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

A.H. was funded by an NWO VIDI grant (Project No. VI.Vidi.193.066). C.M. was funded by a NWO VENI grant (Project No. VI.Veni.192.159). Date: March 18, 2022

Table of Contents

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 $\mathsf{IND\text{-}CCA}$ secure public-key encryption scheme in the random oracle model. Dent [Den03, Table 5] gave an adoption 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 $\mathsf{FO}_m^\perp$. $\mathsf{FO}_m^\perp$ constructs a KEM from a public-key encryption scheme PKE, and the overall transformation $\mathsf{FO}_m^\perp$ can be described by first modifying PKE to obtain a deterministic scheme PKE^G, and then applying a PKE-to-KEM transformation (called $\mathsf{U}_m^\perp$ in [HHK17]) to PKE^G:

Modified scheme PKE^G. Starting from PKE and a hash function G, deterministic encryption scheme PKE^G is built by letting Enc^G encrypt messages m according to the encryption algorithm Enc of PKE, but using the hash value $\mathsf{G}(m)$ as the random coins for Enc:

$$\mathsf{Enc}^\mathsf{G}(pk, m) := \mathsf{Enc}(pk, m; \mathsf{G}(m)) \ ,$$

Dec^G uses the decryption algorithm Dec of PKE to decrypt a ciphertext c to obtain m', and rejects by returning a failure symbol $\perp$ if c fails to decrypt or m' fails to encrypt back to c. (For the formal definition, see Fig. 3 on page 8).

PKE-to-KEM transformation $\mathsf{U}_m^\perp$. Starting from a deterministic encryption scheme PKE' and a hash function H, key encapsulation algorithm $\mathsf{KEM}_m^\perp := \mathsf{U}_m^\perp[\mathsf{PKE}', \mathsf{H}]$ 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 $\perp$. (For the formal definition, see Fig. 2 on page 7).

Combined PKE-to-KEM transformation $\mathsf{FO}_m^\perp$. The 'full FO' transformation $\mathsf{FO}_m^\perp$ is defined by taking PKE and hash functions G and H, and defining $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}_m^\perp[\mathsf{PKE}^\mathsf{G}, \mathsf{H}]$. While there exists a plethora of variants that differ from $\mathsf{FO}_m^\perp$, it was proven [BHH+19] that security of these variants is either equivalent to or implied by security of $\mathsf{FO}_m^\perp$. To offer a more complete picture, we recap these variants and their relations in Appendix A (page 43). The take-away message is that any security result for $\mathsf{FO}_m^\perp$ also covers its variants.

The role of correctness errors in security proofs for FO. Correctness errors play a role during the proof that an FO-transformed KEM is $\mathsf{IND\text{-}CCA}$ 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 PKE^G. [HHK17] further bounded the maximal advantage in game COR for PKE^G in terms of a statistical worst-case quantity δ_{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]

[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

Intuitively, this notion suffers from two related unnatural features:

- First, it looks rather unnatural to provide any adversary with the secret key, as long as the scheme achieves at least some basic notion of security.[4] In particular, this observation applies to adversaries tasked with finding failing plaintexts, and in fact, this is not a mere issue of aesthetics: If the secret key is given to the adversary, an analysis of this bound cannot make use of computational assumptions without becoming heuristic.[5]
- Second, it seems unnatural that the bound contains a Grover-like search term with regard to δ_{wc}: As IND-CCA adversaries do not have access to the secret key, they can only check whether ciphertexts fail via their *classical* CCA oracle, which should render a Grover search impossible. Furthermore, in both ROM and QROM, it should be the (usually much smaller) number of CCA queries that limits the adversary's ability to search, and not the number of random oracle queries. Hence this bound seems overly conservative as long as the scheme achieves at least some basic notion of security.

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 δ_{wc}.

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 δ_{ik}, the worst-case decryption error rate when the message is picked *independently of the key*, and additional statistical parameters of the probability distributions of decryption failures for fixed message. 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 the FO transformation, the advantage in this game is multiplied

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].

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.

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 $\mathsf{FO}_m^\perp$ 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.

Organisation of this work. Section 2 recalls standard definitions for PKE schemes/KEMs, and the formal definition of $\mathsf{FO}_m^\perp$. Section 3 gives our random oracle model reduction, substantiating the upper half of Fig. 1 in the ROM. Section 5 is the QROM equivalent of Section 3. Since Section 5 uses the extractable quantum random oracle simulation from [DFMS21], we squeeze in a recap of this extension in Section 4 to establish notation and for the reader's convenience. Section 6 analyzes FFP-CPA security of PKE^G further, thereby substantiating the lower half of Fig. 1. Section 7 ties together Section 3/5 with Section 6 by providing corollaries that use concrete bounds for the IND-CCA security of $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. The bounds include a term in γ, the spreadness of PKE. In Section 8, we calculate this term for two easy-to-analyze candidates, HQC.PKE and FrodoPKE.

TL;DR for scheme designers. Section 7 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:

- γ, the spreadness of PKE.
- An upper bound for FFP-NG against PKE.
- Either an upper bound for FFP-NK for PKE^G, in our extended oracle model that allows preimage extractions, or alternatively, two statistical values: δ_{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 δ_{ik}.

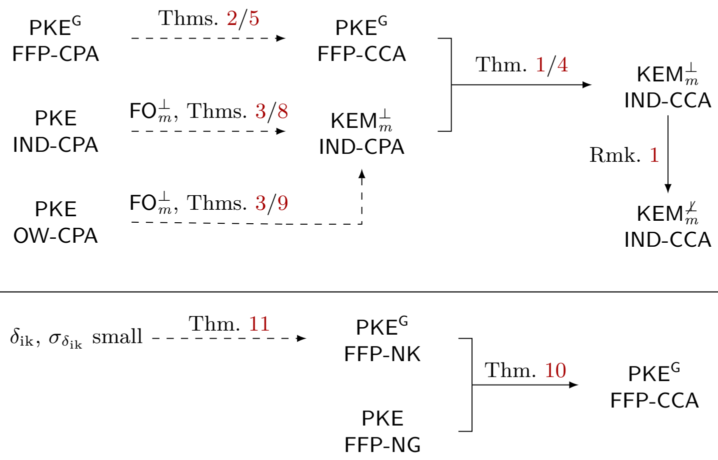

Fig. 1. Summary of our results. Top: "Ths. X/Y" indicates that we provide a ROM theorem X (in Section 3) and a QROM theorem Y (in Section 5). Bottom: Breaking down $\mathsf{FFP\text{-}CPA}$ security of $\mathsf{PKE}^{\mathsf{G}}$ (Section 6). Solid (dashed) arrows indicate tight (non-tight) reductions in the QROM. We want to emphasize that Theorems 2 and 5 have comparably mild tightness loss: The loss is linear in the number of decryption queries. The QROM loss for Theorems 8 and 9 is like the one for previously known reductions.

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 Preliminaries.

For convenience, we recall the formal definition of the Fujisaki-Okamoto transformation with explicit rejection (as already described above) in Section 2.1, and standard definitions for public-key encryption and key encapsulation algorithms in Section 2.2.

For a finite set S, we denote the sampling of a uniform random element x by $x \leftarrow_{\$} S$, and we denote deterministic computation of an algorithm $\mathcal{A}$ on input x by $y := \mathcal{A}(x)$. By $[\![B]\!]$ we denote the bit that is 1 if the Boolean statement B is true, and otherwise 0.

2.1 The Fujisaki-Okamoto transformation with explicit rejection

This section recalls the definition of $\mathsf{FO}_m^{\perp}$. To a public-key encryption scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$, randomness space $\mathcal{R}$, and hash functions $\mathsf{G} : \mathcal{M} \to \mathcal{R}$ and $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$, we associate

$$\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps}) \ .$$

Its constituting algorithms are given in Fig. 2. $\mathsf{FO}_m^{\perp}$ uses the underlying scheme PKE in a derandomized way by using $\mathsf{G}(m)$ as the encryption coins (see line 02) and checks during decapsulation whether the decrypted plaintext does re-encrypt to the ciphertext (see line 06). This building block of $\mathsf{FO}_m^{\perp}$, i.e., the derandomisation of PKE and performing a reencryption check, is incorporated in the following transformation T:

$$\mathsf{PKE}^{\mathsf{G}} := \mathsf{T}[\mathsf{PKE}, \mathsf{G}] := (\mathsf{KG}, \mathsf{Enc}^{\mathsf{G}}, \mathsf{Dec}^{\mathsf{G}}) \ ,$$

with its constituting algorithm given in Fig. 3.

2.2 Security Notions for Public-Key Encryption

We also consider all security games in the (quantum) random oracle model, where PKE and adversary $\mathcal{A}$ are given access to (quantum) random oracles. (How we model quantum access is made explicit in Section 4.)

Definitions for PKE

Definition 1 (γ-spreadness). *We say that* PKE *is γ-spread iff for all key pairs $(pk, sk) \in \mathrm{supp}(\mathsf{KG})$ and all messages $m \in \mathcal{M}$ it holds that*

$$\max_{c \in \mathcal{C}} \Pr[\mathsf{Enc}(pk, m) = c] \leq 2^{-\gamma} \ ,$$

where the probability is taken over the internal randomness Enc.

We also recall two standard security notions for public-key encryption: One-Wayness under Chosen Plaintext Attacks ($\mathsf{OW\text{-}CPA}$) and Indistinguishability under Chosen-Plaintext Attacks ($\mathsf{IND\text{-}CPA}$).

Definition 2 ($\mathsf{OW\text{-}CPA}$, $\mathsf{IND\text{-}CPA}$). *Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme with message space $\mathcal{M}$. We define the $\mathsf{OW\text{-}CPA}$ game as in Fig. 4 and the $\mathsf{OW\text{-}CPA}$ advantage function of an adversary $\mathcal{A}$ against PKE as*

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

Furthermore, we define the 'left-or-right' version of $\mathsf{IND\text{-}CPA}$ by defining games $\mathsf{IND\text{-}CPA}_b$, where $b \in \{0, 1\}$ (also in Fig. 4), and the $\mathsf{IND\text{-}CPA}$ advantage function of an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against PKE (where $\mathcal{A}_2$ has binary output) as

$$\mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) := |\Pr[\mathsf{IND\text{-}CPA}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{IND\text{-}CPA}_1^{\mathcal{A}} \Rightarrow 1]| \ .$$

Standard notions for KEM We now define Indistinguishability under Chosen-Plaintext Attacks ($\mathsf{IND\text{-}CPA}$) and under Chosen-Ciphertext Attacks ($\mathsf{IND\text{-}CCA}$).

$\mathsf{Encaps}(pk)$	$\mathsf{Decaps}(sk, c)$
$m \leftarrow_\$ \mathcal{M}$	$m' := \mathsf{Dec}(sk, c)$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	**if** $m' = \bot$ **or** $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$
$K := \mathsf{H}(m)$	**return** $\bot$
return (K, c)	**else**
	return $K := \mathsf{H}(m')$

Fig. 2. Key encapsulation mechanism $\mathsf{KEM}_m^{\bot} = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$, obtained from $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ by setting $\mathsf{KEM}_m^{\bot} := \mathsf{FO}_m^{\bot}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$.

$\underline{\mathsf{Enc}^{\mathsf{G}}(pk)}$	$\underline{\mathsf{Dec}^{\mathsf{G}}(sk, c)}$
$m \leftarrow_\$ \mathcal{M}$	$m' := \mathsf{Dec}(sk, c)$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	**if** $m' = \bot$ **or** $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$
return c	**return** $\bot$
	else
	return m'

Fig. 3. Derandomized PKE scheme $\mathsf{PKE}^{\mathsf{G}} = (\mathsf{KG}, \mathsf{Enc}^{\mathsf{G}}, \mathsf{Dec}^{\mathsf{G}})$, obtained from $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ by encrypting a message m with randomness $\mathsf{G}(m)$ for a random oracle G, and incorporating a re-encryption check during $\mathsf{Dec}^{\mathsf{G}}$.

Game $\mathsf{OW\text{-}CPA}$	**Game** $\mathsf{IND\text{-}CPA}_b$
$(pk, sk) \leftarrow \mathsf{KG}$	$(pk, sk) \leftarrow \mathsf{KG}$
$m^* \leftarrow_\$ \mathcal{M}$	$(m_0^*, m_1^*, \mathrm{st}) \leftarrow \mathcal{A}_1(pk)$
$c^* \leftarrow \mathsf{Enc}(pk, m^*)$	$c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$
$m' \leftarrow \mathcal{A}(pk, c^*)$	$b' \leftarrow \mathcal{A}_2(pk, c^*, \mathrm{st})$
return $[\![m' = m^*]\!]$	**return** b'

Fig. 4. Games $\mathsf{OW\text{-}CPA}$ and $\mathsf{IND\text{-}CPA}_b$ for PKE.

Definition 3 ($\mathsf{IND\text{-}CPA}$, $\mathsf{IND\text{-}CCA}$). *Let* $\mathsf{KEM} = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$ *be a key encapsulation mechanism with key space* $\mathcal{K}$. *For* $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, *we define* $\mathsf{IND\text{-}ATK\text{-}KEM}$ *games as in Fig. 5, where*

$$\mathsf{O}_{\mathsf{ATK}} := \begin{cases} - & \mathsf{ATK} = \mathsf{CPA} \\ \textsc{oDecaps} & \mathsf{ATK} = \mathsf{CCA} \end{cases} .$$

We define the $\mathsf{IND\text{-}ATK\text{-}KEM}$ *advantage function of an adversary* $\mathcal{A}$ *against* KEM *as*

$$\mathrm{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}ATK\text{-}KEM}}(\mathcal{A}) := |\Pr[\mathsf{IND\text{-}ATK\text{-}KEM}^{\mathcal{A}} \Rightarrow 1] - 1/2| \enspace .$$

Game $\mathsf{IND\text{-}ATK\text{-}KEM}$	$\underline{\textsc{oDecaps}(c \neq c^*)}$
$(pk, sk) \leftarrow \mathsf{KG}$	$K := \mathsf{Decaps}(sk, c)$
$b \leftarrow_\$ \{0, 1\}$	**return** K
$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
$K_1^* \leftarrow_\$ \mathcal{K}$	
$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ATK}}}(pk, c^*, K_b^*)$	
return $[\![b' = b]\!]$	

Fig. 5. Game $\mathsf{IND\text{-}ATK\text{-}KEM}$ for KEM, where $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$ and $\mathsf{O}_{\mathsf{ATK}}$ is defined in Definition 3.

3 ROM reduction

This section substantiates the upper half of Fig. 1 in the random oracle model. 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 PKE^G of PKE (as defined in Fig. 3 on page 8). 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 4 (also see Fig. 6). For $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$, 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*. We will first show in Theorem 1 that any attacker against the IND-CCA-KEM security of $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ can be used to construct an IND-CPA-KEM attacker against $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ and an attacker against the correctness of PKE^G that has access to oDecrypt, i.e., an attacker that succeeds in game FFP-CCA. The maximum winning probability in the FFP-CCA game is still quite an unwieldy object. In particular, clever strategies have been devised to make adaptive use of a decryption oracle towards finding failing plaintexts. Fortunately, we can use the same strategy underlying our simulation of oDecaps once more to show in Theorem 2 that any successful FFP-CCA adversary can be used to construct an adversary succeeding in the FFP-CPA game, meaning that it is sufficient to analyse the success probability of attackers trying to come up with failing plaintexts, having nothing on their hands but the public key. It is then shown how [HHK17] can be used to argue that IND-CPA-KEM security of $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ can be based on either OW-CPA or IND-CPA security of PKE, with the latter implication being tight up to a factor of 3. Lastly, we discuss that the main result of this section also works if we consider the implicitly rejecting variant $\mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ instead of $\mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ (see Remark 1). 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.

Definition 4 (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. 6, where*

$$\mathsf{O}_{\mathsf{ATK}} := \begin{cases} - & \mathsf{ATK} = \mathsf{CPA} \\ \text{oDecrypt} & \mathsf{ATK} = \mathsf{CCA} \end{cases} .$$

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

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

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 begin by introducing two simulations of the Decaps oracle, oracle oDecaps′ and a variant oDecaps″ of oDecaps′. oDecaps″ extracts failing plaintexts from adversarial decapsulation queries, and is simulatable by FFP adversaries with access to the decryption oracle oDecrypt for PKE^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 G′ that keeps track of all issued queries and compiles $\mathcal{L}$. The original Decaps oracle and its simulations are defined in Fig. 7,

```
Game FFP-ATK                         oDECRYPT(c)
01 (pk, sk) ← KG                     06 m := Dec(sk, c)
02 m ← A^{O_ATK, G}(pk)              07 return m
03 c := Enc(pk, m)
04 m' := Dec(sk, c)
05 return [[m' ≠ m]]
```

Fig. 6. Games FFP-ATK for a deterministic PKE, where ATK $\in$ {CPA, CCA}. $\mathsf{O}_{\mathsf{ATK}}$ is the decryption oracle present in the respective IND-ATK-KEM game (see Definition 4) and G is a random oracle, provided if it is used in the definition of PKE.

```
oDECAPS(c)                       oDECAPS'(c ≠ c*)                 oDECAPS''(c ≠ c*)
01 m' := Dec(sk, c)              14 m := L_G^{-1}(c)              26 m := L_G^{-1}(c)
02 if m' = ⊥                     15 if m = ⊥                      27 m' := oDECRYPT(c)
03   return K:=⊥                 16   return K:=⊥                 28 if m ≠ ⊥ and m ≠ m'
04 else                          17 else                          29   L_FAIL := L_FAIL ∪ {m}
05   c' := Enc(pk, m'; G(m'))    18   return K := H(m)            30 if m = ⊥
06   if c ≠ c'                                                    31   return K:=⊥
07     return ⊥                                                   32 else
08   else                        oDECRYPT(c ≠ c*)                 33   return K := H(m)
09     return H(m')              19 m' := Dec(sk, c)
                                 20 if m' = ⊥
                                 21   return ⊥
G'(m)                            22 else
10 r := G(m)                     23   if Enc(pk, m'; G(m')) ≠ c
11 c := Enc(pk, m; r)            24     return ⊥
12 L_G := L_G ∪ {(m, c)}         25   else return m'
13 return r
```

Fig. 7. Simulation oDECAPS′ of oracle oDECAPS for $\mathsf{KEM}_m^{\perp}$, failing-plaintext-extracting version oDECAPS′′ of oDECAPS′, and decryption oracle oDECRYPT for $\mathsf{PKE}^{\mathsf{G}}$. Oracles oDECAPS′ and oDECAPS′′ use in lines 14 and 26 the notation introduced in Equation (1). Note that G′ only differs from G by compiling list $\mathcal{L}_{\mathsf{G}}$ (which we assume to be initialized to $\emptyset$).

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

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

The simulation oDECAPS′ can, however, only *reverse* encryptions that were already computed by the adversary (with a query to oracle G′) *before* their query to oracle oDECAPS′, which is where the spreadness of PKE comes into play: If γ 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 **($\mathsf{FO}_m^{\perp}$[PKE] IND-CPA and $\mathsf{PKE}^{\mathsf{G}}$ FFP-CCA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}$[PKE] IND-CCA).** *Let* PKE *be a (randomised)* PKE *scheme that is γ-spread, and let* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. *Let $\mathcal{A}$ be an*

$\mathsf{IND\text{-}CCA\text{-}KEM}$-*adversary (in the ROM) against* $\mathsf{KEM}^{\perp}_{m}$*, making at most* q_{D} *many queries to its decapsulation oracle* oDecaps*. Then there exist an* $\mathsf{IND\text{-}CPA\text{-}KEM}$ *adversary* $\tilde{\mathcal{A}}$ *and an* $\mathsf{FFP\text{-}CCA}$ *adversary* $\mathcal{B}$ *against* $\mathsf{PKE}^{\mathsf{G}}$ *such that*

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

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

Proof. Let $\mathcal{A}$ be an adversary against $\mathsf{KEM}^{\perp}_{m}$. We define $\tilde{\mathcal{A}}$ as the $\mathsf{IND\text{-}CPA\text{-}KEM}$ adversary against $\mathsf{KEM}^{\perp}_{m}$ that runs $b' \leftarrow \mathcal{A}^{\mathsf{G}',\mathsf{H},\mathrm{oDecaps}'}$ and returns b'. We furthermore define our $\mathsf{FFP\text{-}CCA}$ adversary $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$ as follows: $\mathcal{B}$ runs $\mathcal{A}^{\mathsf{G}',\mathsf{H},\mathrm{oDecaps}''}$, using its own $\mathsf{FFP\text{-}CCA}$ oracle oDecrypt to simulate oDecaps$''$. As soon as oDecaps$''$ adds a plaintext m to $\mathcal{L}_{\mathrm{FAIL}}$, $\mathcal{B}$ aborts $\mathcal{A}$ and returns m. If $\mathcal{A}$ finishes and $\mathcal{L}_{\mathrm{FAIL}}$ is still empty, $\mathcal{B}$ returns $\perp$.

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 $\mathsf{Decaps}(sk, c) \neq \mathrm{oDecaps}'(c)$. We bound

$$\frac{1}{2} + \mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}] \tag{3}$$

$$= \Pr[\mathcal{A} \text{ wins} \wedge \neg\mathsf{DIFF}] + \Pr[\mathcal{A} \text{ wins} \wedge \mathsf{DIFF}] \tag{4}$$

$$= \Pr[\tilde{\mathcal{A}} \text{ wins} \wedge \neg\mathsf{DIFF}] + \Pr[\mathcal{A} \text{ wins} \wedge \mathsf{DIFF}] \tag{5}$$

$$\leq \Pr[\tilde{\mathcal{A}} \text{ wins}] + \Pr[\mathsf{DIFF}] \tag{6}$$

$$= \frac{1}{2} + \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\tilde{\mathcal{A}}) + \Pr[\mathsf{DIFF}]. \tag{7}$$

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 G', but fails the reencryption check performed by oDecaps(c). Since the latter means that either $m' := \mathsf{Dec}(sk, c) = \perp$ or that $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, this cases 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 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 G'.

In either of the former two cases, 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 $\mathsf{FFP\text{-}CCA}$ 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 G' on m yet. Whenever DIFF occurs, $\mathcal{B}$ succeeds unless GUESS occurs. In formulae,

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

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 Theorem 2, in which we revisit the $\mathsf{FFP\text{-}CCA}$ attacker $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$, and where we will simulate $\mathcal{B}$'s oracle oDecrypt via an oracle oDecrypt′ (see Fig. 8). Therefore, we generalize the definition of event GUESS accordingly.

Lemma 1. *Let* PKE *be* γ*-spread, and let* $\mathcal{A}$ *be an adversary expecting random oracles* G, H *as well as either a decapsulation oracle* oDecaps *for* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *or a decryption oracle* oDecrypt *for* $\mathsf{PKE}^{\mathsf{G}}$*, issuing at most* q_D *queries to the latter. When run with* 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*

$$\Pr[\mathsf{GUESS}] \leq q_D \cdot 2^{-\gamma}. \tag{8}$$

Proof. The event GUESS, i.e. the case that oDecaps′ (oDecrypt′) rejects on a ciphertext c where oDecaps (oDecrypt) does not, requires that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$ for $m := \mathsf{Dec}(sk, c)$, and that G' was not yet queried on m. Let c be any ciphertext queried by the adversary for which oDecaps does not reject, and let $m := \mathsf{Dec}(sk, c)$. We can bound

$$\begin{aligned}\Pr\left[\textsc{oDecaps}'(c) = \perp\right] &\leq \Pr[\mathsf{Enc}(pk, m, \mathsf{G}'(m)) = c \wedge \mathsf{G}' \text{ not yet queried on } m] \\ &\leq \Pr_{r \leftarrow_\$ \mathcal{R}}[\mathsf{Enc}(pk, m; r) = c] \leq 2^{-\gamma},\end{aligned}$$

where the penultimate step used that G' has the same distribution as random oracle G and that $\mathsf{G}(m)$ has not yet been sampled, and the last step used that PKE scheme is γ-spread. Applying a union bound, we conclude that

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

□

So far, we have shown that whenever an $\mathsf{IND\text{-}CCA}$ 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 $\mathsf{FFP\text{-}CCA}$ decryption oracle oDecrypt for $\mathsf{PKE}^{\mathsf{G}}$. We now proceed by showing that oDecrypt can be simulated via oracle oDecrypt′ (see Fig. 8) without the secret key, thereby being able to construct an $\mathsf{FFP\text{-}CPA}$ adversary from any $\mathsf{FFP\text{-}CCA}$ adversary that succeeds with the same probability up to (at most) a multiplicative factor equal to the number of decryption queries the $\mathsf{FFP\text{-}CCA}$ adversary makes.

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

$$\mathrm{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CCA}}(\mathcal{B}) \leq (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CPA}}\left(\tilde{\mathcal{B}}\right) + q_{\mathsf{D}} \cdot 2^{-\gamma} \ . \tag{9}$$

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

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 $\mathsf{FFP\text{-}CCA}$ game to $\mathcal{B}$ and use $\mathcal{B}$'s output to win its own $\mathsf{FFP\text{-}CPA}$ game. The simulation will fail if either GUESS happens (with probability at most $q_D \cdot 2^{-\gamma}$ due to Lemma 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

$\underline{\text{oDecrypt}'(c)}$	$\underline{\tilde{\mathcal{B}}^{\mathsf{G}}}$
$m := \mathcal{L}_{\mathsf{G}}^{-1}(c)$	$i \leftarrow_{\$} \{1, ..., q_{\mathsf{D}} + 1\}$
return m	**if** $i < q_{\mathsf{D}} + 1$
	Run $\mathcal{B}^{\mathsf{G}', \text{oDecrypt}'}(pk)$ until i-th query c_i to oDecrypt′
$\underline{\mathsf{G}'(m)}$	$m := \mathcal{L}_{\mathsf{G}}^{-1}(c_i)$
$c := \mathsf{Enc}(m; \mathsf{G}(m))$	**else**
$\mathcal{L}_{\mathsf{G}} := \mathcal{L}_{\mathsf{G}} \cup \{(m, c)\}$	$m \leftarrow \mathcal{B}^{\mathsf{G}', \text{oDecrypt}'}(pk)$
return $\mathsf{G}(m)$	**return** m

Fig. 8. Simulation oDecrypt′ of oracle oDecrypt for $\mathsf{PKE}^{\mathsf{G}}$, which is defined analogously to oDecaps′ (see Figure 7), and FFP-CPA adversary $\tilde{\mathcal{B}}$. For the reader's convenience, we repeat the definition of G'.

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_{D} many queries to oracle oDecrypt. Consider the adversary $\tilde{\mathcal{B}}^{\mathsf{G}}$ in Fig. 8. $\tilde{\mathcal{B}}$ 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 G', $\tilde{\mathcal{B}}$ uses its oracle G. Simulation oDecrypt′ is defined in Fig. 8 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 Theorem 1, and again using Lemma 1, we obtain

$$\text{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CCA}}(\mathcal{B}) \leq \Pr[\mathcal{B} \text{ wins } \wedge \neg\mathsf{DIFF}] + \Pr[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}] + q_{\mathsf{D}} \cdot 2^{-\gamma}. \tag{10}$$

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.

$$\text{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CPA}}(\tilde{\mathcal{B}}) = \frac{1}{q_{\mathsf{D}} + 1} \cdot (\Pr[\mathcal{B} \text{ wins } \wedge \neg\mathsf{DIFF}] + \Pr[\mathsf{DIFF} \wedge \neg\mathsf{GUESS}]) \tag{11}$$

Combining Equations (10) and (11) yields the desired bound. □

Combining Theorems 1 and 2, we obtain the following straightforwardly.

Corollary 1 ($\mathsf{FO}_m^{\perp}$[PKE] IND-CPA **and** $\mathsf{PKE}^{\mathsf{G}}$ FFP-CPA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}$[PKE] IND-CCA**).** *Let* PKE *be* γ*-spread, and let* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. *Let* $\mathcal{A}$ *be an* IND-CCA-KEM *adversary (in the ROM) against* $\mathsf{KEM}_m^{\perp}$*, issuing at most* q_{G} *many queries to its oracle* G*,* q_{H} *many queries to its oracle* H*, and at most* q_{D} *many queries to its decapsulation oracle* oDecaps*. Then there exist an* IND-CPA-KEM *adversary* $\tilde{\mathcal{A}}$ *and an* FFP-CPA *adversary* $\mathcal{B}$ *such that*

$$\text{Adv}_{\mathsf{KEM}_m^{\perp}}^{\mathsf{IND\text{-}CCA\text{-}KEM}}(\mathcal{A}) \leq \text{Adv}_{\mathsf{KEM}_m^{\perp}}^{\mathsf{IND\text{-}CPA\text{-}KEM}}(\tilde{\mathcal{A}}) + (q_{\mathsf{D}} + 1) \cdot \text{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CPA}}(\mathcal{B}) + 2q_{\mathsf{D}} \cdot 2^{-\gamma} \ . \tag{12}$$

Adversary $\tilde{\mathcal{A}}$ *makes* q_{G} *queries to* G *and* $q_{\mathsf{H}} + q_{\mathsf{D}}$ *queries to* H*, adversary* $\mathcal{B}$ *makes* q_{G} *queries to* G*, and both 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 Corollary 1 can be proven without the factor of 2, when directly analyzing the composition of the reductions from Theorems 1 and 2.

Next, we observe in Theorem 3 that $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ can be based on passive security of PKE. While Theorem 3 is implicitly contained in [HHK17], we make explicit in Appendix B how it can be easily obtained.

Theorem 3 **(PKE $\mathsf{OW\text{-}CPA}$ or $\mathsf{IND\text{-}CPA}$ $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CPA}$).** *Let* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ *for some PKE scheme* PKE. *For any* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}_m^{\perp}$*, issuing at most* q_{G} *many queries to its oracle* G *and* q_{H} *many queries to its oracle* H*, there exist an* $\mathsf{OW\text{-}CPA}$ *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *and an* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ *of roughly the same running time such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq (q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}})$$

and

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{2 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}}) + 1}{|\mathcal{M}|} \ .$$

Combining Corollary 1 and Theorem 3, we obtain the following straightforward

Corollary 2 **(PKE $\mathsf{OW\text{-}CPA}$ or $\mathsf{IND\text{-}CPA}$ and $\mathsf{PKE}^{\mathsf{G}}$ $\mathsf{FFP\text{-}CPA}$ $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CCA}$).** *Let* PKE *be a (randomized)* PKE *scheme that is* γ*-spread, and let* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$. *Let* $\mathcal{A}$ *be an* $\mathsf{IND\text{-}CCA\text{-}KEM}$ *adversary (in the ROM) against* $\mathsf{KEM}_m^{\perp}$*, making at most* q_{RO} *many queries to its random oracles* G *and* H*, and* q_{D} *many queries to its decapsulation oracle* ODECAPS. *Then there exist a* $\mathsf{OW\text{-}CPA}$ *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *and an* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq &(q_{\mathsf{RO}} + q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) \\ &+ (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) + 2q_{\mathsf{D}} \cdot 2^{-\gamma}\end{aligned}$$

and

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{2 \cdot (q_{\mathsf{RO}} + q_{\mathsf{D}}) + 1}{|\mathcal{M}|} \\ + (q_{\mathsf{D}} + 1) \cdot \mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) + 2q_{\mathsf{D}} \cdot 2^{-\gamma}.\end{aligned}$$

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

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 δ_{wc} (there denoted by δ) present in [HHK17] by $\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B})$, and having an additional term in γ even for $\mathsf{KEM}_m^{\not\perp}$. We believe that the additional γ-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 Section 6.

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 Corollary 2 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 $\mathsf{IND\text{-}CCA\text{-}KEM}$ attacker $\mathcal{A}$ against $\mathsf{KEM}_m^{\not\perp}$, there exists an $\mathsf{IND\text{-}CCA\text{-}KEM}$ 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 Corollary 2 does not contain any terms relative to $\mathsf{KEM}_m^{\perp}$ itself, it only contains terms relative to the underlying schemes PKE and $\mathsf{PKE}^{\mathsf{G}}$.

4 Compressed oracles and extraction

We want to generalize the ROM results obtained in Section 3 to the QROM. To this end, we will use an extension of the compressed oracle technique [Zha19] that was introduced in [DFMS21] and that we will now quickly recap. To describe the technique, we start with the observation that for each input value x, its oracle value $\mathsf{O}(x)$ is a uniformly distributed random variable that can equivalently be sampled by measuring a uniform superposition in the computational basis. 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 O has finished. In [DFMS21], this oracle simulation was generalized to obtain an *extractable* oracle simulator 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 $f : X \times Y \to T$. Informally, $\mathsf{eCO.E}_f$ takes as input a classical value t. Consider the classical procedure of going through a lexicographically ordered list of lazy-sampled input output pairs (x, y) and outputting the first one such that $f(x, y) = t$. $\mathsf{eCO.E}_f$ performs the quantum analogue of that: a measurement that partially collapses the oracle database, just enough so that the classical procedure would yield one particular outcome x for all parts of the superposition. After the measurement, D is thus in a state such that the superposition held in database entry D_x only contains possibilities y for $\mathsf{eCO.RO}(x)$ such that $f(x, y) = t$, and no entry $D_{x'}$ for any $x' < x$ will have any possibilities y' left such that also $f(x', y') = t$. 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 eQRO_f, and a proof that uses an eQRO_f will be called *a proof in the* eQROM_f.

We will now make this description more formal, closely following notation and conventions from [DFMS21]. Like in [DFMS21], we keep the formalism as simple as possible by describing 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 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 input value $x \in \{0,1\}^m$, 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 one of the registers 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

$$O_{XYD} |x\rangle_X = |x\rangle_X \otimes \left(F_{D_x} \mathsf{CNOT}^{\otimes n}_{D_x:Y} F_{D_x}\right), \tag{13}$$

  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 O: apply $F^{\otimes 2^m}$ to D and measure it to obtain the function table of O.

In section 5.2, we will use the superposition oracle to analyze algorithms that make parallel (quantum) queries to a random oracle. For a standard quantum oracle for a function H, an algorithm that makes w parallel queries sends $2w$ quantum regisers X_i, Y_i, $i = 1, ..., w$ to the oracle. The query is then processed by applying the oracle unitary U_H to each pair X_i, Y_i. We can think of this parallel-query oracle as being implemented by a simulator with query access to the non-parallel oracle for H: upon input regisers X_i, Y_i, $i = 1, ..., w$ the simulator sends the register pairs X_i, Y_i to its own oracle sequentially. Using this trivial reformulation, it is clear how parallel queries can be handled when H is a random function and the oracle for H is simulated using the compressed oracle.

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.

$$\Sigma^{t,x} = \bigotimes_{x'<x} \bar{\Pi}^{t,x'}_{D'_x} \otimes \Pi^{t,x}_{D_x}, \quad \text{with} \quad \Pi^{t,x} = \sum_{\substack{y\in\{0,1\}^n:\\ f(x,y)=t}} |y\rangle\langle y| \tag{14}$$

and $\bar{\Pi} = \mathbb{1} - \Pi$. The remaining projector corresponds to the case where no register contains such a y, i.e.

$$\Sigma^{t,\emptyset} = \bigotimes_{x'\in\{0,1\}^m} \bar{\Pi}^{t,x'}_{D'_x}. \tag{15}$$

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

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 a classical oracle interface that, on input t, applies $\mathcal{M}^t$ to eCO's internal state (i.e. the state of the compressed oracle) and returns the result. The simulator eCO has several useful properties that were characterized in [DFMS21, Theorem 3.4], for convenience included below. These characterisations are in terms of the quantity

$$\begin{aligned} \Gamma(f) &= \max_t \Gamma_{R_{f,t}}, \text{ with} \\ R_{f,t}(x,y) &:\Leftrightarrow f(x,y) = t \text{ and} \\ \Gamma_R &:= \max_x |\{y \mid R(x,y)\}|. \end{aligned} \tag{16}$$

For $f = \mathsf{Enc}(\cdot;\cdot)$, the encryption function of a PKE that takes as first input a message m and as second input an encryption randomness r, we have $\Gamma(f) = 2^{-\gamma}|\mathcal{R}|$ if PKE is γ-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.

We now state the parts of [DFMS21, Theorem 3.4] that we will use in our proofs.

Lemma 2 (Part of theorem 3.4 in [DFMS21]). *The extractable RO simulator* eCO *described above, with interfaces* $\mathsf{eCO.RO}$ *and* $\mathsf{eCO.E}$*, satisfies the following properties.*

1. *If* $\mathsf{eCO.E}$ *is unused,* eCO *is perfectly indistinguishable from a random oracle.*

2.a *Any two subsequent independent queries to* $\mathsf{eCO.RO}$ *commute. In particular, two subsequent* classical $\mathsf{eCO.RO}$-*queries with the same input* x *give identical responses.*

2.b *Any two subsequent independent queries to* $\mathsf{eCO.E}$ *commute. In particular, two subsequent* $\mathsf{eCO.E}$-*queries with the same input* t *give identical responses.*

2.c *Any two subsequent independent queries to* $\mathsf{eCO.E}$ *and* $\mathsf{eCO.RO}$ $8\sqrt{2\Gamma(f)/2^n}$-*almost-commute.*

Game FFP-ATK

$(pk, sk) \leftarrow \mathsf{KG}$
$m \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ATK}}, \mathsf{eCO}}(pk)$
$c := \mathsf{Enc}(pk, m)$
$m' := \mathsf{Dec}(sk, c)$
return $[\![m' \neq m]\!]$

oDecrypt(c)

$m := \mathsf{Dec}(sk, c)$
return m

Fig. 9. Games FFP-ATK for a deterministic PKE, where ATK $\in \{\mathsf{CPA}, \mathsf{CCA}\}$, in the eQROM$_f$. Like in its classical counterpart (see Fig. 6, page 10), $\mathsf{O}_{\mathsf{ATK}}$ is the decryption oracle present in the respective IND-ATK-KEM game (see Definition 4 on page 9). The only difference is that the random oracle G is now modelled as an extractable superposition oracle eCO.

Furthermore, the total runtime and quantum memory footprint of eCO*, when using the sparse representation of the compressed oracle, are bounded as*

$$\mathrm{Time}(\mathsf{eCO}, q_{RO}, q_E) = O\big(q_{RO} \cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2\big), \text{ and}$$
$$\mathrm{QMem}(\mathsf{eCO}, q_{RO}, q_E) = O\big(q_{RO}\big).$$

where q_E *and* q_{RO} *are the number of queries to* eCO.E *and* eCO.RO*, respectively.*

5 QROM reduction

In this section, we generalize the reductions from Section 3 to the quantum-accessible random oracle model. To do so, we give in Fig. 10 the quantum analogues of the simulated decapsulation oracles oDecaps$'$ and oDecaps$''$ from Fig. 7, which were (essentially) developed in [DFMS21]. We have to adapt our simulations since the ROM simulations from Fig. 7 use book-keeping techniques and therefore cannot be easily implemented in the standard QROM. Instead, we use the formalism described in Section 4, i.e., we use a simulation of a quantum-accessible random oracle and *make use of the additional extraction interface* eCO.E: While the simulations in Fig. 7 had access to a list $\mathcal{L}_{\mathsf{G}}$ that could be used to extract potential ciphertext preimages, the simulations in Fig. 10 can now extract them by accessing extractor eCO.E (see lines 12 and 17). The rest of the simulation is exactly as before. Using the notation from Section 4, we denote the modelling of the ROM as extractable by 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.

While Section 3 concluded by showing in Theorem 3 how to base IND-CPA-KEM security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ on passive security of PKE in the ROM, we need to develop an additional tool to do the same in the eQROM$_{\mathsf{Enc}}$. Therefore, we split this section as follows: Section 5.1 ends with IND-CCA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ being based on IND-CPA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ and FFP-CPA security of $\mathsf{PKE}^{\mathsf{G}}$. Note that the notions on which we base IND-CCA security are now in the eQROM$_{\mathsf{Enc}}$.We give the eQROM$_f$ definition of FFP-ATK in Fig. 9. Section 5.2 develops the necessary eQROM$_{\mathsf{Enc}}$ tools to further analyze IND-CPA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. Concretely, Section 5.2 provides an eQROM$_{\mathsf{Enc}}$-compatible variant of the one-way to hiding (OWTH) lemma for semi-classical oracles as introduced in [AHU19]. Intuitively, the eQROM$_{\mathsf{Enc}}$-OWTH lemma states that input depending on particular random oracle values eCO.RO(x) (like, e.g., $\mathsf{G}(m^*)$) can be replaced with input that replaced all involved oracle values with fresh uniform randomness. The change goes unnoticed unless one of the x can be detected in the oracle queries. Section 5.2 is given in a general way and might prove to be of independent interest. Equipped with the results from Section 5.2, we show in Section 5.3 that *also in the* eQROM$_{\mathsf{Enc}}$, IND-CPA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ can be based on passive security of PKE.

```
oDecaps(c ≠ c*)
01 m' := Dec(sk, c)
02 if m' = ⊥
03    return K:=⊥
04 else
05    c' := Enc(pk, m'; G(m'))
06    if c ≠ c'
07       return ⊥
08    else
09       return H(m')

G', input registers X, Y
10 Apply eCO.RO_{XYD}
11 return registers XY

oDecaps'(c ≠ c*)
12 m ← eCO.E(c)
13 if m = ⊥
14    return ⊥
15 else
16    return H(m)

oDecaps''(c ≠ c*)
17 m ← eCO.E(c)
18 m' := oDecrypt(c)
19 if m ≠ ⊥and m ≠ m'
20    L_FAIL := L_FAIL ∪ {m}
21 if m = ⊥
22    return ⊥
23 else
24    return H(m)

oDecrypt(c)
25 m' := Dec(sk, c)
26 if m' = ⊥
27    return ⊥
28 else
29    if Enc(pk, m'; G(m')) ≠ c
30       return ⊥
31    else
32       return m'
```

Fig. 10. Simulated and failing-plaintext-extracting versions of the decapsulation oracle oDecaps for $\mathsf{FO}^{\perp}_{m}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$, using the extractable QRO simulator eCO from [DFMS21] (see Section 4). The simulations of oDecaps are exactly like the ROM ones in Fig. 7 except for how they extract ciphertext preimages in lines 12 and 17. We assume eCO to be freshly initialized before oDecaps′ or oDecaps′′ is used for the first time in a security game, and extraction interface $\mathsf{eCO.E}$ is defined with respect to function $f = \mathsf{Enc}(pk, \cdot; \cdot)$, where Enc is the encryption algorithm of PKE.

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

We begin by proving a quantum analogue of Theorem 1.

Theorem 4 **($\mathsf{FO}^{\perp}_{m}[\mathsf{PKE}]$ IND-CPA and $\mathsf{PKE}^{\mathsf{G}}$ FFP-CCA $\overset{\text{eQROM}_{\mathsf{Enc}}}{\Rightarrow}$ $\mathsf{FO}^{\perp}_{m}[\mathsf{PKE}]$ IND-CCA).** *Let* PKE *be a (randomized)* PKE *that is* γ*-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_{D} *many queries to its decapsulation oracle* oDecaps*, and making* q_{G}*,* q_{H} *queries to its respective random oracles .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* $\text{eQROM}_{\mathsf{Enc}}$*, such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) \leq &\mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\tilde{\mathcal{A}}) + \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \\ &+ 12 q_{\mathsf{D}}(q_{\mathsf{G}} + 4q_{\mathsf{D}}) \cdot 2^{-\gamma/2}. \end{aligned} \tag{17}$$

The adversary $\tilde{\mathcal{A}}$ *makes* $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ *queries to* $\mathsf{eCO.RO}$ *with a combined depth of* $d + q_{\mathsf{D}}$ *and a combined width of* w*, and* q_{D} *queries to* $\mathsf{eCO.E}$*. Here,* $\mathsf{eCO.RO}$ *simulates* $\mathsf{G} \times \mathsf{H}$*. The adversary* $\mathcal{B}$ *makes* q_{D} *many queries to* oDecrypt *and* $\mathsf{eCO.E}$ *and* q_{G} *queries to* $\mathsf{eCO.RO}$*, and neither* $\tilde{\mathcal{A}}$ *nor* $\mathcal{B}$ *query* $\mathsf{eCO.E}$ *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 briefly point out similarities and differences to the ROM counterpart, Theorem 1. First note that the bounds look very similar. The only difference lies in the additive error term that depends on the spreadness parameter γ. 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 Theorem 1. This larger additive loss 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, and calculate the term for two example cases in Section 8. Another important difference between Theorem 4 and Theorem 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 in Section 5.3 culminating in Corollary 9 which gives a standard-QROM IND-CCA-KEM security bound for $\mathsf{KEM}_m^{\perp}$ 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].

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

Game $\mathbf{G_1}$ is like **Game $\mathbf{G_0}$**, except for two modifications: The quantum-accessible random oracle G is replaced by G' as defined in Fig. 10 (i.e., it is simulated using an eQRO$_{\mathsf{Enc}}$), and after the adversary has finished, we compute oracle preimages for all ciphertexts on which oDecaps was queried, i.e., 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 ith decapsulation query. By property 1 in [DFMS21, Lem. 3.4]/Lemma 2, G' perfectly simulates G until the first eCO.E-query, and since the first eCO.E-query occurs only after $\mathcal{A}$ finishes, we have

$$\mathrm{Adv}_{\mathsf{KEM}_m^{\perp}}^{\mathsf{IND\text{-}CCA\text{-}KEM}}(\mathcal{A}) = \mathrm{Adv}^{\textbf{Game } \mathbf{G_0}} = \mathrm{Adv}^{\textbf{Game } \mathbf{G_1}} \ . \tag{18}$$

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 eCO.E call that produces $\hat{m}_1$ with all eCO.RO calls that happen after the adversary submits c_1, including the calls inside oDecaps, then continuing with the eCO.E-call that produces $\hat{m}_2$, etc. We will now use that eCO.RO and eCO.E almost-commute: By property 2.c and possibly 2.b) of [DFMS21, Lem. 3.4]/Lemma 2 and since $\Gamma(\mathsf{Enc}(\cdot;\cdot)) = 2^{-\gamma}|\mathcal{R}|$ for γ-spread PKE schemes, we have that

$$\left|\mathrm{Adv}^{\textbf{Game } \mathbf{G_1}} - \mathrm{Adv}^{\textbf{Game } \mathbf{G_2}}\right| \leq 8\sqrt{2}q_{\mathsf{D}}(q_{\mathsf{G}} + q_{\mathsf{D}}) \cdot 2^{-\gamma/2} \ . \tag{19}$$

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 oDecaps$'(c_i) = \mathsf{H}(\hat{m}_i)$ instead of oDecaps$(c_i) = \mathsf{Decaps}(sk, c_i)$ (using the convention $\mathsf{H}(\perp) := \perp$). We still let the game also compute oDecaps(c_i), as oDecaps makes queries to eCO.RO which can influence the behavior of eCO.E in subsequent queries. (Note that the reencryption step of oDecaps triggers a call to G', which in turn uses eCO.RO.) We define $\mathcal{B}$ exactly as in the proof of Theorem 1, except that it uses the oracles G' and oDecaps$''$ defined in Fig. 10: $\mathcal{B}$ runs $\mathcal{A}^{\mathsf{G}',\mathsf{H},\text{oDecaps}''}$, using its own FFP-CCA oracle oDecrypt to simulate oDecaps$''$ and answering H queries by simulating a fresh compressed oracle.[6] As soon as oDecaps$''$ adds a plaintext m to $\mathcal{L}_{\mathrm{FAIL}}$, $\mathcal{B}$ aborts $\mathcal{A}$ and returns m. If $\mathcal{A}$ finishes and $\mathcal{L}_{\mathrm{FAIL}}$ is still empty, $\mathcal{B}$ returns $\perp$.

[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.

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

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

Again, event DIFF encompasses three cases: For some decapsulation query c,

- Original decapsulation oracle $\textsc{oDecaps}(c)$ rejects, but the simulation $\textsc{oDecaps}'(c)$ does not, the latter meaning that $\textsc{oDecaps}'(c) = \mathsf{H}(\hat{m}_i)$ for $\hat{m} := \mathsf{eCO.E}(c)$. By construction of the oracles this implies that while $\hat{m}$ encrypts to c, c does not decrypt to $\hat{m}$ (under $\mathsf{PKE}^{\mathsf{G}}$, right after). (Otherwise, $\textsc{oDecaps}(c)$ would not reject.) Hence, this case only occurs if c's preimage $\hat{m}$ fails.
- 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 $\perp$, but the re-encryption check inside the $\textsc{oDecaps}$ call in line 18 checked out, meaning that $\mathsf{Enc}(pk, m, \mathsf{eCO.RO}(m) = c$ for $m := \mathsf{Dec}(sk, c)$. (Equivalently, the latter means that $\textsc{oDecrypt}(c) = m$.) Intuitively, this case again implies that $\mathcal{A}$ managed to compute a valid encryption without the respective oracle query on m.

In the above, any statements about 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 GUESS. Whenever DIFF occurs, $\mathcal{B}$ succeeds unless only case 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,

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

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

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

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 $\textsc{oDecaps}$ oracle, except that $\textsc{oDecaps}$ is still invoked before each call of $\textsc{oDecaps}'$, without the result ever being used. This is an artifact from **Game $\mathbf{G_2}$**. Omitting the $\textsc{oDecaps}$ 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 $\textsc{oDecaps}$ 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 $\mathbf{G_4}$**, except that the computations of $\textsc{oDecaps}(c_i)$ are omitted entirely. In game 4, all invocations of $\textsc{oDecaps}$ already happened after the execution of $\mathcal{A}$, hence this omission does not influence $\mathcal{A}$'s success probability and

$$\mathrm{Adv}^{\textbf{Game } \mathbf{G_4}} = \mathrm{Adv}^{\textbf{Game } \mathbf{G_5}} \quad .$$

$\text{oDecrypt}'(c)$	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. 11. Simulation $\text{oDecrypt}'$ of oracle oDecrypt for $\mathsf{PKE}^{\mathsf{G}}$. For the reader's convenience, we repeat the definition of G'.

Let $\tilde{\mathcal{A}}$ be an IND-CPA-KEM adversary against $\mathsf{KEM}_m^{\perp}$ in the eQROM$_{\mathsf{Enc}}$, simulating **Game G$_5$** to $\mathcal{A}$: $\tilde{\mathcal{A}}$ has access to a single extractable oracle whose oracle interface eCO.RO simulates the combination of G and H, i.e., eCO.RO simulates $\mathsf{G} \times \mathsf{H}$. (We decided to combine G and H into one oracle to simplify the subsequent analysis of the IND-CPA advantage against $\mathsf{KEM}_m^{\perp}$ that will be carried out in Section 5.3.) $\tilde{\mathcal{A}}$ runs $b' \leftarrow \mathcal{A}^{\mathsf{G}',\mathsf{H},\text{oDecaps}'}$ and returns b'. The simulation of $\mathcal{A}$'s oracles using eCO.RO is straightforward (preparing the redundant register in uniform superposition, querying the combined oracle, and uncomputing the redundant register), but for completeness, we now explain the technique in more detail: For any algorithm $\mathcal{A}$ expecting an eQRO$_{\mathsf{Enc}}$-modelled oracle G and a QRO H, one can define an algorithm $\tilde{\mathcal{A}}$ with access to a single oracle eCO whose oracle interface eCO.RO represents $\mathsf{G} \times \mathsf{H}$ and whose extraction interface is only relative to G, that perfectly simulates $\mathcal{A}$'s view. Whenever $\mathcal{A}$ issues a query to G, $\tilde{\mathcal{A}}$ prepares an additional output register H_{out} for H in a uniform superposition, queries eCO.RO, uncomputes H_{out} by applying the Hadamard transform to H_{out}, and forwards the input-output registers belonging to G to $\mathcal{A}$. The same idea with reversed oracle roles can be used to answer queries to H. The extraction oracle eCO.E represents an extraction interface for $\mathsf{Enc}(pk, \cdot; \cdot)$ with respect to G: This is possible as the oracle database for $\mathsf{G} \times \mathsf{H} : \mathcal{M} \to \mathcal{R} \times \mathcal{K}$ consists of registers D_m, of which each register D_m now consist of one register R_m to accommodate a superposition of elements in $\mathcal{R}$ (or $\perp$) and one register K_m to accommodate a superposition of elements in $\mathcal{K}$ (or $\perp$). The projectors of the measurements performed by extraction interface eCO.E can hence be defined in a way such that when eCO.E is queried on some ciphertext c, they select the message m where D_m is the first (in lexicographical order) register whose register R_m contains an r such that $\mathsf{Enc}(pk, m; r) = c$.

We now have

$$\mathrm{Adv}^{\textbf{Game G}_5} = \mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\tilde{\mathcal{A}}). \tag{20}$$

Collecting the terms from the hybrid transitions, using Lemma 3 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 Section 3, we continue by bounding the probability of event GUESS, and Lemma 3 below is the eQROM$_{\mathsf{Enc}}$ analogue of Lemma 1. Again, we will soon revisit FFP-CCA attacker $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$, and we will simulate $\mathcal{B}$'s oracle oDecrypt via an oracle $\text{oDecrypt}'$ (see Fig. 11) that differs from oDecrypt if an event equivalent to GUESS occurs. Therefore, we again generalize the definition of event GUESS accordingly.

Lemma 3. *Let* PKE *be γ-spread, and let $\mathcal{A}$ be an* eQROM$_{\mathsf{Enc}}$ *adversary that expects random oracles* G, H *as well as either a decapsulation oracle* oDecaps *for* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *or a decryption oracle* oDecrypt *for* $\mathsf{PKE}^{\mathsf{G}}$*, issuing at most q_D queries to the latter. Let $\mathcal{A}$ be run with* G' *and* oDecaps *or* $\text{oDecaps}'$ *(*oDecrypt *or* $\text{oDecrypt}'$*), but for each query c_i, both $\hat{m}_i = \text{oDecrypt}'(c_i)$ and $m_i = \text{oDecrypt}(c_i)$ are computed in that order, regardless of which of the two oracles* oDecaps *and* $\text{oDecaps}'$ *(*oDecrypt *and* $\text{oDecrypt}'$*) $\mathcal{A}$ has access to. Then* GUESS*, the event that $\hat{m}_i = \perp$ while $m_i \neq \perp$, is very unlikely. Concretely,*

$$\Pr[\mathsf{GUESS}] \leq 2q_D \cdot 2^{-\gamma}. \tag{21}$$

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

$$\sqrt{\Pr[\hat{m}_i = \bot \wedge m_i \neq \bot]} = \sqrt{\Pr[\hat{m}_i = \bot \wedge \mathsf{Enc}(m_i, \mathsf{eCO.RO}(m_i)) = c_i]} = \left\| \Pi_Y^{c,x} O_{XYF} \Sigma_F^{c,\emptyset} |m_i\rangle_X |0\rangle_Y |\psi_i\rangle_{FE} \right\| \tag{22}$$

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. (14)). 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 Π that is diagonal in the computational basis. We can thus simplify

$$\begin{aligned} &\left\| \Pi_Y^{c,x} O_{XYF} \Sigma_F^{c,\emptyset} |m_i\rangle_X |0\rangle_Y |\psi_i\rangle_{FE} \right\| = \left\| \Pi_{F_{m_i}}^{c,x} F_{F_{m_i}} \Sigma^{c,\emptyset} |m_i\rangle_X |0\rangle_Y |\psi_i\rangle_{FE} \right\| \\ \leq &\left\| F_{F_{m_i}} \Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} |m_i\rangle_X |0\rangle_Y |\psi_i\rangle_{FE} \right\| + \| [\Pi^{c,x}, F] \| \\ \leq &\left\| F_{F_{m_i}} \Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} |m_i\rangle_X |0\rangle_Y |\psi_i\rangle_{FE} \right\| + \sqrt{2} \cdot 2^{-\gamma/2} \end{aligned} \tag{23}$$

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

$$\Pi_{F_{m_i}}^{c,x} \Sigma_F^{c,\emptyset} = (\Pi^{c,x} \bar{\Pi}^{c,x})_{F_{m_i}} \otimes (\bar{\Pi}^{c,x})^{\otimes |\mathcal{M}|-1}_{F_{\mathcal{M} \setminus \{m_i\}}} = 0. \tag{24}$$

Combining Eqs. (22) to (24) and squaring the resulting inequality yields

$$\Pr[\hat{m}_i = \bot \wedge m_i \neq \bot] \leq 2 \cdot 2^{-\gamma}. \tag{25}$$

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 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 eQROM$_{\mathsf{Enc}}$-analogue of Theorem 2, i.e., we show that oDecrypt can be simulated via oracle oDecrypt′ (see Fig. 11) without the secret key, thereby being able to construct an FFP-CPA adversary from any FFP-CCA adversary (both in the eQROM$_{\mathsf{Enc}}$).

Theorem 5 (PKE$^{\mathsf{G}}$ FFP-CPA $\overset{\text{eQROM}_{\mathsf{Enc}}}{\Rightarrow}$ PKE$^{\mathsf{G}}$ FFP-CCA)**.** *Let* PKE *be γ-spread, and let $\mathcal{B}$ be an* FFP-CCA *adversary in the* eQROM$_{\mathsf{Enc}}$ *against* PKE$^{\mathsf{G}}$ *that makes at most q_{D} many decryption queries, and at most $q_{\mathsf{eCO.RO}}$ and $q_{\mathsf{eCO.E}}$ to the two interfaces of the* eQROM$_{\mathsf{Enc}}$*, respectively. Then there exist an* FFP-CPA *adversary $\tilde{\mathcal{B}}$ in the* eQROM$_{\mathsf{Enc}}$ *such that*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \leq (q_{\mathsf{D}} + 1)\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{B}}) + 12 q_{\mathsf{D}}(q_{\mathsf{G}} + 4 q_{\mathsf{D}}) 2^{-\gamma/2} \tag{26}$$

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* $\mathrm{Time}(\tilde{\mathcal{B}}) = \mathrm{Time}(\mathcal{B}) + O(q_{\mathsf{D}})$.

Proof. On a high level, the proof works as follows. Analogous to Theorem 4, we simulate oDecrypt by oDecrypt′. As we wish to remove the usage of oDecrypt entirely, however, we cannot use it to determine at which oDecrypt′ query a failure occurs. We thus resort to guessing that information.

On a technical level this proof follows the proof of Theorem 4 with deviations similar as in the proof of Theorem 2. Let oDecrypt′ be the simulation defined in Fig. 11. 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 Theorem 4. Like in the proof of Theorem 4, we have

$$\begin{aligned} &\mathrm{Adv}^{\mathbf{Game\ G_0}} \leq \mathrm{Adv}^{\mathbf{Game\ G_5}} + 12q_{\mathsf{D}}(q_{\mathsf{G}} + 2q_{\mathsf{D}})2^{-\gamma/2} + \Pr[\mathsf{DIFF}] \\ \leq &\mathrm{Adv}^{\mathbf{Game\ G_5}} + 12q_{\mathsf{D}}(q_{\mathsf{G}} + 2q_{\mathsf{D}})2^{-\gamma/2} + \Pr[\mathsf{DIFF} \land \neg\mathsf{GUESS}] + \Pr[\mathsf{GUESS}]. \end{aligned} \tag{27}$$

Assume without loss of generality that $\mathcal{B}$ makes exactly q_{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. 8 (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,

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{B}}) \geq \frac{1}{q_{\mathsf{D}} + 1}\left(\mathrm{Adv}^{\mathbf{Game\ G_5}} + \Pr[\mathsf{DIFF} \land \neg\mathsf{GUESS}]\right) \tag{28}$$

(note that all instances of $\mathrm{Adv}^{\mathbf{Game\ i}}$ are for $\mathcal{B}$ playing **Game i**.) Combining Eqs. (27) and (28) and Lemma 3 yields the desired bound. The statement about $\tilde{\mathcal{B}}$'s running time and number of queries is straightforward. □

Combining Theorems 4 and 5, we obtain the eQROM-analogue of Corollary 1.

Corollary 3 **($\mathsf{FO}^{\perp}_{m}$[PKE] IND-CPA and $\mathsf{PKE}^{\mathsf{G}}$ FFP-CPA $\overset{\mathrm{eQROM}_{\mathsf{Enc}}}{\Rightarrow}$ $\mathsf{FO}^{\perp}_{m}$[PKE] IND-CCA).** *Let* PKE *and* $\mathcal{A}$ *be as in Theorem 4. 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*

$$\begin{aligned} \mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) \leq &\mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\tilde{\mathcal{A}}) + (q_{\mathsf{D}} + 1)\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{B}) \\ &+ 24q_{\mathsf{D}}(q_{\mathsf{G}} + 4q_{\mathsf{D}})2^{-\gamma/2} \end{aligned} \tag{29}$$

Both adversaries $\tilde{\mathcal{A}}$ *and* $\mathcal{B}$ *make* $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ *queries to* eCO.RO*, with a combined depth of* $d + q_{\mathsf{D}}$ *and a combined width of* w*, and* q_{D} *queries to* 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, we remark that the additive error terms are a factor of 2 larger due to our modular proof (in terms of Theorems 4 and 5). It is straightforward to show that the bound of Corollary 3 can be proven without the factor of 2, when directly analyzing the composition of the reductions from Theorems 4 and 5.

While the additive error term that depends on the spreadness parameter γ 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. In Section 8, we bound the spreadness parameter for some schemes where this was relatively easy to do: the alternate candidates in the NIST post-quantum cryptography competition Frodo and HQC.

5.2 Semi-classical OWTH in the eQROM$_f$

To further analzye $\mathsf{IND\text{-}CPA\text{-}KEM}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, in the eQROM$_{\mathsf{Enc}}$, we want to apply an eQROM$_{\mathsf{Enc}}$ argument to show that keys encapsulated by $\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ are random-looking unless the adversary can be used to attack the underlying scheme PKE. In slightly more detail, 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 eQROM$_{\mathsf{Enc}}$. To theoretically justify this argument, this section develops 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 O_0 and O_1 that only differ on some set $\mathcal{S}$, while algorithm $\mathcal{A}$'s input is always defined relative to oracle O_0. In the case where $\mathcal{A}$'s oracle is O_1, the input uses fresh randomness from the adversary's point of view. Here we meet the first 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 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 eCO^0 to generate $\mathcal{A}$'s input and replace $\mathcal{A}$'s access to eCO^0 with access to eCO^1, an independent extractable compressed oracle.

Clearly, if GenInp does not query eCO^0, the two oracles eCO^0 and eCO^1 are perfectly indistinguishable to $\mathcal{A}$. But what if $\mathcal{A}$'s input depends on 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 be the only distinction opportunity in the eQROM$_f$ as there are now two oracle interfaces, $\mathsf{eCO.RO}$ and $\mathsf{eCO.E}$. As an example, let $\mathcal{A}$ have input $(x, t := f(x, \mathsf{eCO}^0.\mathsf{RO}(x)))$ for some oracle input value x. *Without any* $\mathsf{eCO.RO}$ *query*, $\mathcal{A}$ can tell the two cases apart by querying $\mathsf{eCO.E}$ 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 extractable superposition oracles eCO: When an $\mathsf{eCO.RO}$ query is performed, we first apply a *'semi-classical'* oracle O_S^{SC}, and then oracle unitary O_{XYD}. Intuitively, O_S^{SC} marks if an element of $\mathcal{S}$ was found in one of the query registers. (The plural is used since we consider parallel queries.) Formally, O_S^{SC} acts on the query input registers $X_1, \cdots X_w$ and an additional 'flag' register F that holds one qubit per oracle query, by first mapping

$$|x_1, \cdots x_w\rangle_{X_1 \cdots X_w} \otimes |b\rangle_F \mapsto |x_1, \cdots x_w\rangle_{X_1 \cdots X_w} \otimes |b \oplus [\![x_1 \in \mathcal{S} \vee \cdots \vee x_w \in \mathcal{S}]\!]\rangle_F \quad ,$$

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 eCO^0 and eCO^1 is to perform an extraction query where $\mathsf{eCO}^0.\mathsf{E}$ might returns 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. (30) formalizes that if $\mathcal{A}$ neither triggers FIND (and hence never sees a random oracle value on $\mathcal{S}$) nor EXT (meaning no extraction is performed an on a critical point), its behaviour in the two cases is the same: arbitrary events will be equally likely in both cases. Eq. (31) states that if $\mathcal{A}$ does not trigger FIND, any event will only become more likely in the resampled scenario than in the honest scenario if EXT happens. During the proof of one of this section's main theorems, we need to also reason about the probability of FIND in the two cases. Eq. (32) states that the likelihood of FIND only differs in the two scenarios if EXT happens. (To make this statement more intuitive, consider an adversary with input $(x, t := f(x, \mathsf{eCO}^0.\mathsf{RO}(x)))$ that first performs an extraction query on t and then queries the oracle on the result.) The proof of Lemma 4 is mostly reworking the probabilities by reasoning about the cases and eliminating the case where neither FIND nor EXT occurs. It is given in Appendix C (page 45).

Lemma 4. *Let* eCO^0 *and* 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* eCO^0. *Let* $\mathcal{S}$ *be the set of elements* $x \in \mathcal{X}$ *whose oracle values are needed to compute inp, and let* $\mathcal{T}_\mathcal{S} := \{t \mid \exists x \in \mathcal{S} \text{ s.th. } t = f(x, \mathsf{eCO}^0(x))\}$. *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*

$$\Pr[E \wedge \neg FIND \wedge \neg EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] \\ = \Pr[E \wedge \neg FIND \wedge \neg EXT : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}] \ , \tag{30}$$

$$|\Pr[E \wedge \neg FIND : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[E \wedge \neg FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| \\ \leq \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] \ , \tag{31}$$

$$|\Pr[FIND : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| \leq \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] \ , \tag{32}$$

where all probabilities are taken over the coins of 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 eCO^0 and eCO^1 to the probability that FIND or 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. Theorem 6 is the eQROM_f counterpart of [AHU19, Th. 1, 'Semi-classical O2H']. Its proof is given in Appendix D (page 47). In the special case where 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 eQROM_f.

Theorem 6 (Semi-classical OWTH in the eQROM_f: Distinguishing to Finding). *Let* eCO^0, eCO^1, GenInp, $\mathcal{S}$, *FIND and EXT be like in Lemma 4. We define the* 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 taken over the coins of GenInp *and the internal randomness of* $\mathcal{A}$. *For any algorithm* $\mathcal{A}$ *of query depth d with respect to* $\mathsf{eCO.RO}$, *we have that*

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]} \\ + 2 \cdot (\sqrt{d} + 1) \cdot \sqrt{\Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0}] + \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^1}]} \ . \tag{33}$$

In the special case where $\Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}] = \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}] = 0$*, we obtain*

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]} \ . \tag{34}$$

In many cases, a desired reduction will not know the 'resampled' set $\mathcal{S}$. We therefore proceed by giving Theorem 7 which relates the probability of FIND to the advantage of a preimage extractor algorithm ExtractSet that extracts an element of $\mathcal{S}$ without knowing $\mathcal{S}$: ExtractSet will simply run $\mathcal{A}$ with the unpunctured oracle 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 Theorem 7 for *arbitrary* sets $\mathcal{S}''$.

Theorem 7 (Semi-classical OWTH in the eQROM_f**: Finding to Extracting).** *Let* $\mathcal{A}$ *be an algorithm with access to an extractable superposition oracle* 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* eCO.RO*, and let* GenInp *be like in Lemma 4. 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, where the puncturing happens on a set* $\mathcal{S}''$*.*

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

$$\Pr[FIND : \mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}''}] \leq 4d \cdot \Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}] \ . \tag{35}$$

The proof directly follows from [AHU19, Th. 2, 'Search in semi-classical oracle'] since [AHU19, Th. 2] gives a bound with the same ride-hand side as in Theorem 7 for algorithms $\mathcal{B}$ accessing a semi-classical oracle $O^{\mathsf{SC}}_{\mathcal{S}''}$ itself (rather than some oracle punctured on $\mathcal{S}''$). An algorithm $\mathcal{B}^{O^{\mathsf{SC}}_{\mathcal{S}''}}$ hence can perfectly simulate $\mathsf{eCO}\backslash\mathcal{S}''$ to $\mathcal{A}$ by simulating eCO and having the puncturing done by its own oracle $O^{\mathsf{SC}}_{\mathcal{S}''}$.

Proof. Given an algorithm $\mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}}$, we define an algorithm $\mathcal{B}^{O^{\mathsf{SC}}_{\mathcal{S}''}}$ as follows: $\mathcal{B}^{O^{\mathsf{SC}}_{\mathcal{S}''}}$ initializes a fresh extractable superposition oracle simulation eCO. After generating $\mathcal{A}$'s input inp, $\mathcal{B}$ runs $\mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}}$ by simulating $\mathsf{eCO}\backslash\mathcal{S}$ as follows: Extraction queries are simply answered using eCO.E, and random oracle queries with query registers XY are answered by first performing a query to its own oracle $O^{\mathsf{SC}}_{\mathcal{S}''}$ with these registers and then applying eCO.RO.

Since $\mathcal{B}$ perfectly simulates $\mathsf{eCO}\backslash\mathcal{S}$ to $\mathcal{A}$ and since $\mathcal{B}$'s queries to $O^{\mathsf{SC}}_{\mathcal{S}''}$ are exactly $\mathcal{A}$'s queries to $\mathsf{eCO}\backslash\mathcal{S}$,

$$\Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}''}] = \Pr[\mathrm{FIND} : \mathcal{B}^{O^{\mathsf{SC}}_{\mathcal{S}''}}] \ . \tag{36}$$

Applying [AHU19, Th. 2] to $\mathcal{B}$ yields

$$\Pr[\mathrm{FIND} : \mathcal{B}^{O^{\mathsf{SC}}_{\mathcal{S}''}}] \leq 4d \cdot \Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}'(\mathcal{B})] \ , \tag{37}$$

where ExtractSet′ randomly measures one of $\mathcal{B}$'s queries to generate its output. Unwrapping $\mathcal{B}$ into ExtractSet′ defines the theorem's extractor ExtractSet that randomly measures one of $\mathcal{A}$'s queries to generate its output.

$$\Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}'(\mathcal{B})] = \Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}(\mathcal{A})] \ . \tag{38}$$

Collecting the probabilities yields the desired bound. □

In the case that the input inp of $\mathcal{A}$ is independent of $\mathcal{S}''$, we furthermore get the following extraction bound. Corollary 4 is the eQROM_f counterpart of [AHU19, Cor. 1].

Corollary 4 (Semi-classical OWTH in the eQROM_f: 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,*

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

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

$$\Pr[FIND : \mathcal{A}^{\mathsf{eCO}\setminus\{x\}}] \leq \frac{4q}{|X|} \ , \tag{39}$$

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

The proof is the same as in [AHU19]: W.l.o.g., we can assume that $\mathcal{A}$ does not perform parallel queries, meaning that $q = d$ and the probability of $\mathsf{ExtractSet}$ succeeding is the probability that $\mathsf{ExtractSet}$ outputs an element $x \in \mathcal{S}''$ that is independent of its input. Hence $\Pr[\mathcal{S}'' \cap \mathcal{S}' \neq \emptyset : \mathcal{S}' \leftarrow \mathsf{ExtractSet}(inp)] \leq p_{\max}$, and the corollary follows from Theorem 7.

5.3 From $\mathsf{IND\text{-}CPA_{PKE}}$ or $\mathsf{OW\text{-}CPA_{PKE}}$ to $\mathsf{IND\text{-}CPA_{FO[PKE]}}$

We will now use the OWTH results from Section 5.2 to show that the $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ can be based on the passive security of PKE. In Theorem 8, we base $\mathsf{IND\text{-}CPA}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ on the $\mathsf{IND\text{-}CPA}$ security of PKE, and for the sake of completeness, we base it on $\mathsf{OW\text{-}CPA}$ security of PKE in Theorem 9. The obtained bounds are the same as their known plain QROM counterparts.

Theorem 8 (PKE $\mathsf{IND\text{-}CPA}$ $\overset{\mathrm{eQROM}_{\mathsf{Enc}}}{\Rightarrow}$ $\mathsf{FO[PKE]}$ $\mathsf{IND\text{-}CPA\text{-}KEM}$)**.** *Let $\mathcal{A}$ be an* $\mathsf{IND\text{-}CPA}$ *adversary against $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$, 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* $\mathsf{IND\text{-}CPA}$ *adversary $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ against* PKE *such that*

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

The running time and quantum memory footprint of $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ satisfy $\mathrm{Time}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) = \mathrm{Time}(\mathcal{A}) + \mathrm{Time}(\mathsf{eCO}, q, q_E)$ 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 in the context of the overall result: In the bigger picture, $\mathsf{eCO.E}$ queries are only triggered by an $\mathsf{IND\text{-}CCA}$ adversary querying its simulated oracle oDECAPS′, and oDECAPS′ rejects queries on c^* right away.

To summarise the proof, we first define a Game G_1 like the $\mathsf{IND\text{-}CPA\text{-}KEM}$ game for $\mathsf{KEM}_m^{\perp}$, 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 G_1, the forwarded KEM key is a uniformly random key either way, the advantage of $\mathcal{A}$ in Game G_1 hence is 0. It remains to bound the distinguishing advantage between the $\mathsf{IND\text{-}CPA\text{-}KEM}$ game and Game G_1. We apply the 'Distinguishing to Finding' Theorem 6 which bounds this distinguishing advantage in terms of the probability of event FIND_{m^*}, the event that m^* is detected in the adversary's random oracle queries. To further bound $\Pr[\mathrm{FIND}_{m^*}]$, we use $\mathsf{IND\text{-}CPA}$ 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, FIND_{m^*} is highly unlikely for large enough message spaces. (This uses the 'independent values' Corollary 4 .)

Games $\mathbf{G_0} - \mathbf{G_1}$	
01 $(pk, sk) \leftarrow \mathsf{KG}$	
02 $b \leftarrow_\$ \{0,1\}$	
03 $m^* \leftarrow_\$ \mathcal{M}$	
04 $(r^*, K_0^*) := \mathsf{eCO.RO}(m^*)$	$/\!/ G_0$
05 $(r^*, K_0^*) \leftarrow_\$ \mathcal{R} \times \mathcal{K}$	$/\!/ G_1$
06 $c^* := \mathsf{Enc}(pk, m^*; r^*)$	
07 $K_1^* \leftarrow_\$ \mathcal{K}$	
08 $b' \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk, c^*, K_b^*)$	
09 **return** $[\![b' = b]\!]$	

Fig. 12. Games $G_0 - G_1$ for the proof of Theorem 8.

Proof. Let $\mathcal{A}$ be an adversary against the IND-CPA security of $\mathsf{KEM}^{\perp} = \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$, issuing random oracle queries to both its oracles of query depth d, and q many in total. Consider the two games given in Fig. 12.

Game $\mathbf{G_0}$ essentially is game $\mathsf{IND\text{-}CPA\text{-}KEM}_{\mathsf{KEM}^{\perp}}$, the only difference is that we combined oracles G and H into a single oracle eCO. As discussed in the proof of Theorem 4 (before Eq. (20), see page 21), this change is merely of a conceptual nature, simplifying our later reasoning about the synchronous reprogramming of G and H on m^*.

$$\mathrm{Adv}_{\mathsf{KEM}^{\perp}}^{\mathsf{IND\text{-}CPA\text{-}KEM}}(\mathcal{A}) = |\mathrm{Adv}^{\textbf{Game } \mathbf{G_0}} - \frac{1}{2}| \ .$$

In **Game $\mathbf{G_1}$**, we replace oracle values $r^* := \mathsf{G}(m^*)$ and $K_0 := \mathsf{H}(m^*)$ with fresh random values (see line 05). Since K_b^* is now an independent random value regardless of the challenge bit,

$$\mathrm{Adv}^{\textbf{Game } \mathbf{G_1}} = \frac{1}{2} \ .$$

We will now apply Theorem 6 to relate $\mathcal{A}$ being able to distinguish between Game G_0 and Game G_1 to the probability that $\mathcal{A}$'s queries contain m^*, or more precisely, the probability that $\mathcal{A}$ would trigger event FIND_{m^*} in Game G_1, would it be run with the punctured oracle $\mathsf{eCO}\backslash\{m^*\}$ that additionally measures whether any of $\mathcal{A}$'s random oracle queries contained m^* and in that case sets flag FIND_{m^*} to 1. We claim that

$$|\mathrm{Adv}^{\textbf{Game } \mathbf{G_0}} - \mathrm{Adv}^{\textbf{Game } \mathbf{G_1}}| \leq 4 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND}_{m^*} \text{ in } \textbf{Game } \mathbf{G_1}^{\mathsf{eCO}\backslash\{m^*\}}]} \ . \tag{40}$$

To verify this claim, we identify each Game G_b (where $b \in \{0,1\}$) with one of the OWTH games defined in Theorem 6 as follows: As GenInp, we define the algorithm that samples a key pair and a random message m^*, queries eCO^0 on m^* to obtain r^* and K_0^*, and outputs as inp the public key as well as $c^* := \mathsf{Enc}(pk, m^*; r^*)$ and K_0^*. With this identification, set $\mathcal{S}$ from Theorem 6 is $\{m^*\}$.

As the OWTH distinguisher, we define algorithm $\mathcal{D}$ that gets inp, picks a random bit b and a random key K_1^* and then forwards pk, c^* and K_b^* to $\mathcal{A}$. It forwards all of $\mathcal{A}$'s random oracle and extraction queries to its own respective oracle, and at the end, it returns 1 iff $\mathcal{A}$'s output bit is equal to b. When $\mathcal{D}$ is run with access to eCO^0, it perfectly simulates Game G_0, and when $\mathcal{D}$ is run with access to eCO^1, the input is defined relative to oracle eCO^0, while the oracle to which $\mathcal{A}$'s queries are forwarded by $\mathcal{D}$ is eCO^1. Since everything except for the values r^* and K_0^* computed by GenInp is now independent of the oracle eCO^0 which is furthermore inaccessible to $\mathcal{D}$ and $\mathcal{A}$, this is equivalent to simply sampling random values r^* and K_0^* instead, therefore

$$|\mathrm{Adv}^{\textbf{Game } \mathbf{G_0}} - \mathrm{Adv}^{\textbf{Game } \mathbf{G_1}}| = \mathrm{Adv}_{\mathrm{eQRO}_f}^{\mathsf{OWTH}}(\mathcal{D}) \ .$$

Note that EXT from Theorem 6 corresponds to the event that $\mathcal{A}$ queries its extraction oracle $\mathsf{eCO.E}$ on c^*, which we ruled out in the theorem statement as a prerequisite. Therefore, we can apply the special case bound Eq. (34) of Theorem 6, and since $\mathcal{D}$ has exactly the query behaviour of $\mathcal{A}$ and triggers FIND exactly if $\mathcal{A}$ triggers FIND,

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{D}) \leq 4 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND}_{m^*} \text{ in } \mathbf{Game\ G_1}^{\mathsf{eCO}\backslash\{m^*\}}]} \ .$$

What we have shown so far is that

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA\text{-}KEM}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND}_{m^*} \text{ in } \mathbf{Game\ G_1}^{\mathsf{eCO}\backslash\{m^*\}}]} \ . \tag{41}$$

In order to take the last step towards our reduction, consider the two games given in Fig. 13.

Games $\mathbf{G_2} - \mathbf{G_3}$		**Reduction $B^1_{\mathsf{IND\text{-}CPA}}(pk)$**
$(pk, sk) \leftarrow \mathsf{KG}$		$m^*, \tilde{m} \leftarrow_\$ \mathcal{M}$
$m^* \leftarrow_\$ \mathcal{M}$		**return** $(m^*, \tilde{m}, \mathrm{st} := m^*)$
$c^* \leftarrow \mathsf{Enc}(pk, m^*)$	$/\!\!/ G_2$	
$\tilde{m} \leftarrow_\$ \mathcal{M}$	$/\!\!/ G_3$	
$c^* \leftarrow \mathsf{Enc}(pk, \tilde{m})$	$/\!\!/ G_3$	**Reduction $B^2_{\mathsf{IND\text{-}CPA}}(pk, c^*, \mathrm{st} = m^*)$**
$K^* \leftarrow_\$ \mathcal{K}$		$K^* \leftarrow_\$ \mathcal{K}$
$b' \leftarrow \mathcal{A}^{\mathsf{eCO}\backslash\{m^*\}}(pk, c^*, K^*)$		$b' \leftarrow \mathcal{A}^{\mathsf{eCO}\backslash\{m^*\}}(pk, c^*, K^*)$
if FIND_{m^*} **return** 1		**if** FIND_{m^*} **return** 1

Fig. 13. Games $G_2 - G_3$ and $\mathsf{IND\text{-}CPA}$ reduction $B_{\mathsf{IND\text{-}CPA}} = (B^1_{\mathsf{IND\text{-}CPA}}, B^2_{\mathsf{IND\text{-}CPA}})$ for the proof of Theorem 8.

Game $\mathbf{G_2}$ exactly formalises $\Pr[\mathrm{FIND}_{m^*}$ in $\mathbf{Game\ G_1}^{\mathsf{eCO}\backslash\{m^*\}}]$. We cleaned up some variables that are not needed any longer - since r^* is uniformly random in Game G_1 and since it will be used nowhere but in line 06 (of Game G_1), we can drop it altogether and simply write $c^* \leftarrow \mathsf{Enc}(pk, m^*)$ instead. Similarly, since K_0^* is uniformly random in Game G_1 (as is K_1^*), we do not need to distinguish between K_0^* and K_1^* any longer, thereby also rendering bit b redundant.

$$\Pr[\mathrm{FIND}_{m^*} \text{ in } \mathbf{Game\ G_1}^{\mathsf{eCO}\backslash\{m^*\}}] = \mathrm{Adv}^{\mathbf{Game\ G_2}} \ . \tag{42}$$

In **Game $\mathbf{G_3}$**, we replace c^* with an encryption of another random message, while sticking with puncturing the oracle on m^*. With this change, m^* becomes independent of $\mathcal{A}$'s input, and using Eq. (39) from Corollary 4 yields

$$\mathrm{Adv}^{\mathbf{Game\ G_3}} \leq \frac{4q}{|\mathcal{M}|} \ . \tag{43}$$

To upper bound $|\mathrm{Adv}^{\mathbf{Game\ G_2}} - \mathrm{Adv}^{\mathbf{Game\ G_3}}|$, consider the reduction given in Fig. 13. Since $B_{\mathsf{IND\text{-}CPA}}$ perfectly simulates either Game G_2 or Game G_3, depending on which message is encrypted in its $\mathsf{IND\text{-}CPA}$ challenge,

$$|\mathrm{Adv}^{\mathbf{Game\ G_2}} - \mathrm{Adv}^{\mathbf{Game\ G_3}}| = \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) \ . \tag{44}$$

Combining equations (42), (43) and (44) yields

$$\Pr[\mathrm{FIND}_{m^*} \text{ in } \mathbf{Game\ G_1}^{\mathsf{eCO}\backslash\{m^*\}}] \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{4q}{|\mathcal{M}|} \ . \tag{45}$$

Plugging Eq. (45) into Eq. (40) and using that $d \leq q$ yields the bound claimed in Theorem 8. The statement about $B_{\mathsf{IND\text{-}CPA}}$'s runtime is straightforward.

□

Theorem 9 (PKE OW-CPA $\overset{\text{eQROM}_{\text{Enc}}}{\Rightarrow}$ FO[PKE] IND-CPA). *For any* IND-CPA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *in the* $\text{eQROM}_{\mathsf{Enc}}$ *that issues* q *many queries to* eCO.RO *in total, with a query depth (width) of* d *(*w*), and* q_E *many queries to* eCO.E*, where none of them is with its challenge ciphertext. there furthermore exists an* OW-CPA *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *such that*

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

The running time and quantum memory footprint of $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *satisfy* $\text{Time}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) = \text{Time}(\mathcal{A}) + \text{Time}(\mathsf{eCO}, q, q_E)$ *and* $\text{QMem}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) = \text{Time}(\mathcal{A}) + \text{QMem}(\mathsf{eCO}, q, q_E)$*.*

In a nutshell, the proof proceeds by going through exactly the same steps as the one of Theorem 8, up to the point where we bound $\Pr[\text{FIND}_{m^*}]$. To bound $\Pr[\text{FIND}_{m^*}]$, we use the 'Finding to Extracting' Theorem 7 to relate $\Pr[\text{FIND}_{m^*}]$ to the OW-CPA advantage of an algorithm that extracts m^* from $\mathcal{A}$'s oracle queries.

Proof. Let $\mathcal{A}$ again be an adversary against the IND-CPA security of $\mathsf{KEM}^{\perp}$, issuing random oracle queries of query depth d, and q many in total. Defining Game G_0 to Game G_2 exactly like in the proof of Theorem 8 and combining Eq. (41) and Eq. (42), we obtain

$$\text{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \text{Adv}^{\textbf{Game } \mathbf{G_2}}} \quad . \tag{46}$$

To bound $\text{Adv}^{\textbf{Game } \mathbf{G_2}}$, we use Theorem 7 to relate $\text{Adv}^{\textbf{Game } \mathbf{G_2}}$ to the OW-CPA advantage of an algorithm that extracts m^* from the oracle queries: In order to relate $\text{Adv}^{\textbf{Game } \mathbf{G_2}}$ to OW-CPA security using Theorem 7, consider reduction $B_{\mathsf{OW\text{-}CPA}}$ given in Fig. 14. $B_{\mathsf{OW\text{-}CPA}}$ is exactly the query extractor ExtractSet from Theorem 7 *until* $B_{\mathsf{OW\text{-}CPA}}$*'s last additional step*, where $B_{\mathsf{OW\text{-}CPA}}$ randomly chooses its output from the candidate list it extracted (in line 11). Since Game G_2 exactly models the probability that $\mathcal{A}$ triggers FIND_{m^*}, applying Theorem 7 yields

$$\text{Adv}^{\textbf{Game } \mathbf{G_2}} \leq 4d \cdot \Pr[m^* \in \mathcal{S}' : \mathcal{S}' \leftarrow \mathsf{ExtractSet}(pk, c^*)] \ , \tag{47}$$

where ExtractSet is the query extractor from Theorem 7, meaning $\mathcal{S}'$ is the result of running $\mathcal{A}^{\mathsf{eCO}}(inp)$ until (just before) the i-th query, measuring all query input registers, and returning as $\mathcal{S}'$ the set of measurement outcomes. Since $B_{\mathsf{OW\text{-}CPA}}$ does exactly the same and then picks a random element of $\mathcal{S}'$, and since $B_{\mathsf{OW\text{-}CPA}}$ wins if it randomly picked m^* from $\mathcal{S}'$,

$$\Pr[m^* \in \mathcal{S}' : \mathcal{S}' \leftarrow \mathsf{ExtractSet}(pk, c^*)] \leq |\mathcal{S}'| \cdot \text{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) \ . \tag{48}$$

Combining equations (47) and (48) yields

$$\text{Adv}^{\textbf{Game } \mathbf{G_2}} \leq 4d \cdot w \cdot \text{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) \ , \tag{49}$$

where we used that $|\mathcal{S}'|$, the number of parallel queries issued during $\mathcal{A}$'s i-th query, can be upper bounded by w, the maximal query width.

Plugging Eq. (49) into Eq. (46) yields the bound claimed in Theorem 9. Again, the statement about $B_{\mathsf{OW\text{-}CPA}}$'s runtime is straightforward.

□

Game G_2	**Reduction $B_{\mathsf{OW\text{-}CPA}}(pk, c^*)$**
$(pk, sk) \leftarrow \mathsf{KG}$	$i \leftarrow_\$ \{1, \cdots, d\}$
$m^* \leftarrow_\$ \mathcal{M}$	$K^* \leftarrow_\$ \mathcal{K}$
$c^* \leftarrow \mathsf{Enc}(pk, m^*)$	Run $\mathcal{A}^{\mathsf{eCO}}(pk, c^*, K^*)$
$K^* \leftarrow_\$ \mathcal{K}$	until its i-th query to $\mathsf{eCO.RO}$
$b' \leftarrow \mathcal{A}^{\mathsf{eCO}\setminus\{m^*\}}(pk, c^*, K^*)$	$\{m'_1, m'_2, \cdots\} \leftarrow$ MEASURE query input registers
if FIND_{m^*} **return** 1	$m' \leftarrow_\$ \{m'_1, m'_2, \cdots\}$
	return m'

Fig. 14. Game G_2 and OW-CPA reduction $B_{\mathsf{OW\text{-}CPA}}$ for the proof of Theorem 9.

6 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 PKE^G can already be bounded for particular instantiations of PKE^G without too much technical overhead, even in the $\mathrm{eQROM}_{\mathsf{Enc}}$, this section offers an alternative way to bound this probability: In Theorem 10, we relate the success probability in game FFP-CPA for PKE^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 PKE^G *without knowledge of the key.*

We begin by defining these two new notions related to decryption failures: In Fig. 15 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. 15. 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

$$\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{A}) := \left| \Pr[\mathsf{FFP\text{-}NG}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1] - \frac{1}{2} \right| \ .$$

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), the latter being 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 $\mathsf{FFP\text{-}CPA}_{\mathsf{PKE}^\mathsf{G}}$ or correctness-related games from the existing literature.

Theorem 10 (PKE FFP-NG and PKE^G FFP-NK $\Rightarrow$ PKE^G FFP-CPA). *Let* PKE *be a public-key encryption scheme. For any* FFP-CPA *adversary $\mathcal{A}$ in the* $\mathrm{eQROM}_{\mathsf{Enc}}$ *against* PKE^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* $\mathrm{eQROM}_{\mathsf{Enc}}$ *against* PKE^G *and an* FFP-NG *adversary $\mathcal{B}$ against* PKE *with*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^\mathsf{G}}(\mathcal{A}) \le 2\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{B}) + \mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^\mathsf{G}}(\mathcal{C}) \ .$$

Game FFP-NK	Game FFP-NG	$\mathsf{FCO}_b(m;r)$ //one query
$m \leftarrow \mathcal{A}$	$(sk_0, pk_0) \leftarrow \mathsf{KG}$	$c \leftarrow \mathsf{Enc}(pk_b, m; r)$
$(pk, sk) \leftarrow \mathsf{KG}$	$(sk_1, pk_1) \leftarrow \mathsf{KG}$	$m' := \mathsf{Dec}(sk_b, c)$
$c := \mathsf{Enc}(pk, m)$	$b \leftarrow \{0,1\}$	**return** $[\![m \neq m']\!]$
$m' := \mathsf{Dec}(sk, c)$	$b' \leftarrow \mathcal{A}^{\mathsf{FCO}_b}(pk_0)$	
return $[\![m' \neq m]\!]$	**return** $[\![b = b']\!]$	

Fig. 15. Key-independent game FFP-NK for deterministic schemes PKE, and the find non-generically failing ciphertexts game FFP-NG, for PKE. $\mathcal{A}$ can make at most one query to FCO_{sk_b}.

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

Proof. By definition of the FFP-CPA advantage, we have

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{A}) = \Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk, pk)] \ .$$

To upper bound this probability, we begin by defining FFP-NG adversary $\mathcal{B}$: On input pk, $\mathcal{B}$ runs $\mathcal{A}(pk)$, simulating eCO to $\mathcal{A}$. When $\mathcal{A}$ finishes by outputting its message m, $\mathcal{B}$ computes $r := \mathsf{eCO.RO}(m)$, uses its failure-checking oracle to compute $b' := \mathsf{FCO}_b(m, r)$ and outputs b'. In the case where the challenge bit b of $\mathcal{B}$'s FFP-NG game is 0, $\mathcal{B}$ perfectly simulates the FFP-CPA game to $\mathcal{A}$ and wins iff $\mathcal{A}$ wins in game FFP-CPA. Therefore,

$$\begin{aligned}\Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk, pk)] &= \Pr[1 \leftarrow \mathcal{B}(pk) | b = 0] \\ &\leq \Pr[1 \leftarrow \mathcal{B}(pk) | b = 1] + 2\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{B}) \ ,\end{aligned}$$

where the last line used the definition of the FFP-NG advantage.

To upper bound $\Pr[1 \leftarrow \mathcal{B}(pk) | b = 1]$, note that this probability formalizes $\mathcal{A}$ outputting a message that fails to decrypt, but under an independently drawn key pair (sk', pk'):

$$\Pr[1 \leftarrow \mathcal{B}(pk) | b = 1] = \Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk', pk')] \ , \tag{50}$$

where the probability is taken additionally over $(sk', pk') \leftarrow \mathsf{KG}$.

To upper bound this probability, we define FFP-NK adversary $\mathcal{C}^{\mathsf{eCO}}$ against $\mathsf{PKE}^{\mathsf{G}}$: Upon initialisation, $\mathcal{C}$ computes a key pair (pk, sk) on its own and runs $\mathcal{A}^{\mathsf{eCO}}(pk)$. When $\mathcal{A}$ finishes by outputting its message m, $\mathcal{C}$ forwards the message to its own game. Since $\mathcal{C}$ perfectly simulates the game in Eq. (84) to $\mathcal{A}$ and wins iff $\mathcal{A}$ wins,

$$\Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk', pk')] = \mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) \ .$$

□

6.1 Characterizing $\mathsf{FFP\text{-}NK}_{\mathsf{PKE}^{\mathsf{G}}}$

In the last section, we have related the success probability of an adversary in game FFP-CPA for $\mathsf{PKE}^{\mathsf{G}}$ to the success property of an adversary in game FFP-NK for $\mathsf{PKE}^{\mathsf{G}}$, in the $\mathrm{eQROM}_{\mathsf{Enc}}$. 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 a certain predicate, i.e., the predicate that (m, r) fails with respect to pk. To prove the upper bound we provide in Theorem 11, we therefore generically bound the success probability for a certain search problem in Section 6.2. 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 δ_{ik} and $\sigma_{\delta_{\text{ik}}}$: Below, we define the worst-case decryption error rate δ_{ik} *under independent keys*, and the maximal variance of the decryption error rate $\sigma_{\delta_{\text{ik}}}$.

Definition 5 (worst-case independent-key decryption error rate, maximal decryption error variance). *We define the* worst-case decryption error rate under independent keys δ_{ik} *and the* maximal decryption error variance under independent keys $\sigma_{\delta_{\text{ik}}}$ *of a public-key encryption scheme* PKE *as*

$$\delta_{\text{ik}} := \max_{m\in\mathcal{M}}[\Pr_{(sk,pk),r}[(m,r)\ \textit{fails}]] = \max_{m\in\mathcal{M}} \mathbb{E}_r[\Pr_{(sk,pk)}[(m,r)\ \textit{fails}]]\ \ ,\ \textit{and}$$
$$\sigma^2_{\delta_{\text{ik}}} := \max_{m\in\mathcal{M}} \mathbb{V}_r[\Pr_{(sk,pk)}[(m,r)\ \textit{fails}]]$$

for uniformly random r*.*

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

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

Intuitively, δ_{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 δ_{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 δ_{wc} serves as an upper bound for δ_{ik}.

Theorem 11 (Upper bound for FFP-NK of $\mathsf{PKE}^{\mathsf{G}}$). *Let* PKE *be a public-key encryption scheme with worst-case independent-key decryption error rate* δ_{ik} *and decryption error rate variance* $\sigma_{\delta_{\text{ik}}}$*. For any* FFP-NK *adversary* $\mathcal{A}$ *in the* $\text{eQROM}_{\mathsf{Enc}}$ *against* $\mathsf{PKE}^{\mathsf{G}}$*, setting* $C = 304$*, we have that*

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

In Section 6.3 , we give an alternative bound that grows with the logarithm of the number of RO queries, assuming a *Gaussian -shaped tail bound* for the decryption error probabilitydistribution.

Proof. The claimed bounds result from applying Corollaries 5 and 7 that we give in Section 6.2 below: The success probability of $\mathcal{A}$ in game FFP-NK is the probability that $\mathcal{A}$'s output message fails to decrypt. If we define function F by setting $F(m,r) := \Pr[(m;r)$ fails wrt. $(sk,pk)]$, we can alternatively describe $\mathcal{A}$'s task as the task to find a superposition oracle input m such that $F(m, \mathsf{eCO.RO}(m))$ is large, having access to the extractable oracle simulation eCO. We prove general upper bounds for the success probability in finding large values for arbitrary functions F in Section 6.2, which immediately yields the claimed bound. □

6.2 Finding large values of a function in the eQROM_f

In this section, we provide the technical results for the eQROM_f that we need to prove Theorem 11. Throughout this section, f is a fixed function such that eQROM_f is well-defined. We begin by providing a bound for the success probability of an algorithm in the 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 that provides this upper bound, we will use the quantity Γ_R that was defined in Eq. (16) (see page 16).

Lemma 5 (Slight generalization of [DFMS21, Proposition 3.5]). *Let $R \subset \mathcal{X}\times\mathcal{Y}$ be a relation and $\mathcal{A}^{\mathsf{eCO}}$ an algorithm with access to eQRO_f from $\mathcal{X}$ to $\mathcal{Y}$ for some function $f : \mathcal{X}\times\mathcal{Y} \to \mathcal{T}$, making q queries to* eCO.RO*. Then*

$$\Pr_{x\leftarrow\mathcal{A}^{\mathsf{eCO}}}[R(x,\mathsf{eCO.RO}(x))] \le 152(q+1)^2\frac{\Gamma_R}{|\mathcal{Y}|}, \tag{51}$$

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

The generalization consists of allowing $\mathcal{A}$ to query eCO.E as well.

Proof. The only difference between [DFMS21, Proposition 3.5] and Lemma 5 is that $\mathcal{A}$ now additionally has access to eCO.E. The proof is thus the same as for [DFMS21, Proposition 3.5], with the additional observation that queries to eCO.E commute with the progress measure operator M for any relation R. This is because i) both M and the operator applied upon an eCO.E query are controlled unitaries controlling on the database register of the compressed oracle database of the eQRO_f, and ii) the target registers of M and eCO.E are disjoint. □

According to Lemma 5, it is hard to search a random oracle, even given extraction access. We will now use Lemma 5 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 12. *Let F and I be as above. Let further $G : [0,1] \to [0,1]$ be non-increasing such that $G(t) \ge \Pr_{r\leftarrow Y}[F(x,r) \ge t]$ 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 $\kappa_q := \min\{i | Cq^2 G(t_i) \le 1\}$. Then for any algorithm $\mathcal{A}^{\mathsf{eCO}}$ making at most $q \ge 1$ queries to* eCO.RO*,*

$$\mathbb{E}_{x\leftarrow\mathcal{A}^{\mathsf{eco}}}[F(x,\mathsf{eCO.RO}(x))] \le t_{\kappa_q} + Cq^2 \sum_{i=\kappa_q+1}^{R} t_i \Delta G(i) \ . \tag{52}$$

eCO.RO *is queried once in the end to determine* eCO.RO(x)*.*

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

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

where we have used Lemma 5 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

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

finishing the proof. □

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

Corollary 5. *Let F, I, and C be as in Theorem 12, 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* $\mathsf{eCO.RO}$,

$$\mathbb{E}_{x \leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x))] \leq \mu + 3\sqrt{C}q\sigma + 2Cq^2\sigma^2\mu \log \frac{1}{\sqrt{C}q\sigma}. \tag{53}$$

Proof. By Chebyshev's inequality, we can set

$$G(t) = \frac{\sigma^2}{(t-\mu)^2}. \tag{54}$$

We thus obtain $t_{\kappa_q} \leq \sqrt{C}q\sigma + \mu$. We bound

$$\sum_{i=\kappa_q+1}^{R} t_i \Delta G(i) = -\sum_{i=\kappa_q+1}^{R} t_i \int_{t_i}^{t_{i+1}} G'(t)\mathrm{d}t \tag{55}$$

$$\leq -\int_{t_{\kappa_q}}^{1} tG'(t)\mathrm{d}t \tag{56}$$

$$=2\sigma^2 \int_{t_{\kappa_q}}^{1} \frac{t}{t-\mu}\mathrm{d}t \tag{57}$$

$$=2\sigma^2 \int_{t_{\kappa_q}-\mu}^{1-\mu} \frac{u+\mu}{u}\mathrm{d}u \tag{58}$$

$$=2\sigma^2 \left(1 - t_{\kappa_q} + \mu \log \frac{1-\mu}{t_{\kappa_q}-\mu}\right). \tag{59}$$

We arrive at the bound

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

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

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

6.3 Alternative bound for FFP-NK based on a stronger tail bound

In this subsection, we show how to use a stronger uniform tail bound in place of Chebyshev's inequality to obtain a stronger bound for the adversarial advantage in FFP-NK.

We begin by defining the decryption error tail envelope.

Definition 6 (decryption error tail envelope). *We define the* decryption error tail envelope *as*

$$\tau(t) := \max_m \Pr_{r\leftarrow\mathcal{R}}\left[\Pr_{(sk,pk)}[(m,r)\ \textit{fails}] \geq t\right] .$$

We obtain the following stronger bound for FFP-NK that scales logarithmically with the adversary's random oracle queries.

Theorem 13 (Upper bound for FFP-NK of $\mathsf{PKE}^{\mathsf{G}}$). *Let* PKE *be a public-key encryption scheme with worst-case random-key decryption error rate δ_{ik} and decryption error tail envelope τ. For any* FFP-NK *adversary $\mathcal{A}$ in the* eQROM$_{\mathsf{Enc}}$ *against* $\mathsf{PKE}^{\mathsf{G}}$*, setting $C = 304$, we have that*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{A}) \leq \delta_{\mathrm{ik}} + 2\beta^{-1/2}\sqrt{\ln(2C\sqrt{\beta}) + 2\ln(q)}.$$

The above theorem follows directly by an application of Corollary 7 given below. Combining Theorem 13 with the reductions from Sections 5.1 and 5.3 we get the following alternative to Corollary 9.

Corollary 6 (PKE FFP-NG and pass. secure ⇒ $\mathsf{FO}^{\perp}_{m}$[PKE] IND-CCA). *Let* PKE *be a (randomized)* PKE *scheme that is γ-spread and with worst-case random-key decryption error rate δ_{ik}, decryption error rate variance $\sigma_{\delta_{\mathrm{ik}}}$ and decryption error tail envelope τ. Let $\mathcal{A}$ be an* IND-CCA-KEM *adversary (in the QROM) against $\mathsf{KEM}^{\perp}_{m} := \mathsf{FO}^{\perp}_{m}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, issuing at most q_{G} many queries to its oracle* G*, q_{H} many queries to its oracle* H*, and at most q_{D} many queries to its decapsulation oracle* ODECAPS*. Let $q = q_{\mathsf{G}} + q_{\mathsf{H}}$, and let d and w be the query depth and query width of the combined queries to* G *and* H*. Set $C = 304$ and assume $\sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}} \leq 1/2$. Then there exist an* IND-CPA *adversary $\mathcal{B}_{\mathsf{IND}}$, a* OW-CPA *adversary $\mathcal{B}_{\mathsf{OW}}$ and an* FFP-NG *adversary $\mathcal{C}$ against* PKE*, such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}^{\perp}_{m}}(\mathcal{A}) \leq \widetilde{\mathrm{Adv}}_{\mathsf{PKE}} + (q_{\mathsf{D}}+1)\left(2\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{C}) + \varepsilon_{\delta_{\mathrm{ik}}}\right) + \varepsilon_{\gamma} \tag{60}$$

with

$$\widetilde{\mathrm{Adv}}_{\mathsf{PKE}} = \begin{cases} 4\cdot\sqrt{(d+q_{\mathsf{D}})\cdot\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND}})} + \frac{8(q+q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} & \textit{or} \\ 8\,(d+q_{\mathsf{D}})\cdot\sqrt{w\cdot\mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW}})}. \end{cases} \tag{61}$$

The additive error term $\varepsilon_{\delta_{\mathrm{ik}}}$ is given by

$$\varepsilon_{\delta_{\mathrm{ik}}} \leq \delta_{\mathrm{ik}} + (3+2\delta_{rk})\sqrt{C}q_{\mathsf{G}}\sigma_{\delta_{\mathrm{ik}}}\ , \tag{62}$$

and the additive error term ε_{γ} is given by

$$\varepsilon_{\gamma} = 24q_{\mathsf{D}}(q_{\mathsf{G}}+2q_{\mathsf{D}})2^{-\gamma/2} + 4q_{\mathsf{D}}\cdot 2^{-\gamma}.$$

Here, $\delta_{\mathrm{ik}}, \sigma_{\delta_{\mathrm{ik}}}$ and γ are the worst-case random-key decryption error rate, the maximal decryption failure variance under random keys, and the ciphertext spreadness parameter, respectively. If the Gaussian tail bound

$$\max_m \Pr_{r\leftarrow\mathcal{R}}\left[\Pr_{(sk,pk)}[\mathsf{Dec}(sk,\mathsf{Enc}(pk,m;r)) \neq m] \geq t\right] \leq \exp\left(-\beta(t-\delta_{\mathrm{ik}})^2\right)$$

holds for some parameter β, the dependency of $\epsilon_{\delta_{\mathrm{ik}}}$ on q_{G} can be improved to

$$\varepsilon_{\delta_{\mathrm{ik}}} \leq \delta_{\mathrm{ik}} + 2\eta_1\sqrt{\ln\left(\eta_2 q_{\mathsf{G}}^2\right)} \tag{63}$$

with $\eta_1 = \beta^{-1/2}$ and $\eta_2 = 2C\sqrt{\beta}$. The running time of the adversaries $\mathcal{B}_{\mathsf{IND}}$, $\mathcal{B}_{\mathsf{OW}}$ and $\mathcal{C}$ are all bounded by

$$\mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}) + O(q_{\mathsf{D}}).$$

We continue to prove the corollary of Theorem 12 which yields Theorem 13

Corollary 7. *Let F, I, and C be as in Theorem 12. Let furthermore $\mathbb{E}[F(x, H(x))] \leq \mu$ for some $\mu \in [0,1]$ and suppose in addition that we can set $G(t) = c\exp(-\beta(t-\mu)^2)$ with $\beta \geq e/(2C)$. Then, for an algorithm $\mathcal{A}^{\mathsf{eCO}}$ making at most $q \geq 1$ quantum queries to* $\mathsf{eCO.RO}$

$$\mathbb{E}_{x\leftarrow A^{\mathsf{eco}}}[F(x, \mathsf{eCO.RO})] \leq \mu + 2\beta^{-1/2}\sqrt{\ln(2C\sqrt{\beta}) + 2\ln(q)} \tag{64}$$

Proof. Here, we directly use Lemma 5 for simplicity (a slightly tighter but less pretty bound can be obtained from Theorem 12). For any $a \in [0,1]$, we have

$$\mathbb{E}_{x\leftarrow A^{\mathsf{eco}}}[F(x, \mathsf{eCO.RO}(x))] \leq a + \Pr_{x\leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x)) \geq a]. \tag{65}$$

Setting $a = \mu + \hat{a}$ and using the definition of G as well as Lemma 5 (in the same way as in the proof of Theorem 12), we obtain

$$\Pr_{x\leftarrow A^{\mathsf{eCO}}}[F(x, \mathsf{eCO.RO}(x)) \geq \mu + \hat{a}] \leq Cq^2\exp(-\beta\hat{a}^2) \tag{66}$$

Setting $\hat{a} = \sqrt{\ln(2Cq^2\sqrt{\beta})/\beta}$ and using $\ln(2Cq^2\sqrt{\beta}) \geq 1$, we obtain

$$\mathbb{E}_{x\leftarrow A^{\mathsf{eco}}}[F(x, \mathsf{eCO.RO}(x))] \leq \mu + \beta^{-1/2}\left(1 + \sqrt{\ln(2Cq^2\sqrt{\beta})}\right) \tag{67}$$

$$\leq \mu + 2\beta^{-1/2}\sqrt{\ln(2C\sqrt{\beta}) + 2\ln(q)}, \tag{68}$$

where ln is the natural logarithm. □

7 Tying everything together

Combining the reductions from Sections 5.1 and 5.3, we obtain a first corollary that still relies on FFP-CPA of $\mathsf{PKE}^{\mathsf{G}}$.

Corollary 8 ($\mathsf{PKE}^{\mathsf{G}}$ **FFP-CPA and PKE pass. secure** $\Rightarrow$ $\mathsf{FO}_m^{\perp}$**[PKE] IND-CCA).** *Let* PKE *be a (randomized)* PKE *scheme that is γ-spread, and let $\mathcal{A}$ be an* IND-CCA-KEM *adversary (in the QROM) against* $\mathsf{KEM}^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$, *issuing at most q_{G} many queries to its oracle* G, *q_{H} many queries to its oracle* H, *and at most q_{D} many queries to its decapsulation oracle* ODECAPS. *Let $q = q_{\mathsf{G}} + q_{\mathsf{H}}$, and let d and w be the query depth and query width of the combined queries to* G *and* H. *Then there exist an* IND-CPA *adversary $\mathcal{B}_{\mathsf{IND}}$, a* OW-CPA *adversary $\mathcal{B}_{\mathsf{OW}}$ and an* FFP-CPA *adversary $\mathcal{C}$ against* $\mathsf{PKE}^{\mathsf{G}}$ *in the* $\mathrm{eQROM}_{\mathsf{Enc}}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^{\perp}}(\mathcal{A}) \leq \widetilde{\mathrm{Adv}}_{\mathsf{PKE}} + (q_{\mathsf{D}} + 1)\mathrm{Adv}^{\mathsf{FFP\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{C}) + \varepsilon_{\gamma}, \text{ with} \tag{69}$$

$$\widetilde{\mathrm{Adv}}_{\mathsf{PKE}} = \begin{cases} 4 \cdot \sqrt{(d + q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND}})} + \frac{8(q+q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} & \text{or} \\ 8\,(d + q_{\mathsf{D}}) \cdot \sqrt{w \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW}})}. \end{cases} \tag{70}$$

The additive error term is given by

$$\varepsilon_\gamma = 24 q_{\mathsf{D}}(q_{\mathsf{G}} + 4 q_{\mathsf{D}}) 2^{-\gamma/2} \ .$$

$\mathcal{C}$ *makes* $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ *queries to* eCO.RO *and* q_{D} *to* eCO.E*. The running time of the adversaries* $\mathcal{B}_{\mathsf{IND}}$*,* $\mathcal{B}_{\mathsf{OW}}$ *and* $\mathcal{C}$ *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* $\mathrm{Time}(\mathcal{C}) = \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$*.*

Combining Corollary 8 with Theorem 10 from Section 6 and Theorem 11 from Section 6.1, we now obtain our main result as a corollary.

Corollary 9 (PKE FFP-NG and pass. secure $\Rightarrow$ $\mathsf{FO}_m^\perp$[PKE] IND-CCA). *Let* PKE *be a (randomized)* PKE *scheme that is* γ*-spread and with worst-case random-key decryption error rate* δ_{ik}*, decryption error rate variance* $\sigma_{\delta_{\mathrm{ik}}}$ *and decryption error tail envelope* τ*. Let* $\mathcal{A}$ *be an* IND-CCA-KEM *adversary (in the QROM) against* $\mathsf{KEM}_m^\perp := \mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$*, issuing at most* q_{G} *many queries to its oracle* G*,* q_{H} *many queries to its oracle* H*, and at most* q_{D} *many queries to its decapsulation oracle* ODECAPS*. Let* $q = q_{\mathsf{G}} + q_{\mathsf{H}}$*, and let* d *and* w *be the query depth and query width of the combined queries to* G *and* H*. Set* $C = 304$ *and assume* $\sqrt{C} q_{\mathsf{G}} \sigma_{\delta_{\mathrm{ik}}} \leq 1/2$*. Then there exist an* IND-CPA *adversary* $\mathcal{B}_{\mathsf{IND}}$*, a* OW-CPA *adversary* $\mathcal{B}_{\mathsf{OW}}$ *and an* FFP-NG *adversary* $\mathcal{C}$ *against* PKE *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{KEM}_m^\perp}(\mathcal{A}) \leq \widetilde{\mathrm{Adv}}_{\mathsf{PKE}} + (q_{\mathsf{D}} + 1) \left(2 \mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{C}) + \varepsilon_{\delta_{\mathrm{ik}}} \right) + \varepsilon_\gamma \tag{71}$$

with

$$\widetilde{\mathrm{Adv}}_{\mathsf{PKE}} = \begin{cases} 4 \cdot \sqrt{(d + q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND}})} + \frac{8(q + q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} & \text{and} \\ 8\,(d + q_{\mathsf{D}}) \cdot \sqrt{w \cdot \mathrm{Adv}^{\mathsf{OW}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW}})}. & \end{cases} \tag{72}$$

The additive error term $\varepsilon_{\delta_{\mathrm{ik}}}$ *is given by*

$$\varepsilon_{\delta_{\mathrm{ik}}} \leq \delta_{\mathrm{ik}} + (3 + 2\delta_{\mathrm{ik}}) \sqrt{C} q_{\mathsf{G}} \sigma_{\delta_{\mathrm{ik}}} \ , \tag{73}$$

and the additive error term ε_γ *is given by*

$$\varepsilon_\gamma = 24 q_{\mathsf{D}}(q_{\mathsf{G}} + 2 q_{\mathsf{D}}) 2^{-\gamma/2} + 4 q_{\mathsf{D}} \cdot 2^{-\gamma}.$$

The running time of the adversaries $\mathcal{B}_{\mathsf{IND}}$*,* $\mathcal{B}_{\mathsf{OW}}$ *and* $\mathcal{C}$ *is bounded by*

$$\mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}) + O(q_{\mathsf{D}}).$$

In 6.3 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. The corollary follows by combining Corollary 3 and Theorems 8 to 10. Exploiting the very mild condition $\sqrt{C} q_{\mathsf{G}} \sigma_{\delta_{\mathrm{ik}}} \leq 1/2$[7] we have used the inequality $x^2/\log(x) \leq x$ for $x \leq 1/2$ for $x = \sqrt{C} q_{\mathsf{G}} \sigma_{\delta_{\mathrm{ik}}}$ to simplify the error term $\varepsilon_{\delta_{\mathrm{ik}}}$ from Theorem 10. □

We remark that the two alternative bounds in Eqs. (63) and (73) are just examples. If, e.g., an exponential tail bound is available instead of a Gaussian one, the techniques from Section 6.2 can be used to prove a similar, intermediate bound. The above result has two main advantages over previous theorems for the FO transformation:

---

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

– The additive loss $(q_{\mathsf{D}}+1)\left(\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{C})+\varepsilon_{\delta_{\mathrm{ik}}}\right)$, with the two alternative bounds for $\varepsilon_{\delta_{\mathrm{ik}}}$ given in Eqs. (63) and (73), can be much smaller than the additive loss of roughly $q_{\mathsf{G}}^2\delta_{\mathrm{wc}}$ that is present in all previous bounds for the FO transformation. In particular, instead of the *quadratic* dependence on the number of hash queries q_{G}, the asymptotic dependence is at most *linear*. If an appropriate tail bound can be proven, it is even logarithmic.

– It holds for the explicit rejection variant of the transformation, while the bounds are competitive with previous ones in the literature that were limited to the implicit rejection variant.

8 γ-Spreadness of selected NIST proposals

Theorem 4 provides a tight reduction of IND-CCA-KEM to IND-CPA-KEM and FFP-CCA, albeit at the cost of an additive error depending on the spreadness factor γ of the underlying PKE. In this section, we will analyze the spreadness of some of the alternates candidates of the NIST post-quantum competition. Since this work is considered with schemes that exhibit decryption failure and get derandomized to a scheme $\mathsf{PKE}^{\mathsf{G}}$, we do not consider ClassicMcEliece, NTRU, NTRU prime and SIKE (since they are perfectly correct) and BIKE (as BIKE encrypts deterministically without incorporating $\mathsf{PKE}^{\mathsf{G}}$). We chose our two examples, HQC.PKE and FrodoPKE, because computing γ for these two examples requires little additional technical overhead. Computing γ for other submissions to the NIST PQC standardisation process, like, e.g., Kyber or Saber, is out of the scope of this work.

If q_{D} is upper bounded by 2^{64} as in NIST's CFP, we can give a simpler upper bound for the term showing up in Theorem 4 by computing

$$q_{\mathsf{D}}\cdot(q_{\mathsf{G}}+2q_{\mathsf{D}})\cdot 2^{-\gamma/2}\leq 2^{64}\cdot(q_{\mathsf{G}}+2^{65})\cdot 2^{-\gamma/2}\leq q_{\mathsf{G}}\cdot 2^{65-\gamma/2}\quad .$$

The following lemma makes the bound above explicit for FrodoPKE.

Lemma 6 (γ-Spreadness of FrodoPKE). FrodoPKE-i *is* γ*-spread for*

$$\gamma=\begin{cases}10752 & i=1344\\ 15616 & i=976\\ 10240 & i=640\end{cases}\quad ,$$

hence

$$q_{\mathsf{G}}\cdot 2^{65-\gamma/2}\leq\begin{cases}q_{\mathsf{G}}\cdot 2^{-5311} & i=1344\\ q_{\mathsf{G}}\cdot 2^{-7743} & i=976\\ q_{\mathsf{G}}\cdot 2^{-5055} & i=640\end{cases}\quad .$$

Proof. Let $(pk = (seed_A, B), sk) \in \mathrm{supp}(\mathsf{FrodoPKE.KG})$, let $m \in \mathsf{FrodoPKE}.\mathcal{M}$, and let $c = (B', V') \in \mathsf{FrodoPKE}.\mathcal{C}$. According to the definition of FrodoPKE.Enc, we have that

$$\begin{aligned}
&\Pr_{\mathsf{FrodoPKE.Enc}}[\mathsf{FrodoPKE.Enc}(pk, m) = (B', V')] \\
&= \Pr_{S',E' \leftarrow \chi^{\overline{m}\times n}, E'' \leftarrow \chi^{\overline{m}\times \overline{n}}}[S'A + E' = B' \ \wedge\ S'B + E'' + \mathsf{Frodo.Encode}(m) = V'] \\
&\leq \Pr_{S',E' \leftarrow \chi^{\overline{m}\times n}}[S'A + E' = B'] \\
&= \sum_{s' \in \mathrm{supp}(\chi^{\overline{m}\times n})} \Pr_{S',E' \leftarrow \chi^{\overline{m}\times n}}[S'A + E' = B' \wedge S' = s'] \\
&= \sum_{s' \in \mathrm{supp}(\chi^{\overline{m}\times n})} \Pr_{E' \leftarrow \chi^{\overline{m}\times n}}[s'A + E' = B'] \cdot \Pr_{S' \leftarrow \chi^{\overline{m}\times n}}[S' = s'] \\
&\leq \sum_{s' \in \mathrm{supp}(\chi^{\overline{m}\times n})} \Pr_{E' \leftarrow \chi^{\overline{m}\times n}}[E' = 0] \cdot \Pr_{S' \leftarrow \chi^{\overline{m}\times n}}[S' = s'] \\
&= \Pr_{E' \leftarrow \chi^{\overline{m}\times n}}[E' = 0] \leq \left(\Pr_{x \leftarrow \chi}[x = 0]\right)^{\overline{m}\times n} ,
\end{aligned}$$

where we applied the law of total probability and used the fact that χ is a symmetric distribution centered at zero.

We will now plug in the parameters of FrodoPKE-i: For all instantiations of i as specified in [NAB+20], $\overline{m} = 8$ and $n = i$. According to table 3 of [NAB+20], we furthermore have that

$$\Pr_{x \leftarrow \chi}[x = 0] = 2^{-16} \cdot \begin{cases} 18286 & i = 1344 \\ 11278 & i = 976 \\ 9288 & i = 640 \end{cases} < \begin{cases} 2^{-1} & i = 1344 \\ 2^{-2} & i \in \{976, 640\} \end{cases} .$$

Hence we obtain

$$\max_{c \in \mathsf{FrodoPKE}.\mathcal{C}} \Pr_{\mathsf{FrodoPKE.Enc}}[\mathsf{FrodoPKE.Enc}(pk, m) = c] \leq \begin{cases} 2^{-8 \cdot 1344} & i = 1344 \\ 2^{-16 \cdot i} & i \in \{976, 640\} \end{cases} .$$

The following lemma makes the bound above explicit for HQC.PKE.

Lemma 7 (γ-Spreadness of HQC.PKE). HQC.PKE-i *is* γ*-spread for*

$$\gamma = 2 \cdot \begin{cases} \log_2 \binom{57600}{149} > 1490 & i = 256 \\ \log_2 \binom{35840}{114} > 1105 & i = 192 \\ \log_2 \binom{17664}{75} > 694 & i = 128 \end{cases} ,$$

hence

$$q_{\mathsf{G}} \cdot 2^{65-\gamma/2} \leq \begin{cases} q_{\mathsf{G}} \cdot 2^{-1425} & i = 256 \\ q_{\mathsf{G}} \cdot 2^{-1040} & i = 192 \\ q_{\mathsf{G}} \cdot 2^{-629} & i = 128 \end{cases} .$$

Proof. Let $(pk = (h, s), sk) \in \text{supp}(\mathsf{HQC.PKE.KG})$, let $m \in \mathsf{HQC.PKE}.\mathcal{M}$, and let $c = (u, v) \in \mathsf{HQC.PKE}.\mathcal{C}$. According to the definition of $\mathsf{HQC.PKE.Enc}$, we have that

$$\Pr_{\mathsf{HQC.PKE.Enc}}[\mathsf{HQC.PKE.Enc}(pk, m) = (u, v)] = \Pr_{R_1, R_2 \leftarrow \mathcal{U}(S_{w_r}^{n_1 \cdot n_2}), E \leftarrow \mathcal{U}(S_{w_e}^{n_1 \cdot n_2})}[R_1 + h \cdot R_2 = u \ \wedge \ mG + s \cdot R_2 + E = v] \ ,$$

where $S_w^{n_1 \cdot n_2}$ denotes the subset of elements of hamming weight w in $\{0, 1\}^{n_1 \cdot n_2}$.

By the law of total probability,

$$\begin{aligned} &\Pr_{R_1, R_2 \leftarrow \mathcal{U}(S_{w_r}^{n_1 \cdot n_2}), E \leftarrow \mathcal{U}(S_{w_e}^{n_1 \cdot n_2})}[R_1 + h \cdot R_2 = u \ \wedge \ mG + s \cdot R_2 + E = v] \\ &= \sum_{r_2 \in S_{w_r}^{n_1 \cdot n_2}} \Pr_{R_1 \leftarrow \mathcal{U}(S_{w_r}^{n_1 \cdot n_2}), E \leftarrow \mathcal{U}(S_{w_e}^{n_1 \cdot n_2})}[R_1 = u - h \cdot r_2 \ \wedge \ E = v - (mG + s \cdot r_2)] \\ &\qquad \cdot \Pr_{R_2 \leftarrow \mathcal{U}(S_{w_r}^{n_1 \cdot n_2})}[R_2 = r_2] \\ &\leq \sum_{r_2 \in S_{w_r}^{n_1 \cdot n_2}} \frac{1}{\binom{n_1 \cdot n_2}{w_r}} \cdot \frac{1}{\binom{n_1 \cdot n_2}{w_e}} \cdot \Pr_{R_2 \leftarrow \mathcal{U}(S_{w_r}^{n_1 \cdot n_2})}[R_2 = r_2] = \frac{1}{\binom{n_1 \cdot n_2}{w_r}} \cdot \frac{1}{\binom{n_1 \cdot n_2}{w_e}} \ , \end{aligned}$$

where we used the fact that $|S_w^N| = \binom{N}{w}$ in the last line.

We will now plug in the parameters of $\mathsf{HQC.PKE}\text{-}i$: For the instantiations of i as specified in [MAB+21, Section 2.7], we have that

$$w_e = w_r = \begin{cases} 149 & i = 256 \\ 114 & i = 192 \\ 75 & i = 128 \end{cases} ,$$

and that

$$n_1 \cdot n_2 = \begin{cases} 90 \cdot 640 & i = 256 \\ 56 \cdot 640 & i = 192 \\ 46 \cdot 384 & i = 128 \end{cases} = \begin{cases} 57600 & i = 256 \\ 35840 & i = 192 \\ 17664 & i = 128 \end{cases} .$$

Hence we obtain

$$\max_{c \in \mathsf{HQC.PKE}.\mathcal{C}} \Pr_{\mathsf{HQC.PKE.Enc}}[\mathsf{HQC.PKE.Enc}(pk, m) = c] \leq \begin{cases} (\frac{1}{\binom{57600}{149}})^2 & i = 256 \\ (\frac{1}{\binom{35840}{114}})^2 & i = 192 \\ (\frac{1}{\binom{17664}{75}})^2 & i = 128 \end{cases} .$$

References

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

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 Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany.

BS20. Nina Bindel and John M. Schanck. Decryption failure is more likely after success. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 206–225, Paris, France, April 15–17 2020. Springer, Heidelberg, Germany.

Den03. Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 133–151, Cirencester, UK, December 16–18, 2003. Springer, Heidelberg, Germany.

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. https://eprint.iacr.org/2021/280, accepted for publication at Eurocrypt 2022.

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 Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 3–33, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

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. https://eprint.iacr.org/2018/1089.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany.

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 Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany.

HKSU20. Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020: 23rd International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 12111 of *Lecture Notes in Computer Science*, pages 389–422, Edinburgh, UK, May 4–7, 2020. Springer, Heidelberg, Germany.

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 Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany.

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 Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 703–728, Zagreb, Croatia, May 10–14, 2020. Springer, Heidelberg, Germany.

MAB+21. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jurjen Bos, Jean-Christophe Deneuville, Arnaud Dion, Philippe Gaborit, Jérôme Lacan, Edoardo Persichetti, Jean-Marc Robert, Pascal Véron, and Gilles Zémor. Hamming Quasi-Cyclic (HQC), 2021. https://pqc-hqc.org/doc/hqc-specification_2021-06-06.pdf.

NAB+20. Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Learning With Errors Key Encapsulation. Algorithm Specifications And Supporting Documentation, 2020. https://frodokem.org/files/FrodoKEM-specification-20200930.pdf.

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 Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 239–268, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Heidelberg, Germany.

A Overview: Relations between FO-like transformations

There exists a plethora of FO-like transformations, and one might wonder if a result for transformation variant X also is applicable to transformation variant Y. In order to systematize existing knowledge and to simplify such considerations, this section recaps known relations between the security properties of FO variants on a high level.

We will now revisit other well-known variants for the FO transformation, introduced by [HHK17] as $\mathsf{FO}_m^{\not\perp}$, $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}_{m,c}^{\perp}$. In all variants, the $^\perp$ and $^{\not\perp}$ stands for the way in which the KEMs reject ciphertexts that are not well-formed, i.e., ciphertexts that either fail to decrypt or whose decrypted plaintexts fail to re-encrypt: $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}_{m,c}^{\perp}$ will return a dedicated failure symbol $\perp$, $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}_{m,c}^{\not\perp}$ will instead use an additional hash function to compute from the ciphertext a deterministic, but pseudorandom value. Since this pseudorandom value does not communicate explicitly that the ciphertext was rejected, $\mathsf{FO}_m^{\not\perp}$ and $\mathsf{FO}_{m,c}^{\not\perp}$ are often called *FO with implicit rejection* (or a 'silent' KEM), and $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}_{m,c}^{\perp}$ are called *FO with explicit rejection*. In both $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}_m^{\not\perp}$, the $_m$ represents how the KEM computes its keys: the key is computed by simply feeding the message m into the key derivation oracle. In $\mathsf{FO}_{m,c}^{\perp}$ and $\mathsf{FO}_{m,c}^{\not\perp}$, the key instead is computed by including both message m and ciphertext c into the key derivation oracle's input. $\mathsf{FO}_{m,c}^{\perp}$ and $\mathsf{FO}_{m,c}^{\not\perp}$ are hence also called *ciphertext-contributing* variants.

At the time [HHK17] was written, all transformations above only had proofs in the classical ROM. In order to facilitate a proof that also holds against quantum attackers, [HHK17] further modified transformations $\mathsf{FO}_m^{\perp}$ and $\mathsf{FO}_m^{\not\perp}$ and denoted these modifications by $\mathsf{QFO}_m^{\perp}$ and $\mathsf{QFO}_m^{\not\perp}$, respectively. The only difference between $\mathsf{FO}_m^{\perp}/\mathsf{FO}_m^{\not\perp}$ and their Q counterpart is that during encapsulation, the ciphertext is concatenated with the hash value of m (using a length-preserving hash function), which is then used during decapsulation to perform an additional validity check. This additional hash value is often called *key confirmation tag*, and $\mathsf{QFO}_m^{\perp}$ and $\mathsf{QFO}_m^{\not\perp}$ are often called *FO with key confirmation*. Since appending a length-preserving hash value induces communicative overhead, and since the original proofs were highly non-tight, a lot of effort has been invested into improving on both aspects.

Fortunately, the situation can be simplified a bit: It was proven in [BHH+19] that for either rejection variant $\mathsf{FO} \in \{\mathsf{FO}^{\not\perp}, \mathsf{FO}^{\perp}\}$, it does not matter which mode of key derivation is chosen, since FO_m is as secure as $\mathsf{FO}_{m,c}$ and vice versa. (A summary of the proven relations is given in Fig. 16.) We can hence neglect this distinction and drop the subscript for the rest of this discussion. It was furthermore shown first in [SXY18] that $\mathsf{FO}^{\not\perp}$ is secure even against quantum attackers with bounds similar to $\mathsf{QFO}_m^{\not\perp}$, assuming that the underlying encryption scheme is perfectly correct. For schemes that are not perfectly correct, established strategies can be used to generalise the result from [SXY18]

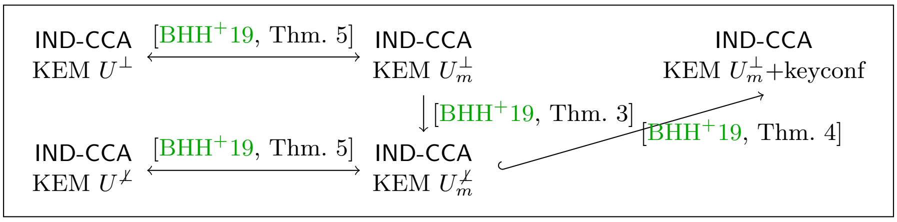

Fig. 16. Relations between the security of different types of U-constructions as shown in [BHH+19]. The hooked arrow indicates a theorem with an ϵ-injectivity constraint on the underlying deterministic scheme. Figure taken from [BHH+19] with updated references.

(e.g., see [HKSU20]). To achieve security against quantum attackers, we can hence dispense with the more costly 'key confirmation variant' $\mathsf{QFO}_m^{\not\perp}$ and simply use $\mathsf{FO}^{\not\perp}$. One might wonder if a similar result could be achieved for the explicit rejection variant $\mathsf{QFO}_m^{\perp}$, and while an asymptotic security proof for $\mathsf{FO}^{\perp}$ has already been established [Zha19, DFMS21], giving a proof for $\mathsf{FO}^{\perp}$ with bounds comparable to the bounds for $\mathsf{FO}^{\not\perp}$ was still an open problem until now. While it has been proven in [BHH+19] that security of $\mathsf{FO}^{\perp}$ implies security of $\mathsf{FO}^{\not\perp}$, and that security of $\mathsf{FO}^{\not\perp}$ implies security of $\mathsf{QFO}_m^{\perp}$, it was hence not clear until now whether explicit rejection variants might not turn out to be less robust against quantum attackers than their implicit rejection counterparts.

B Proof of Theorem 3 (From $\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$ or $\mathsf{OW\text{-}CPA}_{\mathsf{PKE}}$ to $\mathsf{IND\text{-}CPA}_{\mathsf{FO[PKE]}}$)

For easier reference, we repeat the statement of Theorem 3.

Theorem 3 (PKE OW-CPA **or** IND-CPA $\overset{\text{ROM}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}$[PKE] IND-CPA)**.** *Let* $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *for some PKE scheme* PKE*. For any* IND-CPA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}_m^{\perp}$*, issuing at most* q_{G} *many queries to its oracle* G *and* q_{H} *many queries to its oracle* H*, there exist an* OW-CPA *adversary* $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ *and an* IND-CPA *adversary* $\mathcal{B}_{\mathsf{IND\text{-}CPA}}$ *of roughly the same running time such that*

$$\mathrm{Adv}_{\mathsf{KEM}_m^{\perp}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) \leq (q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{OW}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}})$$

and

$$\mathrm{Adv}_{\mathsf{KEM}_m^{\perp}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) \leq 3 \cdot \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{2 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}}) + 1}{|\mathcal{M}|} \enspace .$$

Proof. In [HHK17], the security proof for $\mathsf{KEM}_m^{\not\perp} = \mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] = \mathsf{U}^{\perp}[\mathsf{PKE}^{\mathsf{G}}, \mathsf{H}]$ is modularized into one proof for $\mathsf{PKE}^{\mathsf{G}}$ and one proof for transformation $\mathsf{U}_m^{\perp}$, as sketched in Fig. 17. Here, One-Wayness under Validity checking Attacks (OW-VA) is an intermediate helper notion that models OW-CPA security in the presence of an additional Ciphertext Validity Oracle CVO that tells the attacker whether a ciphertext is valid. To avoid confusion when looking up the theorems, Theorems 3.1 and 3.2 actually prove a security notion stronger than OW-VA security, called OW-PCVA security. The OW-PCVA game is like the OW-VA one except that it provides one more additional oracle to the adversary. Since OW-PCVA security immediately implies OW-VA security by dismissing the additional oracle, and since Theorem 3.5 only requires OW-VA security, we omitt further details on OW-PCVA security.

[HHK17, Theorem 3.5] states that IND-CCA security of $\mathsf{KEM}_m^{\not\perp} = \mathsf{U}^{\perp}[\mathsf{PKE}^{\mathsf{G}}, \mathsf{H}]$ can be based on OW-VA security of $\mathsf{PKE}^{\mathsf{G}}$, tightly. Clearly, the same holds when IND-CCA is replaced with IND-CPA, as one can simply set the number q_D of decapsulation queries to 0. In fact, when we only need IND-CPA security, we can disregard all terms in the bound of [HHK17, Theorem 3.5] that stem from how the random oracle and the decapsulation oracle were changed during the proof in order for the the decapsulation oracle to be simulatable without the secret key. Dismissing the respective changes, we obtain from [HHK17, Theorem 3.5] that for any IND-CPA adversary $\mathcal{A}$ against $\mathsf{KEM}^{\perp}$, issuing at most $q_{\mathsf{G}}/q_{\mathsf{H}}$ many queries to its respective random oracles, there exist an OW-VA adversary $\tilde{\mathcal{A}}$ of roughly the same running time, issuing no queries to its oracle CVO, such that

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{KEM}^{\not\perp}_m}(\mathcal{A}) \leq \mathrm{Adv}^{\mathsf{OW\text{-}VA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{A}}) \ .$$

[HHK17, Theorem 3.1] states that OW-PCVA security of $\mathsf{PKE}^{\mathsf{G}}$ can be based on OW-CPA security of PKE, non-tightly. Since in our use case, we are only considering adversaries $\tilde{\mathcal{A}}$ that do not pose any queries to oracle CVO or the other oracle present in the OW-PCVA game, we can again disregard all terms in the bound of [HHK17, Theorem 3.1] that stem from how the additional oracles got simulated during the proof. Dismissing the simulation of the redundant additional oracles, we obtain from [HHK17, Theorem 3.1] that for any OW-VA adversary $\tilde{\mathcal{A}}$ against $\mathsf{PKE}^{\mathsf{G}}$ as the reduction above, there exist an OW-CPA adversary $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ of roughly the same running time such that

$$\mathrm{Adv}^{\mathsf{OW\text{-}VA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{A}}) \leq (q_{\mathsf{G}} + q_{\mathsf{H}} + 1) \cdot \mathrm{Adv}^{\mathsf{OW\text{-}VA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{OW\text{-}CPA}}) \ .$$

[HHK17, Theorem 3.2] states that OW-PCVA security of $\mathsf{PKE}^{\mathsf{G}}$ can be based on IND-CPA security of PKE, tightly. Again, we can dismiss the simulation of the redundant additional oracles and obtain from [HHK17, Theorem 3.2] that for any OW-VA adversary $\tilde{\mathcal{A}}$ against $\mathsf{PKE}^{\mathsf{G}}$ as the reduction above, there exist an IND-CPA adversary $\mathcal{B}_{\mathsf{OW\text{-}CPA}}$ of roughly the same running time such that

$$\mathrm{Adv}^{\mathsf{OW\text{-}VA}}_{\mathsf{PKE}^{\mathsf{G}}}(\tilde{\mathcal{A}}) \leq 3 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}_{\mathsf{IND\text{-}CPA}}) + \frac{2 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}}) + 1}{|\mathcal{M}|} \ .$$

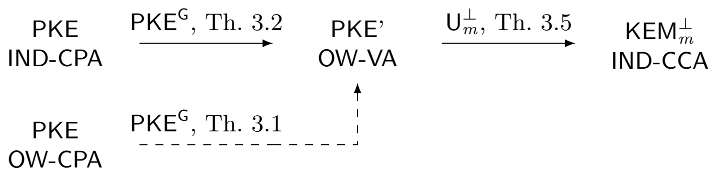

Fig. 17. Modular approach in [HHK17] for transformation $\mathsf{FO}^{\perp}_m$, in the ROM. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions. The used theorem numbers are the respective theorem numbers in [HHK17].

C Proof of Lemma 4 (OWTH: event prob. distances if ¬FIND etc)

For easier reference, we repeat the statement of Lemma 4.

Lemma 4. *Let* eCO^0 *and* 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* eCO^0*. Let* $\mathcal{S}$ *be the set of elements* $x \in \mathcal{X}$ *whose oracle values are needed to compute inp, and* *let* $\mathcal{T}_{\mathcal{S}} := \{t \mid \exists x \in \mathcal{S} \text{ s.th. } t = f(x, \mathsf{eCO}^0(x))\}$*. 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*

$$\Pr[E \wedge \neg FIND \wedge \neg EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] = \Pr[E \wedge \neg FIND \wedge \neg EXT : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}] \ , \tag{30}$$

$$|\Pr[E \wedge \neg FIND : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[E \wedge \neg FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| \leq \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] \ , \tag{31}$$

$$|\Pr[FIND : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| \leq \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] \ , \tag{32}$$

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

Proof. During this proof, we use $\mathcal{A}^{\mathsf{eCO} \setminus S}$ as a shorthand for $\mathcal{A}^{\mathsf{eCO} \setminus S}(inp)$ and F for FIND. As argued in Section 5.2, $\mathcal{A}$'s view is exactly the same in both games unless FIND or EXT occur, therefore Eq. (30) holds. We will first use Eq. (30) to prove Eq. (31): We have

$$\begin{aligned}
&\left|\Pr[\mathrm{E} \wedge \neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] - \Pr[\mathrm{E} \wedge \neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S}]\right| \\
&= \left|\Pr[\mathrm{E} \wedge \neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] - \Pr[\mathrm{E} \wedge \neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S}]\right| \\
&= \Big|\Pr[\mathrm{E} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S} | \neg \mathrm{F} \wedge \mathrm{EXT}] \cdot \Pr[\neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] \\
&\quad - \Pr[\mathrm{E} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S} | \neg \mathrm{F} \wedge \mathrm{EXT}] \cdot \Pr[\neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S}]\Big| \\
&\overset{(*)}{=} \left|\Pr[\mathrm{E} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S} | \neg \mathrm{F} \wedge \mathrm{EXT}] - \Pr[\mathrm{E} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S} | \neg \mathrm{F} \wedge \mathrm{EXT}]\right| \\
&\quad \cdot \Pr[\neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] \\
&\leq \Pr[\neg \mathrm{F} \wedge \mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] \leq \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] \ ,
\end{aligned}$$

where (*) used that if FIND does not occur, all case-depending information is hidden from $\mathcal{A}$ until EXT occurs, hence EXT is equally likely in that case and the common factor can hence be moved to outside of the absolute value.

To prove Eq. (32), it is sufficient to instead upper bound the difference between the probabilities of event ¬FIND for the two oracles: since the equation $\Pr[E] = 1 - \Pr[\neg E]$ holds for arbitrary events, we have that

$$|\Pr[\mathrm{F} : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[\mathrm{F} : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| = |\Pr[\neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] - \Pr[\neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]| \ .$$

The bound then follows directly from Eq. (31): We have that

$$\begin{aligned}
&|\Pr[\neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] - \Pr[\neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S}]| \\
&= |\Pr[\mathsf{true} \wedge \neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] - \Pr[\mathsf{true} \wedge \neg \mathrm{F} : \mathcal{A}^{\mathsf{eCO}^1 \setminus S}]| \\
&\overset{Eq.\ (31)}{\leq} \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0 \setminus S}] \ .
\end{aligned}$$

D Proof of Theorem 6 (OWTH: Distinguishing to Finding)

For easier reference, we repeat the statement of Theorem 6.

Theorem 6 (Semi-classical OWTH in the eQROM_f: Distinguishing to Finding). *Let* eCO^0, eCO^1, GenInp, $\mathcal{S}$, *FIND and EXT be like in Lemma 4. We define the* 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 taken over the coins of GenInp *and the internal randomness of* $\mathcal{A}$. *For any algorithm* $\mathcal{A}$ *of query depth d with respect to* $\mathsf{eCO.RO}$, *we have that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq & 4 \cdot \sqrt{d \cdot \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]} \\ & + 2 \cdot (\sqrt{d}+1) \cdot \sqrt{\Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0}] + \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^1}]} \ . \end{aligned} \tag{33}$$

In the special case where $\Pr[EXT : \mathcal{A}^{\mathsf{eCO}^0 \setminus \mathcal{S}}] = \Pr[EXT : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}] = 0$, *we obtain*

$$\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) \leq 4 \cdot \sqrt{d \cdot \Pr[FIND : \mathcal{A}^{\mathsf{eCO}^1 \setminus \mathcal{S}}]} \ . \tag{34}$$

Proof. In the following helper definitions, we will again use $\mathcal{A}^{\mathsf{O}}$ as a shorthand for $\mathcal{A}^{\mathsf{O}}(inp)$. For either oracle $\mathsf{eCO} \in \{\mathsf{eCO}^0, \mathsf{eCO}^1\}$, we let

$$\begin{aligned} p_b &:= \Pr[1 \leftarrow \mathcal{A}^{\mathsf{eCO}^b}] \\ p_{b,\neg\mathrm{EXT}} &:= \Pr[b' = 1 \wedge \neg\mathrm{EXT} : b' \leftarrow \mathcal{A}^{\mathsf{eCO}^b}] \\ p_{b,\neg\mathrm{EXT},\neg\mathrm{F}} &:= \Pr[b' = 1 \wedge \neg\mathrm{FIND} \wedge \neg\mathrm{EXT} : b' \leftarrow \mathcal{A}^{\mathsf{eCO}^b \setminus \mathcal{S}}] \\ p_{b,\neg\mathrm{EXT},\mathrm{F}} &:= \Pr[\mathrm{FIND} \wedge \neg\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^b \setminus \mathcal{S}}] \ . \end{aligned}$$

In order to prove Theorem 6, we want to bound $\mathrm{Adv}^{\mathsf{OWTH}}_{\mathrm{eQRO}_f}(\mathcal{A}) = |p_0 - p_1|$. Applying the triangle inequality yields

$$\begin{aligned} |p_0 - p_1| \leq & |p_0 - p_{0,\neg\mathrm{EXT}}| + |p_1 - p_{1,\neg\mathrm{EXT}}| + |p_{0,\neg\mathrm{EXT}} - p_{1,\neg\mathrm{EXT}}| \\ & \overset{(*)}{\leq} \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0}] + \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^1}] \\ & \quad + |p_{0,\neg\mathrm{EXT}} - p_{1,\neg\mathrm{EXT}}| \ , \end{aligned} \tag{74}$$

where (*) used that $|p_b - p_{b,\neg\mathrm{EXT}}| = \Pr[b' = 1 \wedge \mathrm{EXT} : b' \leftarrow \mathcal{A}^{\mathsf{eCO}^b}] \leq \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^b}]$, it hence remains to bound $|p_{0,\neg\mathrm{EXT}} - p_{1,\neg\mathrm{EXT}}|$.

We claim that for either oracle $\mathsf{eCO} \in \{\mathsf{eCO}^0, \mathsf{eCO}^1\}$, we have that

$$|p_{b,\neg\mathrm{EXT}} - p_{b,\neg\mathrm{EXT},\neg\mathrm{F}}| \leq 2 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^b \setminus \mathcal{S}}]} \ . \tag{75}$$

Assuming that claim (75) is true, we can then once more apply the triangle inequality to obtain

$$\begin{aligned}
|p_{0,\neg\mathrm{EXT}} - p_{1,\neg\mathrm{EXT}}| &\le |p_{0,\neg\mathrm{EXT}} - p_{0,\neg\mathrm{EXT},\neg\mathrm{F}}| + |p_{1,\neg\mathrm{EXT}} - p_{0,\neg\mathrm{EXT},\neg\mathrm{F}}| \\
&\overset{(*)}{=} |p_{0,\neg\mathrm{EXT}} - p_{0,\neg\mathrm{EXT},\neg\mathrm{F}}| + |p_{1,\neg\mathrm{EXT}} - p_{1,\neg\mathrm{EXT},\neg\mathrm{F}}| \\
&\overset{(75)}{\le} 2 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}]} + 2 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]} \\
&\overset{(**)}{\le} 2 \cdot \sqrt{d \cdot (\Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}] + \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0}])} + 2 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]} \\
&\le 4 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}]} + 2 \cdot \sqrt{d \cdot \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0}]} \ . \qquad (76)
\end{aligned}$$

Here, (*) replaced $p_{0,\neg\mathrm{EXT},\neg\mathrm{F}}$ with $p_{1,\neg\mathrm{EXT},\neg\mathrm{F}}$ in the last term, using Eq. (30) from Lemma 4 which states that all events are equally likely regardless which oracle is used if neither EXT nor FIND occur. (**) used Eq. (32) from Lemma 4 which states that $\Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^0\backslash\mathcal{S}}]$ can be upper bounded by $\Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^1\backslash\mathcal{S}}] + \Pr[\mathrm{EXT} : \mathcal{A}^{\mathsf{eCO}^0}]$, and then used that the square root function is monotone increasing.

Plugging Eq. (76) into Eq. (74) yields the bound claimed in Theorem 6, it hence remains to prove Eq. (75), which we break down into the following steps: Due to the deferred measurement principle, both the puncturing operation O_S^{SC} and extraction oracle $\mathsf{eCO.E}$ can be rewritten such that they consist of a unitary, acting on the adversary-oracle registers and an additional measurement outcome register, and a final measurement of the outcome register at the end of the execution of $\mathcal{A}$. We will denote the respective outcome registers by L_F (for 'finding') and L_E (for extractions). Second, show that it suffices to bound the distance of the states before this final measurement of L_F and L_E. Third, show that it suffices to bound the distance of the states for any fixed instantiation of set S, oracle values $(y_x)_{x\in S}$, and input string inp. Lastly, prove the distance bound for any fixed instantiation by considering that the two states that emerge from the same initial state; and that the two chains of state transitions only increase the distance in terms of the probability that FIND occurs.

To flesh out this summary, we will first write O_S^{SC} as a concrete combination of unitaries and measurements: A ´logging' register L_F holding bitstrings of length d is initialised in state $|0\cdots 0\rangle$. Intuitively, L_F will log at its i-th position if the i-th query triggered FIND: Whenever $\mathcal{A}$ performs an oracle query, say it is the i-th, we slot in a unitary U_S^i that marks in the i-th position of L_F whether the query register holds an element of S. More formally, U_S^i acts on query register $X = X_1 \cdots X_w$ (recall that $\mathcal{A}$ can issue parallel oracle queries) and logging register L_F by

$$U_S^i \, |x_1, \cdots, x_w\rangle_X \, |b_1, \cdots, b_d\rangle_{L_F} := \begin{cases} |x_1, \cdots, x_w\rangle_X \, |b_1, \cdots, b_d\rangle_{L_F} & x_j \notin S \, \forall j \\ |x_1, \cdots, x_w\rangle_X \, |\mathrm{flip}_i(b_1, \cdots, b_d)\rangle_{L_F} & \exists j : x_j \in S \end{cases} .$$

Processing oracle queries according to $\mathsf{eCO}\backslash\mathcal{S}$ consists of first applying U_S^i to X and L_F, then measuring L_F in the computational basis, and then applying the oracle unitary O_{XYD} (see Section 4 for a brief description how parallel queries are answered).

Next, we also write $\mathsf{eCO.E}$ for function $f : X \times Y \to \{0,1\}^\ell$ as a concrete combination of unitaries and measurements: Let A be the register that holds the state of $\mathcal{A}$, and let D be the oracle database register. Note that A contains a register L_E that accommodates q_{E} many elements of X, which is used to log the outcome of the i-th query to $\mathsf{eCO.E}$ at its i-th position. Whenever $\mathcal{A}$ performs a query to $\mathsf{eCO.E}$, say it is the i-th, we apply a unitary U_f^i that adds to the i-th position of L_E the extraction outcome. More formally, U_f^i acts on query register T, database register D and register L_E by

$$U_f^i \, |t\rangle_T := |t\rangle_T \otimes \sum_{x \in X \cup \{\perp\}} \Sigma_{x,t} \otimes U_{f,x}^i \ ,$$

where $\Sigma_{x,t}$ acts on D and is defined by

$$\Sigma_{x,t} := \begin{cases} \bigotimes_{x'<x} \left(\sum_{y\in Y: f(x,y)\neq t} |y\rangle\langle y|_{D_{x'}}\right) \otimes \left(\sum_{y\in Y: f(x,y)=t} |y\rangle\langle y|_{D_x}\right) & x \in X \\ \mathrm{id} - \sum_{x\in X} \Sigma_{x,t} & x = \bot \end{cases} ,$$

and $U^i_{f,x}$ acts on L_E by

$$U^i_{f,x} |x_1, \cdots, x_{q_\mathrm{E}}\rangle_{L_E} := |x_1, \cdots, x_{i-1}, x_i + x, x_{i+1}, \cdots, x_{q_\mathrm{E}}\rangle_{L_E} \quad .$$

Processing the i-th extraction query according to $\mathsf{eCO.E}$ consists of first applying U^i_f to T, D and L_E, and then measuring L_E in the computational basis.

We can now lift the final joint adversary-oracle state ρ'_0 of $\mathcal{A}$, when run with access to original oracle eCO^0, to the joint adversary-oracle-log state $\rho''_0 := \rho'_0 \otimes |0\cdots 0\rangle\langle 0\cdots 0|_{L_F}$. (Note that L_F is initialised to and will maintain to be in state $|0\cdots 0\rangle$.) We will furthermore denote by ρ''_1 the joint adversary-oracle-log state when $\mathcal{A}$ is run with access to $\mathsf{eCO}^1\backslash\mathcal{S}$. This means that ρ'_0 is the final state of $\mathcal{A}$ *without* puncturing, and ρ''_1 is the final state of $\mathcal{A}$ *with* puncturing. Let M be the measurement that measures, given the registers A, D, L_F, whether $\mathcal{A}$ outputs 1, EXT did not occur, and L_F is equal to $|0\cdots 0\rangle$, the latter meaning that FIND did not happen. Let $P_M(\Phi)$ denote the probability that M returns 1 when measuring a state Φ. As our arguments will work for both oracle cases, we will simply write $p_{\neg\mathrm{EXT}}$ instead of $p_{b,\neg\mathrm{EXT}}$ and $p_{\mathrm{EXT},\neg\mathrm{F}}$ instead of $p_{b,\neg\mathrm{EXT},\neg\mathrm{F}}$. We have that $p_{\neg\mathrm{EXT}} = P_M(\rho''_0)$ and that $p_{\neg\mathrm{EXT},\neg\mathrm{F}} = P_M(\rho''_1)$, hence we want to upper bound

$$|p_{\neg\mathrm{EXT}} - p_{\neg\mathrm{EXT},\neg\mathrm{F}}| = |P_M(\rho''_0) - P_M(\rho''_1)| \ , \tag{77}$$

and due to [AHU19, Lemma 4], we know that

$$|P_M(\rho''_0) - P_M(\rho''_1)| \le B(\rho''_0, \rho''_1) \ , \tag{78}$$

where B is the Bures distance. I.e., for two density operators τ_1 and τ_2, $B(\tau_1, \tau_2) := \sqrt{2 - 2F(\tau_1, \tau_2)}$, and the fidelity F is defined by $F(\tau_1, \tau_2) := \mathrm{Tr}\sqrt{\sqrt{\tau_1}\tau_2\sqrt{\tau_1}}$. According to the definition of the Bures distance,

$$B(\rho''_0, \rho''_1)^2 = 2(1 - F(\rho''_0, \rho''_1)) \ .$$

Combining Eq. (78)) with Eq. (78) and plugging in the definition of the Bures distance hence yields

$$|p_{\neg\mathrm{EXT}} - p_{\neg\mathrm{EXT},\neg\mathrm{F}}| \le \sqrt{2(1 - F(\rho''_0, \rho''_1))} \ .$$

To show that $|p_{b,\neg\mathrm{EXT}} - p_{b,\neg\mathrm{EXT},\neg\mathrm{F}}| \le 2 \cdot \sqrt{d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}^b\backslash\mathcal{S}}]}$, it hence suffices to prove that

$$F(\rho''_0, \rho''_1) \ge 1 - 2d \cdot \Pr[\mathrm{FIND} : \mathcal{A}^{\mathsf{eCO}\backslash\mathcal{S}}(inp)] \ . \tag{79}$$

To lower bound $F(\rho''_0, \rho''_1)$, we make the following observation: The measurements performed by O^{SC}_S and $\mathsf{eCO.E}$ can be delayed, i.e., when processing an oracle query, we apply the respective unitary U^i_S, but do not perform the measurement of L_F. Similarly, when performing extraction queries, we apply the respective unitary U^i_f, but do not perform the measurement of L_F. Instead, we perform a measurement of L_F and L_E in the end, which we will denote by $\mathcal{E}_{L_F,L_E}$. Let ρ_0 denote the final state of $\mathcal{A}$, when run with access to original oracle eCO^0, but without the extraction measurements. Since the 'FIND' register L_F of ρ''_0 will never be touched as ρ''_0 represents the case where no puncturing is performed, ρ_0'' is stable under 'FIND' measurements, we hence have that $\rho_0'' = \mathcal{E}_{L_F,L_E}(\rho_0)$. Let ρ_1 denote the final state of $\mathcal{A}$ when run with access to $\mathsf{eCO}\backslash\mathcal{S}$, but without the final measurement $\mathcal{E}_{L_F,L_E}$, meaning $\rho_1'' = \mathcal{E}_{L_F,L_E}(\rho_1)$. Using monotonicity of fidelity, we obtain

$$F(\rho_0'', \rho_1'') = F(\mathcal{E}_{L_F,L_E}(\rho_0), \mathcal{E}_{L_F,L_E}(\rho_1)) \geq F(\rho_0, \rho_1) \ .$$

We will now break down the fidelity term $F(\rho_0, \rho_1)$ into an expected value for instances of set S, oracle values $\overrightarrow{y} := (y_x)_{x \in S}$ and input inp: Let $\mathsf{GenInst}$ denote the sampling of an instance $ins := (S, (y_x)_{x\in S}, inp)$ according to their distribution. Let $|\phi_0^{ins}\rangle$ be the pure state corresponding to ρ_0 that would be obtained by running $\mathcal{A}$ with a fixed instance ins. Similarly, let $|\phi_1^{ins}\rangle$ be the state corresponding to ρ_1. Then $\rho_0 = \mathbb{E}_{ins}[||\phi_0^{ins}\rangle\rangle\langle|\phi_0^{ins}\rangle|]$, and $\rho_1 = \mathbb{E}_{ins}[||\phi_1^{ins}\rangle\rangle\langle|\phi_1^{ins}\rangle|]$. Hence we can identify

$$\begin{aligned} F(\rho_0, \rho_1) &= F(\mathop{\mathbb{E}}_{ins} |\phi_0^{ins}\rangle\langle\phi_0^{ins}| \, , \mathop{\mathbb{E}}_{ins} |\phi_1^{ins}\rangle\langle\phi_1^{ins}|) \\ &\overset{(*)}{\geq} \mathop{\mathbb{E}}_{ins} F(|\phi_0^{ins}\rangle\langle\phi_0^{ins}| \, , |\phi_1^{ins}\rangle\langle\phi_1^{ins}|) \overset{(**)}{\geq} 1 - \frac{1}{2} \mathop{\mathbb{E}}_{ins} \left\| |\phi_0^{ins}\rangle - |\phi_1^{ins}\rangle \right\|^2 \ , \end{aligned}$$

Here, (*) follows from the joint concavity of the fidelity, and (**) uses the fact that for any two normalised states $|\Psi\rangle$ and $|\Phi\rangle$, we have that $F(|\Psi\rangle\langle\Psi| \, , |\Phi\rangle\langle\Phi|) \geq 1 - \frac{1}{2} \, \||\Psi\rangle - |\Phi\rangle\|^2$. (This was proven in [AHU19, Lemma 3]).

In order to prove Eq. (79), it hence remains to show that for any instantiation $ins = (S, \overrightarrow{y}, inp)$, it holds that

$$\left\| |\phi_0^{ins}\rangle - |\phi_1^{ins}\rangle \right\|^2 \leq 4d \cdot P_{\mathrm{FIND}}^{ins} \ , \tag{80}$$

where P_{FIND}^{ins} denotes the probability of measuring the L_F register of the final state $|\phi_1^{ins}\rangle$ resulting in anything else than $|0, \cdots, 0\rangle$. For the rest of the proof, we hence consider $ins = (S, \overrightarrow{y}, inp)$ to be fixed and omit the indices from our notation.

Both final states $|\phi_0\rangle$ and $|\phi_1\rangle$ result from a chain of state transitions, applied to the same initial state $|\Phi^{(0)}\rangle$. Out of these transitions, d many represent oracle queries and hence are either of the form $T_0^{(j)} = U_A \circ O_{XYD}$ (to end up with $|\phi_0\rangle$), where U_A models the adversary's behaviour, or of the form $T_1^{(j)} = U_A \circ O_{XYD} \circ U_S^{i_j}$ for some i_j (to end up with $|\phi_1\rangle$). The remaining q_{E} many transitions represent extraction queries and are of the form $T_0^{(j)} = T_1^{(j)} = U_A \circ U_f^{i_j}$ for some i_j, where U_A again models the adversary's behaviour. Let $|\Phi_0^j\rangle$ denote the j-th intermediate state on the way to final state $|\phi_0\rangle$, i.e., let $|\Phi_0^j\rangle = T_0^{(j)} \circ T_0^{(j-1)} \circ \cdots \circ T_0^{(0)} \, |\Phi^{(0)}\rangle$, and let $|\Phi_1^j\rangle$ denote the j-th intermediate state on the way to final state $|\phi_1\rangle$, i.e., let $|\Phi_1^j\rangle = T_1^{(j)} \circ T_1^{(j-1)} \circ \cdots \circ T_1^{(1)} \, |\Phi^{(0)}\rangle$. Furthermore, let ϵ_j denote the distance between these intermediate states, i.e., $\epsilon_j := \left\| |\Phi_0^j\rangle - |\Phi_1^j\rangle \right\|$. With this notation, we have that

$$\||\phi_0\rangle - |\phi_1\rangle\| = \epsilon_{d+q_{\mathrm{E}}} = \sum_{j=1}^{d+q_{\mathrm{E}}} \epsilon_j - \epsilon_{j-1} \ ,$$

hence

$$\||\phi_0\rangle - |\phi_1\rangle\|^2 \leq \left(\sum_{j=1}^{d+q_{\mathrm{E}}} |\epsilon_j - \epsilon_{j-1}| \right)^2 \ . \tag{81}$$

We will now bound this sum by bounding the summands $|\epsilon_j - \epsilon_{j-1}|$, depending on which kind of query they represent. To this end, let $Q_{\mathrm{Or}} \subset \{1, \cdots, d + q_{\mathrm{E}}\}$ be the index set of oracle queries, i.e., the set of indices j such that $T_b^{(j)} = U_A \circ O_{XYD} \circ (U_S^{i_j})^b$ for some i_j, and let $Q_{\mathrm{Ext}} \subset \{1, \cdots, d + q_{\mathrm{E}}\}$ be the index set of extraction queries, i.e., the set of indices j such that $T_b^{(j)} = U_A \circ U_f^{i_j}$ for some i_j. We claim that for any oracle query, i.e., for any $j \in Q_{\mathrm{Or}}$, we have that

$$|\epsilon_j - \epsilon_{j-1}| \leq 2 \cdot \left\| P_S \left| \Phi_1^{j-1} \right\rangle \right\| \ , \tag{82}$$

where P_S is the projector unto the subspace spanned by S. For any extraction query, i.e., for any $j \in Q_{\mathrm{Ext}}$, we furthermore claim that

$$\epsilon_j = \epsilon_{j-1} \ . \tag{83}$$

Plugging claims (82) and (83) into Eq. (81), we obtain

$$\begin{aligned} \| |\phi_0\rangle - |\phi_1\rangle \|^2 &\leq \left(\sum_{j=1}^{d+q_{\mathrm{E}}} |\epsilon_j - \epsilon_{j-1}| \right)^2 \leq \left(\sum_{j \in Q_{\mathrm{Or}}} 2 \cdot \left\| P_S \left| \Phi_1^{j-1} \right\rangle \right\| \right)^2 \\ &\overset{(*)}{\leq} 4d \cdot \left(\sum_{j \in Q_{\mathrm{Or}}} \left\| P_S \left| \Phi_1^{j-1} \right\rangle \right\|^2 \right) \overset{(**)}{\leq} 4d \cdot (P_{\mathrm{FIND}}) \ , \end{aligned}$$

where (*) used Jensen's inequality; and (**) used that P_S is precisely the measurement operator corresponding to the event FIND.

It hence remains to prove claims (82) and (83). In order to prove claim (82), note that for $j \in Q_{\mathrm{Or}}$, we have that

$$\begin{aligned} \epsilon_j &= \left\| |\Phi_0^j\rangle - |\Phi_1^j\rangle \right\| = \left\| U_A \circ O_{XYD} \left| \Phi_0^{j-1} \right\rangle - U_A \circ O_{XYD} \circ U_S^{i_j} \left| \Phi_1^{j-1} \right\rangle \right\| \\ &= \left\| U_A \circ O_{XYD} \left(\left| \Phi_0^{j-1} \right\rangle - U_S^{i_j} \left| \Phi_1^{j-1} \right\rangle \right) \right\| \overset{(*)}{=} \left\| \left| \Phi_0^{j-1} \right\rangle - U_S^{i_j} \left| \Phi_1^{j-1} \right\rangle \right\| \\ &\leq \left\| \left| \Phi_0^{j-1} \right\rangle - \left| \Phi_1^{j-1} \right\rangle \right\| + \left\| \left| \Phi_1^{j-1} \right\rangle - U_S^{i_j} \left| \Phi_1^{j-1} \right\rangle \right\| = \epsilon_{j-1} + \left\| (\mathrm{id} - U_S^{i_j}) \left| \Phi_1^{j-1} \right\rangle \right\| \ , \end{aligned}$$

where (*) used that U_A and O_{XYD} are unitaries. Using that id and $U_S^{i_j}$ coincide on the image of $(\mathrm{id} - P_S)$, we can identify

$$\left\| (\mathrm{id} - U_S^{i_j}) \left| \Phi_1^{j-1} \right\rangle \right\| = \left\| (\mathrm{id} - U_S^{i_j}) P_S \left| \Phi_1^{j-1} \right\rangle \right\| \leq \left\| (\mathrm{id} - U_S^{i_j}) \right\|_\infty \left\| P_S \left| \Phi_1^{j-1} \right\rangle \right\| \leq 2 \cdot \left\| P_S \left| \Phi_1^{j-1} \right\rangle \right\| \ ,$$

where the second-to-last inequality holds by definition of the operator norm, and the last follows from the triangle inequality.

In order to prove claim (83), note that

$$\epsilon_j = \left\| |\Phi_0^j\rangle - |\Phi_1^j\rangle \right\| = \left\| U_A \circ U_f^{i_j} \left| \Phi_0^{j-1} \right\rangle - U_A \circ U_f^{i_j} \left| \Phi_1^{j-1} \right\rangle \right\| \overset{(*)}{=} \left\| \left| \Phi_0^{j-1} \right\rangle - \left| \Phi_1^{j-1} \right\rangle \right\| \ ,$$

where (*) used that U_A and U_f^i are unitaries.

□

Proof. By definition of the FFP-CPA advantage, we have

$$\mathrm{Adv}_{\mathsf{PKE}^{\mathsf{G}}}^{\mathsf{FFP\text{-}CPA}}(\mathcal{A}) = \Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk, pk)] \ .$$

To upper bound this probability, we begin by defining FFP-NG adversary $\mathcal{B}$: On input pk, $\mathcal{B}$ runs $\mathcal{A}(pk)$, simulating eCO to $\mathcal{A}$. When $\mathcal{A}$ finishes by outputting its message m, $\mathcal{B}$ computes $r :=$

$\mathsf{eCO.RO}(m)$, uses its failure-checking oracle to compute $b' := \mathsf{FCO}_b(m, r)$ and outputs b'. In the case where the challenge bit b of $\mathcal{B}$'s FFP-NG game is 0, $\mathcal{B}$ perfectly simulates the FFP-CPA game to $\mathcal{A}$ and wins iff $\mathcal{A}$ wins in game FFP-CPA. Therefore,

$$\Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk, pk)] = \Pr[1 \leftarrow \mathcal{B}(pk)|b = 0]$$
$$\leq \Pr[1 \leftarrow \mathcal{B}(pk)|b = 1] + 2\mathrm{Adv}^{\mathsf{FFP\text{-}NG}}_{\mathsf{PKE}}(\mathcal{B}) \ ,$$

where the last line used the definition of the FFP-NG advantage.

To upper bound $\Pr[1 \leftarrow \mathcal{B}(pk)|b = 1]$, note that this probability formalizes $\mathcal{A}$ outputting a message that fails to decrypt, but under an independently drawn key pair (sk', pk'):

$$\Pr[1 \leftarrow \mathcal{B}(pk)|b = 1] = \Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk', pk')] \ , \tag{84}$$

where the probability is taken additionally over $(sk', pk') \leftarrow \mathsf{KG}$.

To upper bound this probability, we define FFP-NK adversary $\mathcal{C}^{\mathsf{eCO}}$ against $\mathsf{PKE}^{\mathsf{G}}$: Upon initialisation, $\mathcal{C}$ computes a key pair (pk, sk) on its own and runs $\mathcal{A}^{\mathsf{eCO}}(pk)$. When $\mathcal{A}$ finishes by outputting its message m, $\mathcal{C}$ forwards the message to its own game. Since $\mathcal{C}$ perfectly simulates the game in Eq. (84) to $\mathcal{A}$ and wins iff $\mathcal{A}$ wins,

$$\Pr_{m \leftarrow \mathcal{A}^{\mathsf{eCO}}(pk)}[(m, \mathsf{eCO.RO}(m)) \text{ fails wrt. } (sk', pk')] = \mathrm{Adv}^{\mathsf{FFP\text{-}NK}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) \ .$$

□