFP-CCA-game, and let **Games $\mathbf{G_1}-\mathbf{G_5}$** be defined based on **Game $\mathbf{G_0}$** like in the proof of Thm. 3. Like in the proof of Thm. 3, we have

$$\begin{aligned}\text{Adv}^{\mathbf{Game\ G_0}} &\leq \text{Adv}^{\mathbf{Game\ G_5}} + 12q_{\mathsf{D}}(q_{\mathsf{G}}+2q_{\mathsf{D}})2^{-\gamma/2} + \Pr[\mathsf{DIFF}] \\ &\leq \text{Adv}^{\mathbf{Game\ G_5}} + 12q_{\mathsf{D}}(q_{\mathsf{G}}+2q_{\mathsf{D}})2^{-\gamma/2} + \Pr[\mathsf{DIFF}\wedge\neg\mathsf{GUESS}] + \Pr[\mathsf{GUESS}].\end{aligned} \tag{20}$$

Assume without loss of generality that $\mathcal{B}$ makes exactly $q_{\mathsf{D}}$ many queries to the oracle for $\mathsf{Dec}^{\mathsf{G}}$ (if it does not, we modify $\mathcal{B}$ by adding a number of useless decryption queries in the end). We define an FFP-CPA adversary $\tilde{\mathcal{B}}^{\mathsf{eCO}}$ defined exactly like the classical one in Fig. 4 (except that it has quantum access to its oracles), i.e., $\tilde{\mathcal{B}}$ samples $i \leftarrow \{1,...,q_{\mathsf{D}}+1\}$ and runs $\mathcal{B}^{\mathsf{G}',\text{oDecrypt}'}$ until the $i$-th query, or until the end if $i = q_{\mathsf{D}}+1$. Finally, $\tilde{\mathcal{B}}$ outputs $m_i$, the output of $\mathcal{B}^{\mathsf{G}',\text{oDecrypt}'}$'s $i$-th decryption query, unless $i = q_{\mathsf{D}}+1$, in which case $\tilde{\mathcal{B}}$ outputs the output of $\mathcal{B}^{\mathsf{G}',\text{oDecrypt}'}$. By construction,

$$\text{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{B}}) \geq \left(\text{Adv}^{\mathbf{Game\ G_5}} + \Pr[\mathsf{DIFF}\wedge\neg\mathsf{GUESS}]\right)/(q_{\mathsf{D}}+1) \tag{21}$$

(note that all instances of $\text{Adv}^{\mathbf{Game\ i}}$ are for $\mathcal{B}$ playing **Game i**.) Combining Eqs. (20) and (21) and Lem. 2 yields the desired bound. The statement about $\tilde{\mathcal{B}}$'s running time and number of queries is straightforward. □

Combining Theorems 3 and 4, we obtain the following

**Corollary 2.** *Let* PKE *and* $\mathcal{A}$ *be as in Thm. 3. Then there exist an* IND-CPA-KEM *adversary* $\tilde{\mathcal{A}}$ *and an* FFP-CPA *adversary* $\mathcal{B}$*, both in the* $\mathrm{eQROM}_{\mathsf{Enc}}$*, such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}(\mathcal{A}) \leq & \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}\left(\tilde{\mathcal{A}}\right) + (q_{\mathsf{D}}+1)\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \\ & + 24q_{\mathsf{D}}(q_{\mathsf{G}}+4q_{\mathsf{D}})2^{-\gamma/2} \end{aligned} \tag{22}$$

*Both* $\tilde{\mathcal{A}}$ *and* $\mathcal{B}$ *make* $q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}}$ *queries to* $\mathsf{eCO.RO}$*, with a combined depth (width) of* $d+q_{\mathsf{D}}$ *(*$w$*), and* $q_{\mathsf{D}}$ *queries to* $\mathsf{eCO.E}$*. The running times of* $\tilde{\mathcal{A}}$ *and* $\mathcal{B}$ *satisfy* $\mathrm{Time}(\tilde{\mathcal{A}}) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$ *and* $\mathrm{Time}(\mathcal{B}) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$.

Again, the additive error terms are a factor of 2 larger due to our modular proof (in terms of Theorems 3 and 4), which can be avoided with a direct proof.

While the additive error term depending on $\gamma$ improves by roughly a power 2 over the corresponding term in the security bound of [DFMS21], the only known concrete bound for $\mathsf{FO}^{\perp}_m$, we remark that we do not expect it to be tight. It turns out, however, that many relevant schemes have abundantly randomized ciphertexts.

### 4.2 Semi-classical OWTH in the eQROM$_f$

To analyze IND-CPA-KEM security of $\mathsf{KEM}^{\perp}_m$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$, we want to apply an $\mathrm{eQROM}_{\mathsf{Enc}}$ argument to show that keys encapsulated by $\mathsf{FO}^{\perp}_m[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ are random-looking unless the adversary can be used to attack the underlying scheme PKE. We will need to argue that the challenge key $K^* := \mathsf{H}(m^*)$ and the encryption randomness $\mathsf{G}(m^*)$ used for challenge ciphertext $c^*$ can be replaced with fresh random values, in the $\mathrm{eQROM}_{\mathsf{Enc}}$. To that end, we develop $\mathrm{eQROM}_f$ generalizations of the semi-classical OWTH theorems from [AHU19].

We will first describe how we model this 'replacing with fresh randomness' on a subset $\mathcal{S} \subset \mathcal{X}$ for superposition oracle, and how our approach generalizes previous approaches. Previous work (like [AHU19]) used two oracles $\mathsf{O}_0$ and $\mathsf{O}_1$ that only differ on some set $\mathcal{S}$, while algorithm $\mathcal{A}$'s input is always defined relative to oracle $\mathsf{O}_0$. In the case where $\mathcal{A}$'s oracle is $\mathsf{O}_1$, the input uses fresh randomness from the adversary's point of view. Here we meet the first $\mathrm{eQROM}_{\mathsf{Enc}}$-related roadblock: Superposition oracles have the property that initially, each value $\mathsf{eCO.RO}(x)$ is in *quantum superposition*, which complicates equating two oracles everywhere but on $\mathcal{S}$. As it suffices for our purpose, we define the 'resampling' set $\mathcal{S}$ as follows: We assume $\mathcal{A}$'s input *inp* to be classical, generated by an algorithm GenInp with classical access to $\mathsf{eCO}^0$. We can then define $\mathcal{S}$ as the set of all inputs $x$ queried by GenInp, e.g., for input $(c^*, K^*) := (\mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))), \mathsf{H}(m^*))$, $\mathcal{S}$ is $\{m^*\}$.) Apart from how we model $\mathcal{S}$, we proceed as in [AHU19]: Use $\mathsf{eCO}^0$ to generate $\mathcal{A}$'s input and replace $\mathcal{A}$'s access to $\mathsf{eCO}^0$ with access to an independent extractable compressed oracle $\mathsf{eCO}^1$.

Clearly, if GenInp does not query $\mathsf{eCO}^0$, the two oracles $\mathsf{eCO}^0$ and $\mathsf{eCO}^1$ are perfectly indistinguishable to $\mathcal{A}$. But what if $\mathcal{A}$'s input depends on $\mathsf{eCO}^0$? [AHU19] related $\mathcal{A}$'s distinguishing advantage to the probability of "FIND",

the event that an element of $\mathcal{S}$ is detected in $\mathcal{A}$'s queries to the QRO via a quantum measurement. This result, however, is in the (plain) QROM, and FIND is not the only distinction opportunity in the eQROM$_f$ as there are now two oracle interfaces, eCO.RO and eCO.E. As an example, let $\mathcal{A}$ have input $(x, t \coloneqq f(x, \mathsf{eCO}^0.\mathsf{RO}(x)))$ for some oracle input value $x$. *Without any* eCO.RO *query*, $\mathcal{A}$ can tell the two cases apart by querying eCO.E on $t$: Querying $\mathsf{eCO}^0.\mathsf{E}$ on $t$ results in output $x$ with overwhelming probability, while querying $\mathsf{eCO}^1.\mathsf{E}$ on $t$ yields output $\perp$. Extraction queries hence have to be taken into account.

Before stating this section's main theorems, we will describe our approach more formally. Borrowing the notation from [AHU19], we define *'punctured' versions* $\mathsf{eCO}\backslash\mathcal{S}$ of eCO: During each eCO.RO query, we first apply a *'semi-classical'* oracle $O_S^{\mathsf{SC}}$, and then oracle unitary $O_{XYD}$. Intuitively, $O_S^{\mathsf{SC}}$ marks if an element of $\mathcal{S}$ was found in one of the query registers. Formally, $O_S^{\mathsf{SC}}$ acts on the query input registers $X_1, \cdots X_w$ and a 'flag' register $F$ that holds one qubit per oracle query, by first mapping $|x_1, \cdots x_w, b\rangle$ to $|x_1, \cdots x_w, b \oplus [\![x_1 \in \mathcal{S} \vee \cdots \vee x_w \in \mathcal{S}]\!]\rangle$, and then measuring register $F$ in the computational basis.

Like in [AHU19], we denote the event that any measurement of $F$ returns 1 by FIND. In that case, the query has collapsed to a superposition of states where at least one input register only contains elements of $\mathcal{S}$. If FIND does not occur, then all oracle queries collapsed to states not containing any elements of $\mathcal{S}$, and in consequence, set $\mathcal{S}$ defining $\mathcal{A}$'s input is effectively removed from the query input domain. In this case, the only way to distinguish between $\mathsf{eCO}^0$ and $\mathsf{eCO}^1$ is to perform an extraction query where $\mathsf{eCO}^0.\mathsf{E}$ might return an element of $\mathcal{S}$. We will call this event EXT. If neither FIND nor EXT occur, the two scenarios are indistinguishable to $\mathcal{A}$.

The following helper lemma formalizes the above reasoning and extends it to some other probability distances: Eq. (23) formalizes that if $\mathcal{A}$ neither triggers FIND nor EXT, its behaviour in the two cases is the same: arbitrary events will be equally likely in both cases. Eqs. (24) and (25) have a similar interpretation. The proof of Lem. 3 is mostly reworking the probabilities by reasoning about the cases and eliminating the case where neither FIND nor EXT occurs. It is given in the full version.

**Lemma 3.** *Let* $\mathsf{eCO}^0$ *and* $\mathsf{eCO}^1$ *be two extractable superposition oracles from* $\mathcal{X}$ *to* $\mathcal{Y}$ *for some function* $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{T}$*, and let* GenInp *be an algorithm with classical output inp, having access to* $\mathsf{eCO}^0$*. Let* $\mathcal{S}$ *be the set of elements* $x \in \mathcal{X}$ *whose oracle values are needed to compute inp, and let* $\mathcal{T}_{\mathcal{S}} \coloneqq \{t \mid \exists x \in \mathcal{S} \text{ s.th. } t = f(x, \mathsf{eCO}^0(x))\}$*. Let* FIND *be the event that flag register* $F$ *is ever measured to be in state 1 during a call to* $\mathcal{A}$*'s punctured oracle, and let* EXT *be the event that* $\mathcal{A}$ *performs an extraction query on any* $t \in \mathcal{T}_{\mathcal{S}}$*. Let* E *be an arbitrary (classical) event. Then*

$$\Pr[\mathsf{E} \wedge \neg\mathsf{FIND} \wedge \neg\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}] = \Pr[\mathsf{E} \wedge \neg\mathsf{FIND} \wedge \neg\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}], \tag{23}$$

$$|\Pr[\mathsf{E} \wedge \neg\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}] - \Pr[\mathsf{E} \wedge \neg\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]| \leq \Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}], \tag{24}$$

$$|\Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}] - \Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]| \leq \Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}] \tag{25}$$

*where all probabilities are taken over the coins of* $\mathsf{GenInp}$ *and the internal randomness of* $\mathcal{A}$ *and we used* $\mathcal{A}^{\mathsf{O}_0}$ *as a shorthand for* $\mathcal{A}^{\mathsf{O}_0}(inp)$.

The following theorem relates the distinguishing advantage between $\mathsf{eCO}^0$ and $\mathsf{eCO}^1$ to the probability that $\mathsf{FIND}$ or $\mathsf{EXT}$ occur. Intuitively, the theorem states that no algorithm $\mathcal{A}$ will recognize the reprogramming unless $\mathcal{A}$ makes a random oracle or an extraction query related to its input. Thm. 5 is the $\mathrm{eQROM}_f$ counterpart of [AHU19, Th. 1, 'Semi-classical O2H']. Its proof is given in the full version. In the special case where $\mathsf{EXT}$ never happens, e.g., when extraction queries are triggered by an oracle simulation like oDECAPS′ that forbids critical inputs, we obtain the same bound as [AHU19, Th. 1], but in the $\mathrm{eQROM}_f$.

**Theorem 5 ($\mathrm{eQROM}_f$-$\mathsf{OWTH}$: Distinguishing to Finding).** *Let* $\mathsf{eCO}^0$, $\mathsf{eCO}^1$, $\mathsf{GenInp}$, $\mathcal{S}$, $\mathsf{FIND}$ *and* $\mathsf{EXT}$ *be like in Lem. 3. We define the* $\mathsf{OWTH}$ *distinguishing advantage function of* $\mathcal{A}$ *as*

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) := |\Pr[1 \leftarrow \mathcal{A}^{\mathsf{eCO}^0}(inp)] - \Pr[1 \leftarrow \mathcal{A}^{\mathsf{eCO}^1}(inp)]| \ ,$$

*where the probabilities are over the coins of* $\mathsf{GenInp}$ *and the randomness of* $\mathcal{A}$. *For any algorithm* $\mathcal{A}$ *of query depth* $d$ *with respect to* $\mathsf{eCO.RO}$, *we have that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq & 4 \cdot \sqrt{d \cdot \Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]} \\ & + 2 \cdot (\sqrt{d}+1) \cdot \sqrt{\Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^0}]} + \Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^1}] \ . \quad (26)\end{aligned}$$

*If additionally* $\Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] = \Pr[\mathsf{EXT} : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}] = 0$, *we obtain*

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]} \ . \quad (27)$$

In many cases, a desired reduction will not know the 'resampled' set $\mathcal{S}$. Thm. 6 relates the probability of $\mathsf{FIND}$ to the advantage of a preimage extractor $\mathsf{ExtractSet}$ that extracts an element of $\mathcal{S}$ without knowing $\mathcal{S}$: $\mathsf{ExtractSet}$ will run $\mathcal{A}$ with the unpunctured oracle $\mathsf{eCO}$ and measure one of its queries to generate its output. In one of our proofs, we additionally need to puncture on a set different from $\mathcal{S}$. We therefore prove Thm. 6 for *arbitrary* sets $\mathcal{S}''$ .

**Theorem 6 ($\mathrm{eQROM}_f$-$\mathsf{OWTH}$: Finding to Extracting).** *Let* $\mathcal{A}$ *be an algorithm with access to an extractable superposition oracle* $\mathsf{eCO}$ *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *for some function* $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{T}$, *with query depth* $d$ *with respect to* $\mathsf{eCO.RO}$, *and let* $\mathsf{GenInp}$ *be like in Lem. 3. Let* $\mathsf{FIND}$ *be the event that flag register* $F$ *is ever measured to be in state 1 during a call to* $\mathcal{A}$*'s punctured oracle, where the puncturing happens on a set* $\mathcal{S}''$.

*Let* $\mathsf{ExtractSet}$ *be the algorithm that on input* $inp$ *chooses* $i \leftarrow_\$ \{1, \cdots d\}$, *runs* $\mathcal{A}^{\mathsf{eCO}}(inp)$ *until the* $i$*-th query to* $\mathsf{eCO.RO}$; *then measures all query input registers in the computational basis and outputs the set* $\mathcal{S}'$ *of measurement outcomes. Then*

$$\Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO} \setminus \mathcal{S}''}] \leq 4d \cdot \Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}] \ . \quad (28)$$

The proof (given in the full version) directly follows from [AHU19, Th. 2, 'Search in semi-classical oracle'] since [AHU19, Th. 2] gives the bound of Thm. 6 for algorithms $\mathcal{B}$ accessing a semi-classical oracle $O^{\mathsf{SC}}_{S''}$ itself (rather than some oracle punctured on $\mathcal{S}''$). An algorithm $\mathcal{B}^{O^{\mathsf{SC}}_{S''}}$ hence can perfectly simulate $\mathsf{eCO}\backslash\mathcal{S}''$ to $\mathcal{A}$ by simulating $\mathsf{eCO}$ and having the puncturing done by its own oracle $O^{\mathsf{SC}}_{\mathcal{S}''}$.

If the input $inp$ of $\mathcal{A}$ is independent of $\mathcal{S}''$, we also get an extraction bound, an eQROM$_f$ counterpart of [AHU19, Cor. 1], which is proven in the same way.

**Corollary 3** (eQROM$_f$-OWTH**: Extracting independent values).** *If $\mathcal{S}$ and $inp$ are independent, then for any algorithm $\mathcal{A}^{\mathsf{eCO}}$ issuing $q$ many queries to* $\mathsf{eCO.RO}$ *in total,*

$$\Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}''}] \leq 4q \cdot p_{\max} \ ,$$

*where $p_{\max} := \max_{xinX} \Pr_{\mathcal{S}''}[x \in S]$. As a special case, we obtain that*

$$\Pr[\mathsf{FIND} : \mathcal{A}^{\mathsf{eCO}\backslash\{x\}}] \leq 4q|X|^{-1} \ , \tag{29}$$

*for $\mathcal{S}'' = \{x\}$ with uniformly chosen $x \in X$, assuming that $x$ was not needed to generate the input to $\mathcal{A}$.*

### 4.3 From IND-CPA$_{\mathsf{PKE}}$ or OW-CPA$_{\mathsf{PKE}}$ to IND-CPA$_{\mathsf{FO[PKE]}}$

We will now use the OWTH results from Sect. 4.2 to show that the IND-CPA security of $\mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ can be based on the passive security of PKE. In Thm. 7, we base IND-CPA security of $\mathsf{FO}^{\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ on the IND-CPA security of PKE, and we base it on OW-CPA security of PKE in Thm. 8. The obtained bounds are the same as their known plain QROM counterparts.

**Theorem 7.** *Let $\mathcal{A}$ be an* IND-CPA *adversary against* $\mathsf{KEM}^{\perp}_m$ *in the* eQROM$_{\mathsf{Enc}}$*, issuing $q$ many queries to* $\mathsf{eCO.RO}$ *in total, with a query depth of $d$, and $q_E$ many queries to* $\mathsf{eCO.E}$*, where none of them is with its challenge ciphertext. Then there exists an* IND-CPA *adversary $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ against* PKE *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_m}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}})} + 8q|\mathcal{M}|^{-1/2} \ ,$$

*with* $\mathrm{Time}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) = \mathrm{Time}(\mathcal{A}) + \mathrm{Time}(\mathsf{eCO}, q, q_E)$ *and* $\mathrm{QMem}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) = \mathrm{QMem}(\mathcal{A}) + \mathrm{QMem}(\mathsf{eCO}, q, q_E)$.

Note that forbidding extraction queries to $\mathsf{eCO.E}$ on $c^*$ is no limitation in our context: $\mathsf{eCO.E}$ queries are only triggered by an IND-CCA adversary querying its simulated oracle oDecaps′, and oDecaps′ rejects queries on $c^*$ .

A full proof is given in the full version. To summarise the proof, we first define a Game $\mathrm{G}_1$ like the IND-CPA-KEM game , except that encryption randomness $r^* := \mathsf{G}(m^*)$ and honest KEM key $K_0 := \mathsf{H}(m^*)$ are replaced with fresh uniform randomness. In Game $\mathrm{G}_1$, the forwarded KEM key is a uniformly random key either way, the advantage of $\mathcal{A}$ in Game $\mathrm{G}_1$ hence is 0. It remains to bound the distinguishing advantage between the IND-CPA-KEM game and Game $\mathrm{G}_1$.

We apply Thm. 5 which bounds this distinguishing advantage in terms of the probability of event $\mathsf{FIND}_{m^*}$, the event that $m^*$ is detected in the adversary's random oracle queries. To further bound $\Pr[\mathsf{FIND}_{m^*}]$, we use IND-CPA security of PKE to replace $\mathcal{A}$'s ciphertext input $c^*$ with an encryption of an independent message. As $m^*$ now is independent of $\mathcal{A}$'s input, $\mathsf{FIND}_{m^*}$ is highly unlikely for large enough message spaces. (This uses Cor. 3 .)

**Theorem 8.** *For any* IND-CPA *adversary* $\mathcal{A}$ *like in Thm. 7, with a query width of* $w$*, there furthermore exists an* OW-CPA *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) \leq 8d \cdot \sqrt{w \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}})},$$

*with* $\mathrm{Time}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) = \mathrm{Time}(\mathcal{A}) + \mathrm{Time}(\mathsf{eCO}, q, q_E)$ *and* $\mathrm{QMem}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) = \mathrm{Time}(\mathcal{A}) + \mathrm{QMem}(\mathsf{eCO}, q, q_E)$.

Again, a full proof is given in the full version. The proof does exactly the same steps as the one of Thm. 7, up to the point where we bound $\Pr[\mathsf{FIND}_{m^*}]$. To bound $\Pr[\mathsf{FIND}_{m^*}]$, we use Thm. 6 to relate $\Pr[\mathsf{FIND}_{m^*}]$ to the OW-CPA advantage of an algorithm that extracts $m^*$ from $\mathcal{A}$'s oracle queries.

## 5 Characterizing $\mathsf{FFP\text{-}CPA}_{\mathsf{PKE}^{\mathsf{G}}}$

While it may very well be that the maximal success probability in game FFP-CPA for $\mathsf{PKE}^{\mathsf{G}}$ can already be bounded for particular instantiations of $\mathsf{PKE}^{\mathsf{G}}$ without too much technical overhead, even in the $\mathrm{eQROM}_{\mathsf{Enc}}$, this section offers an alternative way to bound this probability: In Thm. 9, we relate the success probability in game FFP-CPA for $\mathsf{PKE}^{\mathsf{G}}$ to two failure-related success probabilities that are easier to analyze. This reduction separates the *computationally hard* problem of exploiting knowledge of the public key to find failing ciphertexts for PKE, from the *statistically hard* problem of searching the QRO G for failing plaintexts $m$ for $\mathsf{PKE}^{\mathsf{G}}$ *without knowledge of the key.*

We begin by defining these two new notions related to decryption failures: In Fig. 8 we define a new variant of the FFP game that differs from game FFP-CPA by providing $\mathcal{A}$ not even with the public key. Since the adversary obtains No Key whatsoever, the game is called FFP-NK, and we define the advantage of an FFP-NK adversary $\mathcal{A}$ against PKE as

$$\mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr[\mathsf{FFP\text{-}NK}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] \ .$$

Furthermore, we define a Find non-generically Failing Plaintext (FFP-NG) game, also in Fig. 8. In this game, the adversary gets a public key $pk_0$ as input and is allowed to issue a single message-randomness pair to a Failure Checking Oracle FCO that is defined either relative to $(sk_0, pk_0)$, the key pair whose public key constitutes $\mathcal{A}$'s input, or relative to a key pair $(sk_1, pk_1)$ which is an independent key pair. We define the advantage of an FFP-NG adversary $\mathcal{A}$ against PKE as

$$\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr[\mathsf{FFP\text{-}NG}^{\mathcal{A}}_{\mathsf{PKE},0} \Rightarrow 1] - \Pr[\mathsf{FFP\text{-}NG}^{\mathcal{A}}_{\mathsf{PKE},1} \Rightarrow 1]\right| \ .$$

| **Game** FFP-NK | **Game** FFP-NG$_b$ | FCO$_b(m;r)$ //one query |
|---|---|---|
| 01 $m \leftarrow \mathcal{A}$ | 05 $(sk_0, pk_0) \leftarrow \mathsf{KG}$ | 09 $c \leftarrow \mathsf{Enc}(pk_b, m; r)$ |
| 02 $(pk, sk) \leftarrow \mathsf{KG}$ | 06 $(sk_1, pk_1) \leftarrow \mathsf{KG}$ | 10 $m' := \mathsf{Dec}(sk_b, c)$ |
| 03 $c := \mathsf{Enc}(pk, m)$ | 07 $b' \leftarrow \mathcal{A}^{\mathsf{FCO}_b}(pk_0)$ | 11 **return** $[\![m \neq m']\!]$ |
| 04 **return** $[\![\mathsf{Dec}(sk, c) \neq m]\!]$ | 08 **return** $[\![b = b']\!]$ | |

**Fig. 8.** Key-independent game FFP-NK for deterministic schemes PKE, and the find non-generically failing ciphertexts games FFP-NG (with $b \in \{0, 1\}$). $\mathcal{A}$ can make at most one query to FCO$_b$.

While the game is formalized as an oracle distinguishing game, $\mathcal{A}$ can only win the game with an advantage over random guessing if it queries oracle FCO on a message-randomness pair that fails with a different probability with respect to key pair $(sk_0, pk_0)$ than with respect to key pair $(sk_1, pk_1)$, a key pair about which $\mathcal{B}$ can only gather information by its query to FCO. We expect this game to be a more palatable target for both provable security and cryptanalysis compared to FFP-CPA$_{\mathsf{PKE}^{\mathsf{G}}}$ or correctness-related games from the existing literature.

**Theorem 9.** *Let* PKE *be a public-key encryption scheme. For any* FFP-CPA *adversary* $\mathcal{A}$ *in the* eQROM$_{\mathsf{Enc}}$ *against* PKE$^{\mathsf{G}}$ *making* $q_R$ *and* $q_E$ *queries to* eCO.RO *and* eCO.E*, respectively, there exist an* FFP-NK *adversary* $\mathcal{C}$ *in the* eQROM$_{\mathsf{Enc}}$ *against* PKE$^{\mathsf{G}}$ *and an* FFP-NG *adversary* $\mathcal{B}$ *against* PKE *with*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{B}) + \mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) \ .$$

*The running time of* $\mathcal{C}$ *is about that of* $\mathcal{A}$*, that of* $\mathcal{B}$ *is* $\mathrm{Time}(\mathcal{B}) = \mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q_{RO}, q_E)$ *and* $\mathrm{QMem}(\mathcal{B}) = \mathrm{QMem}(A) + \mathrm{QMem}(\mathsf{eCO}, q_{RO}, q_E)$.

The proof consists of the following two steps: Apply the FFP-NG definition to argue that the FFP-CPA game's key pair can be replaced with an independent one whose public key is not given to $\mathcal{A}$. After this change, winning means solving FFP-NK for PKE$^{\mathsf{G}}$. The full proof is given in the full version.

### 5.1 Characterizing FFP-NK$_{\mathsf{PKE}^{\mathsf{G}}}$

In the last section, we have related the success probability of an adversary in game FFP-CPA for PKE$^{\mathsf{G}}$ to the success property of an adversary in game FFP-NK for PKE$^{\mathsf{G}}$, in the eQROM$_{\mathsf{Enc}}$. Intuitively, an adversary in game FFP-NK will succeed if it can find oracle inputs $m$ such that $m$ and $r := \mathsf{eCO.RO}(m)$ satisfy the predicate that $(m, r)$ fails with respect to $pk$. To prove the upper bound we provide in Thm. 10, we therefore generically bound the success probability for a certain search problem in Sect. 5.2. While we note that the search bound might be of independent interest, it in particular allows us to characterize the maximal advantage in game FFP-NK in terms of two statistical values for the underlying randomised scheme PKE.We begin with the definitions of $\delta_{\mathrm{ik}}$ and $\sigma_{\delta_{\mathrm{ik}}}$: Below, we define the worst-case decryption error rate $\delta_{\mathrm{ik}}$ *under independent keys*, and the maximal variance of the decryption error rate $\sigma_{\delta_{\mathrm{ik}}}$.

**Definition 2 (worst-case independent-key decryption error rate, maximal decryption error variance).** *We define the* worst-case decryption error rate under independent keys $\delta_{\mathrm{ik}}$ *and the* maximal decryption error variance under independent keys $\sigma_{\delta_{\mathrm{ik}}}$ *of a public-key encryption scheme* $\mathsf{PKE}$ *as*

$$\delta_{\mathrm{ik}} \coloneqq \max_{m\in\mathcal{M}}\big[\Pr_{(sk,pk),r}[(m,r)\ \textit{fails}]\big] = \max_{m\in\mathcal{M}}\mathbb{E}_r\big[\Pr_{(sk,pk)}[(m,r)\ \textit{fails}]\big]\ ,\ \textit{and}$$
$$\sigma^2_{\delta_{\mathrm{ik}}} \coloneqq \max_{m\in\mathcal{M}}\mathbb{V}_r\big[\Pr_{(sk,pk)}[(m,r)\ \textit{fails}]\big]\quad \textit{both for uniformly random } r.$$

We want to stress that $\delta_{\mathrm{ik}}$ *differs from the worst-case term* $\delta_{\mathrm{wc}}$ that was introduced in [HHK17] (there denoted by $\delta$) since $\delta_{\mathrm{wc}}$ is defined by

$$\delta_{\mathrm{wc}} \coloneqq \mathbb{E}_{\mathsf{KG}} \max_{m\in\mathcal{M}} \Pr_{r\leftarrow_{\$}\mathcal{R}}[(m,r)\ \mathrm{fails}]\ .$$

Intuitively, $\delta_{\mathrm{wc}}$ is the best possible advantage of an an adversary, trying to find the message most likely to fail for a given key pair, while for $\delta_{\mathrm{ik}}$, the key pair will be randomly sampled *after* the adversary had made its choice $m$. On a formal level, it is easy to verify that $\delta_{\mathrm{wc}}$ serves as an upper bound for $\delta_{\mathrm{ik}}$.

**Theorem 10.** *Let* $\mathsf{PKE}$ *be a public-key encryption scheme with worst-case independent-key decryption error rate* $\delta_{\mathrm{ik}}$ *and decryption error rate variance* $\sigma_{\delta_{\mathrm{ik}}}$. *For any* $\mathsf{FFP\text{-}NK}$ *adversary* $\mathcal{A}$ *in the* $\mathrm{eQROM}_{\mathsf{Enc}}$ *against* $\mathsf{PKE}^{\mathsf{G}}$, *setting* $C = 304$, *we have that*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{A}) \le \delta_{\mathrm{ik}} + 3\sqrt{C}q\sigma_{\delta_{\mathrm{ik}}} + 2Cq^2\sigma^2_{\delta_{\mathrm{ik}}}\delta_{\mathrm{ik}}(-\log\sqrt{C}q\sigma_{\delta_{\mathrm{ik}}})\ ,$$

The proof is given in the full version.

In the full version, we also give an alternative bound that grows with the logarithm of the number of RO queries, assuming a *Gaussian tail bound* for the decryption error distribution.

### 5.2 Finding large values of a function in the $\mathrm{eQROM}_f$

In this section, we provide the technical results for the $\mathrm{eQROM}_f$ that we need to prove Thm. 10. Throughout this section, $f$ is a fixed function such that $\mathrm{eQROM}_f$ is well-defined. We begin by providing a bound for the success probability of an algorithm in the $\mathrm{eQROM}_f$ that searches for a value $x$ that, together with its oracle value $\mathsf{eCO.RO}(x)$, satisfies a relation $R$. In the lemma below , we will use the quantity $\Gamma_R$ that was defined in Eq. (10) (see page 13).

**Lemma 4.** *Let* $R \subset \mathcal{X}\times\mathcal{Y}$ *be a relation and* $\mathcal{A}^{\mathsf{eCO}}$ *an algorithm with access to* $\mathrm{eQRO}_f$ *from* $\mathcal{X}$ *to* $\mathcal{Y}$ *for some function* $f : \mathcal{X}\times\mathcal{Y}\to\mathcal{T}$, *making* $q$ *queries to* $\mathsf{eCO.RO}$. *Then*

$$\Pr_{x\leftarrow\mathcal{A}^{\mathsf{eCO}}}[R(x,\mathsf{eCO.RO}(x))] \le 152(q+1)^2\Gamma_R|\mathcal{Y}|^{-1}, \tag{30}$$

*independently of the number of queries* $\mathcal{A}$ *makes to* $\mathsf{eCO.E}$. *Here it is understood that* $\mathsf{eCO.RO}$ *is queried once in the very end to determine* $\mathsf{eCO.RO}(x)$.

*Proof.* The only difference between [DFMS21, Proposition 3.5] and Lem. 4 is that $\mathcal{A}$ now additionally has access to $\mathsf{eCO.E}$. The proof is thus the same as for [DFMS21, Proposition 3.5], with the additional observation that queries to $\mathsf{eCO.E}$ commute with the progress measure operator $\mathcal{M}$ for any relation $R$. This is because i) both $\mathcal{M}$ and the operator applied upon an $\mathsf{eCO.E}$ query are controlled unitaries controlling on the database register of the compressed oracle database of the $\mathrm{eQRO}_f$, and ii) the target registers of $\mathcal{M}$ and $\mathsf{eCO.E}$ are disjoint. □

According to Lem. 4, it is hard to search a random oracle, even given extraction access. We will now use Lem. 4 to show that it is also hard to produce an input to the oracle so that the resulting input-output pair has a large value under a function $F$, in expectation. To state a theorem making this intuition precise and quantitative, let $F : X \times Y \to I \subset [0,1]$, and let $I$ be ordered as $I = \{t_1, ..., t_R\}$ with $t_i > t_{i-1}$. The hardness of the task of finding large values is related to a "tail bound" $G(t)$ for the probability of $F(x,r)$ being larger than $t$ .

**Theorem 11.** *Let $F$ and $I$ be as above. Let further $G : [0,1] \to [0,1]$ be non-increasing such that $G(t) \geq \Pr_{r\leftarrow Y}[F(x,r) \geq t]$ for all $x$. Let $C := 304$, $\Delta G(i) := G(t_i) - G(t_{i+1})$ (setting formally $G(t_{R+1}) = 0$), and let $\kappa_q := \min\{i | Cq^2 G(t_i) \leq 1\}$. Then for any algorithm $\mathcal{A}^{\mathsf{eCO}}$ making at most $q \geq 1$ queries to $\mathsf{eCO.RO}$,*

$$\mathbb{E}_{x\leftarrow\mathcal{A}^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x))] \leq t_{\kappa_q} + Cq^2 \sum_{i=\kappa_q+1}^{R} t_i \Delta G(i) \ . \tag{31}$$

$\mathsf{eCO.RO}$ *is queried once in the end to determine* $\mathsf{eCO.RO}(x)$.

*Proof.* Let $x \leftarrow \mathcal{A}^{\mathsf{eCO}}$. We bound

$$\begin{aligned}
&\mathbb{E}\left[F(x, \mathsf{eCO.RO}(x))\right] = \sum_{i=1}^{R} t_i \Pr[F(x, \mathsf{eCO.RO}(x)) = t_i] \\
&= \sum_{i=1}^{R} t_i \left(\Pr[F(x, \mathsf{eCO.RO}(x)) \geq t_i] - \Pr[F(x, \mathsf{eCO.RO}(x)) \geq t_{i+1}]\right) \\
&= t_1 + \sum_{i=2}^{R} \Pr[F(x, \mathsf{eCO.RO}(x)) \geq t_i](t_i - t_{i-1})
\end{aligned}$$

$$\leq t_1 + \sum_{i=2}^{R} \min(1, Cq^2 G(t_i))(t_i - t_{i-1}) = t_{\kappa_q} + Cq^2 \sum_{i=\kappa_q+1}^{R} G(t_i)(t_i - t_{i-1}),$$

where we have used Lem. 4 with the relation $R_{f,\geq t_i}$ defined by $R_{f,\geq t_i}(x,y) :\Leftrightarrow f(x,y) \geq t_i$ in the second-to-last line. We further bound

$$\sum_{i=\kappa_q+1}^{R} G(t_i)(t_i - t_{i-1}) = -G(t_{\kappa_q+1})t_{\kappa_q} + \sum_{i=\kappa_q+1}^{R} t_i \Delta G(i) \leq \sum_{i=\kappa_q+1}^{R} t_i \Delta G(i).$$

□

We provide a corollary for the case where $G$ is given by Chebyshev's inequality.

**Corollary 4.** *Let $F$, $I$, and $C$ be as in Thm. 11, and let the expectation values and variances of $F(x,r)$ for random $r \leftarrow \mathcal{Y}$ be bounded as $\mathbb{E}_r[F(x,r)] \leq \mu$ and $\mathbb{V}_r[F(x,r)] \leq \sigma^2$, respectively. Then, for an algorithm $\mathcal{A}^{\mathsf{eCO}}$ making at most $q \geq 1$ quantum queries to $\mathsf{eCO.RO}$,*

$$\mathbb{E}_{x \leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x))] \leq \mu + 3\sqrt{C}q\sigma + 2Cq^2\sigma^2\mu(-\log(\sqrt{C}q\sigma)). \tag{32}$$

*Proof.* By Chebyshev's inequality, we can set $G(t) = \sigma^2(t-\mu)^{-2}$. We thus obtain $t_{\kappa_q} \leq \sqrt{C}q\sigma + \mu$. We bound

$$\sum_{i=\kappa_q+1}^{R} t_i \Delta G(i) = -\sum_{i=\kappa_q+1}^{R} t_i \int_{t_i}^{t_{i+1}} G'(t)\mathrm{d}t \leq -\int_{t_{\kappa_q}}^{1} tG'(t)\mathrm{d}t \tag{33}$$

$$= 2\sigma^2 \int_{t_{\kappa_q}}^{1} \frac{t}{t-\mu}\mathrm{d}t = 2\sigma^2 \int_{t_{\kappa_q}-\mu}^{1-\mu} \frac{u+\mu}{u}\mathrm{d}u = 2\sigma^2 \left(1 - t_{\kappa_q} + \mu \log \frac{1-\mu}{t_{\kappa_q}-\mu}\right). \tag{34}$$

We arrive at the bound

$$\mathbb{E}_{x \leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x))] \leq \mu + \sqrt{C}q\sigma + 2Cq^2\sigma^2(1 + \mu(\log(1-\mu) - \log(\sqrt{C}q\sigma))).$$

If $\sqrt{C}q\sigma \geq 1$, the claimed bound trivially holds, else $\sqrt{C}q\sigma \geq Cq^2\sigma^2$ and thus

$$\mathbb{E}_{x \leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x))] \leq \mu + 3\sqrt{C}q\sigma + 2Cq^2\sigma^2\mu \log(\log(1-\mu) - \log(\sqrt{C}q\sigma)).$$

□

## 6 Tying everything together

Combining the reductions from Sect. 4.1 and 4.3, we obtain a first corollary that still relies on $\mathsf{FFP\text{-}CPA}$ of $\mathsf{PKE}^{\mathsf{G}}$. Cor. 6 states our main result.

**Corollary 5.** *Let $\mathsf{PKE}$ and $\mathsf{IND\text{-}CCA\text{-}KEM}$ $\mathcal{A}$ against $\mathsf{KEM}_m^{\perp}$ be like in Thm. 3 (on page 14). Then there exist an $\mathsf{IND\text{-}CPA}$ adversary $\mathcal{B}_{\mathsf{IND}}$, a $\mathsf{OW\text{-}CPA}$ adversary $\mathcal{B}_{\mathsf{OW}}$ and an $\mathsf{FFP\text{-}CPA}$ adversary $\mathcal{C}$ against $\mathsf{PKE}^{\mathsf{G}}$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$ such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq \widetilde{\mathrm{Adv}}_{\mathsf{PKE}} + (q_{\mathsf{D}}+1)\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{C}) + \varepsilon_\gamma, \text{ with} \tag{35}$$

$$\widetilde{\mathrm{Adv}}_{\mathsf{PKE}} = \begin{cases} 4 \cdot \sqrt{(d+q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND}})} + \frac{8(q+q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} & \text{or} \\ 8\,(d+q_{\mathsf{D}}) \cdot \sqrt{w \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW}})}. \end{cases} \tag{36}$$

*The additive error term is given by $\varepsilon_\gamma = 24q_{\mathsf{D}}(q_{\mathsf{G}}+4q_{\mathsf{D}})2^{-\gamma/2}$, $\mathcal{C}$ makes $q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}}$ queries to $\mathsf{eCO.RO}$ and $q_{\mathsf{D}}$ to $\mathsf{eCO.E}$. $\mathcal{B}_{\mathsf{IND}}$'s, $\mathcal{B}_{\mathsf{OW}}$'s and $\mathcal{C}$'s running time are bounded as $\mathrm{Time}(\mathcal{B}_{\mathsf{IND/OW}}) = \mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q_{\mathsf{G}}+q_{\mathsf{H}}+q_{\mathsf{D}}) + O(q_{\mathsf{D}})$ and $\mathrm{Time}(\mathcal{C}) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$.*

**Corollary 6.** *Let* $\mathsf{PKE}$ *and* $\mathcal{A}$ *be like in Thm. 3, and let* $\mathsf{PKE}$ *furthermore have worst-case random-key decryption error rate* $\delta_{\mathrm{ik}}$*, decryption error rate variance* $\sigma_{\delta_{\mathrm{ik}}}$ *and decryption error tail envelope* $\tau$*. Set* $C = 304$ *and assume* $\sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}} \leq 1/2$*. Then there exists an* $\mathsf{FFP\text{-}NG}$ *adversary* $\mathcal{C}$ *against* $\mathsf{PKE}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) \leq \widetilde{\mathrm{Adv}}_{\mathsf{PKE}} + (q_{\mathsf{D}}+1)(\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{C}) + \varepsilon_{\delta_{\mathrm{ik}}}) + \varepsilon_{\gamma} \tag{37}$$

*with* $\widetilde{\mathrm{Adv}}_{\mathsf{PKE}}$ *and* $\varepsilon_{\gamma}$ *like in Cor. 5. The additive error term* $\varepsilon_{\delta_{\mathrm{ik}}}$ *is given by* $\varepsilon_{\delta_{\mathrm{ik}}} \leq \delta_{\mathrm{ik}} + (3 + 2\delta_{\mathrm{ik}})\sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}}$*.* $\mathcal{C}$*'s running time is bounded by* $\mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}) + O(q_{\mathsf{D}})$*.*

In the full version, we give an alternative corollary with an $\varepsilon_{\delta_{\mathrm{ik}}}$ that only grows logarithmically with the number of RO queries, assuming a *Gaussian-shaped tail bound* for the decryption error probability distribution.

*Proof.* Cor. 6 follows by combining Cor. 5 with Thms. 9 and 10 from Sect. 5. We simplified error term $\varepsilon_{\delta_{\mathrm{ik}}}$ from Thm. 10 by using the inequality $x^2/\log(x) \leq x$ for $x \leq 1/2$ for $x = \sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}}$, exploiting the mild condition $\sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}} \leq 1/2$[7]. □

The above result has two main advantages over previous ones: i) The additive loss can be much smaller than the additive loss of roughly $q_{\mathsf{G}}^2\delta_{\mathrm{wc}}$ present in all previous bounds. ii) It holds for the explicit rejection variant of the transformation, with bounds that are competitive with previous ones that were limited to implicitly rejecting variants.

## References


AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019*, LNCS, pages 269–295, 2019.

BDK+18. Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *IEEE EuroS&P 2018*, pages 353–367, 2018.

BHH+19. Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *TCC 2019*, LNCS, pages 61–90, 2019.

BS20. Nina Bindel and John M. Schanck. Decryption Failure Is More Likely After Success. In *PQCrypto'20*, pages 206–225. Springer, 2020.

Den03. Alexander W. Dent. A designer's guide to KEMs. In *9th IMA International Conference on Cryptography and Coding*, LNCS, pages 133–151, 2003.

DFMS21. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. Cryptology ePrint Archive, Report 2021/280, 2021.


[7] Without it the bound involving $\sigma_{\delta_{\mathrm{ik}}}$ from Thm. 10 is almost trivial

DRV20. Jan-Pieter D'Anvers, Mélissa Rossi, and Fernando Virdia. (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In *EUROCRYPT 2020*, LNCS, pages 3–33, 2020.

DVV18. Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede. On the impact of decryption failures on the security of LWE/LWR based schemes. Cryptology ePrint Archive, Report 2018/1089, 2018.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, LNCS, pages 537–554, 1999.

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

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017*, LNCS, pages 341–371, 2017.

HKSU20. Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In *PKC 2020*, LNCS, pages 389–422, 2020.

JZC+18. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018*, LNCS, pages 96–125, 2018.

KSS+20. Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *EUROCRYPT 2020*, LNCS, pages 703–728, 2020.

NIS17. NIST. National institute for standards and technology. postquantum crypto project, 2017. http://csrc.nist.gov/groups/ST/post-quantum-crypto/.

SXY18. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018*, LNCS, pages 520–551, 2018.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *CRYPTO 2019*, LNCS, pages 239–268, 2019.