A note on *Failing gracefully*: Completing the picture for explicitly rejecting Fujisaki-Okamoto transforms using worst-case correctness

Kathrin Hövelmanns[1] and Christian Majenz[2]

[1] Eindhoven University of Technology, The Netherlands
[2] Department of Applied Mathematics and Computer Science, Technical University of Denmark
kathrin@hoevelmanns.net

Abstract. The Fujisaki-Okamoto (FO) transformation is used in most proposals for post-quantum secure key encapsulation mechanisms (KEMs) like, e.g., Kyber [BDK+18]. The security analysis of FO in the presence of quantum attackers has made huge progress over the last years. Recently, [HHM22] made a particular improvement by giving a security proof that is agnostic towards how invalid ciphertexts are being treated: in contrast to previous proofs, it works regardless whether invalid ciphertexts are rejected by reporting decryption failure explicitly or implicitly (by returning pseudorandom values).
The proof in [HHM22] involves a new correctness notion for the encryption scheme that is used to encapsulate the keys. This allows in principle for a smaller additive security related to decryption failures, but requires to analyze this new notion for the encryption scheme on which a concrete KEM at hand is based.
This note offers a trade-off between [HHM22] and its predecessors: it offers a bound for both rejection variants, being mostly based on [HHM22], but uses a more established correctness notion.

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

1 Introduction

The Fujisaki-Okamoto (FO) transform [FO99, FO13, Den03] has become the de-facto standard to build secure KEMs. In particular, it was used in most KEM submissions to the NIST PQC standardisation process [NIS17]. In the context of post-quantum security, however, two novel issues surfaced:

1. Many of the PKE schemes used to encapsulate keys occasionally fail to decrypt a ciphertext to its plaintext (they do not have perfect correctness), and decryption failures have been shown [DGJ+19, BS20, DRV20, FKK+22] to impact security.
2. To rule out quantum attacks, the security proofs have to be done in the quantum-accessible random oracle model (QROM).

Both issues were tackled in [HHK17] and follow-up work (e.g., [SXY18, JZC+18, BHH+19, HKSU20, KSS+20, HHM22]). The QROM proofs prior to [HHM22], however, had a particular quirk: To avoid extreme additional reduction losses, they required the scheme to *reject implicitly*, that is, to return pseudorandom session keys instead of simply reporting an error when presented with a malformed ciphertext.

The FO transformation. Before discussing the goal of this note, we briefly recall the FO KEM transformation as introduced in [Den03] and revisited as $\mathsf{FO}_m^{\perp}$ by [HHK17]. $\mathsf{FO}_m^{\perp}$ constructs a KEM from a public-key encryption scheme PKE by first modifying PKE to obtain a deterministic scheme $\mathsf{PKE}^{\mathsf{G}}$, and then applying a PKE-to-KEM transformation ($\mathsf{U}_m^{\perp}$ in [HHK17]) to $\mathsf{PKE}^{\mathsf{G}}$:

Derandomised scheme $\mathsf{PKE}^{\mathsf{G}}$. Starting from PKE and a hash function G, $\mathsf{PKE}^{\mathsf{G}}$ encrypts messages m according to the encryption algorithm Enc of PKE, using the hash value $\mathsf{G}(m)$ as the random coins for Enc:

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

$\mathsf{Dec}^{\mathsf{G}}$ uses the decryption algorithm Dec of PKE to decrypt a ciphertext c to plaintext m'. $\mathsf{Dec}^{\mathsf{G}}$ rejects by returning failure symbol $\perp$ if c fails to decrypt or m' fails to encrypt back to c. (The formal definition is recalled on page 12).

PKE-to-KEM transformation $\mathsf{U}_m^{\perp}$. Starting from a deterministic encryption scheme PKE' and a hash function H, key encapsulation algorithm $\mathsf{KEM}_m^{\perp} := \mathsf{U}_m^{\perp}[\mathsf{PKE}', \mathsf{H}]$ encapsulates a key K via a ciphertext c 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 returns $K := \mathsf{H}(\mathsf{Dec}'(c))$ unless c fails to decrypt, in which case it returns failure symbol $\perp$.

The role of correctness errors. The impact of correctness errors on security is reflected in hindrances when trying to show that FO-transformed KEMs are IND-CCA secure: During the proofs, the decapsulation oracle oDecaps is replaced with a simulation. This simulation, however, is "too good" – it accurately decrapsulates ciphertexts for which the real oDecaps would fail. In other words, the change from the honest to a simulated decapsulation oracle is noticeable to attackers if they manage to craft a ciphertext where the honest decapsulation fails detectably. In [HHK17], the resulting advantage in distinguishing oDecaps from its simulation was dealt with in two steps:

1. Bound it via a 'break-correctness' game COR. COR asks the adversary, equipped with the complete key pair *including the secret key*, to produce a plaintext m such that $\mathsf{Enc}^{\mathsf{G}}(m)$ fails to decrypt.
2. Bound the maximal COR advantage in terms of a statistical 'worst-case' quantity δ_{wc} of the underlying scheme PKE. δ_{wc} is the maximal probability for plaintexts to cause decryption failure, averaged over the key pair.

This lead to a typical search bound, as the adversary can use the secret key to check if ciphertexts fail.

Correctness treatment in [HHM22] and open question. A central motivation of [HHM22] was that it is hard to estimate concrete δ_{wc}-bounds for particular schemes without relying on heuristics, and that it might be easier to estimate bounds for notions in which the attacker does not obtain the secret key.

[HHM22] therefore introduced a new family of correctness games that represent the search for failing plaintexts *without* the secret key, called *Find Failing Plaintext* (FFP) games, and then related the respective advantages to properties of the underlying encryption scheme PKE (see Fig. 1):

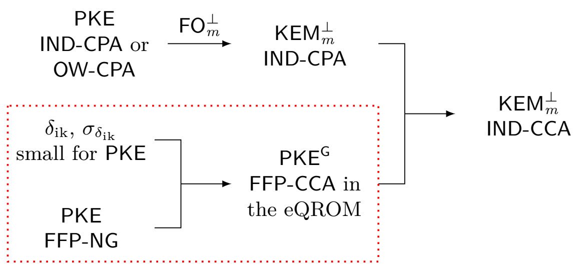


Fig. 1. Simplification of Figure 1 in [HHM22].The red-dotted part introduces new analysis tasks for KEM designers.

The resulting correctness requirements on PKE (δ_{ik}, $\sigma_{\delta_{\mathrm{ik}}}$ and FFP-NG) are defined in a way such reasoning about their concrete estimates can safely involve computational assumptions, as they represent settings in which the attacker does not possess the secret key. On the other hand, [MX23] stressed that these notions nonetheless introduce new analysis tasks for designers who want to argue security of their concrete scheme. We therefore addresses the following open question:

> **Can we reconcile the proof for explicitly rejecting KEMs in [HHM22] with the more established correctness notion (worst-case correctness)?**

Result of this note. We will show that the red-dotted part of Fig. 1 can be replaced with a picture only involving the worst-case correctness parameter δ_{wc}, see Fig. 2.

To achieve this, the only part requiring a change will be how we reason that attackers cannot distinguish oDecaps from its simulation, to which end we would like to simply resort to the original COR notion.

The only hurdle is that COR, as analysed so far, isn't a seamless fit: the simulation of oDecaps in [HHM22] involves a slightly more complicated variant of the QROM, called eQROM. In the eQROM, the attacker gets an additional interface that essentially inverts certain encryptions, Since the search bound for COR was only known in the plain QROM that does not provide this additional interface, we need to reprove the bound in the eQROM.

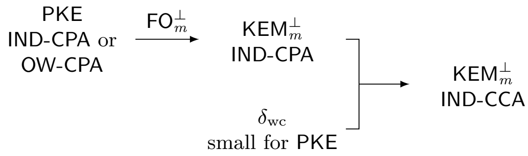


Fig. 2. Analogue of Fig. 1 with the alternative decryption failure analysis developed in this note.

TL;DR for scheme designers. Theorem 1 (on page 4) provides concrete bounds for the IND-CCA security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. Ignoring constant factors up to 10 and an additive term related to the size of the message space (denoted "$\lesssim$"), our bound is roughly of the following form:

$$\epsilon_{\mathsf{IND\text{-}CCA\text{-}KEM}} \lesssim \sqrt{(d + q_{\mathsf{D}}) \cdot \epsilon_{\mathsf{IND\text{-}CPA}}} + (q + q_{\mathsf{D}} + 1)^2 \cdot \delta_{\mathrm{wc}} + q_{\mathsf{D}}(q + q_{\mathsf{D}}) \cdot 2^{-\gamma/2} \ .$$

The bound requires to upper bound the following values:

$\epsilon_{\mathsf{IND\text{-}CPA}}$	IND-CPA advantage against PKE
q	number of issued random oracles queries
q_{D}	number of decryption queries
d	random oracle query depth (can be bounded trivially by q)
$2^{-\gamma/2}$	maximal probability that encryption hits a specific ciphertext (see Def. 5 on page 11)
δ_{wc}	worst-case correctness of PKE as defined in [HHK17] (see Def. 2 on page 4): probability that decrypting $\mathsf{Enc}(m)$ doesn't yield m for the worst message m, averaged over KG

Assuming an attacker makes far less online queries than hash queries (so $q_{\mathsf{D}} \ll q$), trivially bounding $d < q$, and dropping constant factors up to 4, we can further simplify the bound to

$$\epsilon_{\mathsf{IND\text{-}CCA\text{-}KEM}} \lesssim \sqrt{q \cdot \epsilon_{\mathsf{IND\text{-}CPA}}} + q^2 \cdot \delta_{\mathrm{wc}} + q_{\mathsf{D}} \cdot q \cdot 2^{-\gamma/2} \ .$$

2 Preliminaries.

After establishing basic notation, we recall several correctness-related notions for public-key encryption schemes that were introduced in [HHK17] and [HHM22]. (For convenience, we also recall more standard definitions for public-key encryption and key encapsulation algorithms in Appendix A.)

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.

FINDING FAILING PLAINTEXTS (FFP). Following [HHM22], we formalise the finding of failing plaintexts as the winning condition of the FFP game below. In the FFP-CCA game, the adversary is given the public key and access to a decryption oracle, outputs a message m and wins if $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m$. We are only concerned with the game run against $\mathsf{PKE}^{\mathsf{G}}$, i.e., a public-key encryption scheme that stems from derandomising some public-key encryption scheme PKE as sketched in the introduction and formalised in Fig. 9 on page 12).

Definition 1 **(FFP-CCA of $\mathsf{PKE}^{\mathsf{G}}$).** *Let* $\mathsf{PKE}^{\mathsf{G}} = (\mathsf{KG}, \mathsf{Enc}^{\mathsf{G}}, \mathsf{Dec}^{\mathsf{G}})$ *be the modified public-key encryption scheme stemming from derandomising some public-key encryption scheme* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$. *We define the* FFP-CCA *game for* $\mathsf{PKE}^{\mathsf{G}}$ *as in Fig. 3, and the* FFP-CCA *advantage function of an adversary* $\mathcal{A}$ *against* $\mathsf{PKE}^{\mathsf{G}}$ *as*

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

Game $\mathsf{FFP\text{-}CCA}_{\mathsf{PKE}^{\mathsf{G}}}$	**Oracle** $\text{oDecrypt}(c \neq c^*)$
$(pk, sk) \leftarrow \mathsf{KG}$	$m' := \mathsf{Dec}(sk, c)$
$m \leftarrow \mathcal{A}^{\text{oDecrypt}, \mathsf{eCO.RO}, \mathsf{eCO.Ext}}(pk)$	**if** $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	**return** $\perp$
$m' := \mathsf{Dec}(sk, c)$	**else**
if $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$	**return** m'
$m' := \perp$	
return $[\![m' \neq m]\!]$	

Fig. 3. Game $\mathsf{FFP\text{-}CCA}$ for derandomised scheme $\mathsf{PKE}^{\mathsf{G}}$ with G modelled as an extractable compressed oracle eCO with oracle interface $\mathsf{eCO.RO}$ and extractor interface $\mathsf{eCO.Ext}$. We note that the difference between game $\mathsf{FFP\text{-}CCA}$ and COR-eQROM is that in $\mathsf{FFP\text{-}CCA}$, $\mathcal{A}$ has the decryption oracle oDecrypt, while possessing the full secret key in COR-eQROM.

We now recall the definition of worst-case-correctness introduced in [HHK17], there called δ-correctness.

Definition 2 **(δ_{wc}-worst-case-correctness).** *We say that a public-key encryption scheme* PKE *is* δ_{wc}*-worst-case-correct if*

$$\mathbf{E}[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(sk, c) \neq m \mid c \leftarrow \mathsf{Enc}(pk, m)]] \leq \delta_{\text{wc}} \ ,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{KG}$ *and the probability is over the randomness of* Enc.

In particular, δ_{wc}-worst-case correctness means that even (possibly unbounded) adversaries with access to the secret key will succeed in triggering decryption failure with probability at most δ_{wc}. This property was formalised in [HHK17] as the winning condition of a correctness game COR, in which the adversary gets the full key pair, outputs a message, and wins if the message exhibits decryption failure. The difference between $\mathsf{FFP\text{-}CCA}$ and COR is having the full key pair (COR) vs. having access to a decryption oracle ($\mathsf{FFP\text{-}CCA}$).

Like [HHK17], we need to analyse the respective term for $\mathsf{PKE}^{\mathsf{G}}$, i.e., a public-key encryption scheme resulting from derandomising some public-key encryption scheme PKE. Since derandomisation happens via a random oracle G, [HHK17] introduced a QROM analogue of game COR, called $\mathsf{COR\text{-}QRO}$, in which the attacker has quantum access to G.

Unlike in [HHK17], however, the proof structure imposed by [HHM22] makes it necessary to analyse the correctness game in an extension of the QROM, called eQROM. (For convenience, we briefly recapture the eQROM in Appendix D.) With Definition 3 below, we hence extend the $\mathsf{COR\text{-}QRO}$ definition from [HHK17] to the extended QROM. In the extended QROM, G is modelled as an extractable compressed oracle eCO that provides the oracle's interface (called $\mathsf{eCO.RO}$) and, additionally, an extractor interface $\mathsf{eCO.Ext}$ that is defined relative to some function f. We will need to refer to the unitary operator facilitating queries to $\mathsf{eCO.RO}$, which we denote by O. Intuitively, the extractor interface $\mathsf{eCO.Ext}$, when queried on some target value t, produces preimages x such that $f(x, \mathsf{G}(x)) = t$, assuming that such an x was already noticeable in previous oracle queries. Like [HHM22], we will work with $f := \mathsf{Enc}$. This means that $\mathsf{eCO.Ext}$, when queried on a ciphertext c, will produce a plaintext m for c such that m and its random oracle value r have the property that $\mathsf{Enc}(m; r) = c$.

Definition 3. *We define the extended QROM correctness game* COR-eQROM$_{\mathsf{PKE}^{\mathsf{G}}}$ *for* $\mathsf{PKE}^{\mathsf{G}}$ *in Fig. 4, and the advantage of an adversary* $\mathcal{A}$ *against* $\mathsf{PKE}^{\mathsf{G}}$ *as*

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

3 Our main result

We start by stating our main result that relates $\mathsf{IND\text{-}CCA}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ to $\mathsf{IND\text{-}CPA}$ security, δ_{wc}-worst-case correctness and γ-spreadness of PKE.

Theorem 1 **(PKE $\mathsf{IND\text{-}CPA}$ secure and δ_{wc}-worst-case correct $\Rightarrow$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CCA}$).** *Let* PKE *be a (randomized)* PKE *scheme that is* γ*-spread and* δ_{wc}*-worst-case-correct, with message space of size* $|\mathcal{M}|$*. Let* $\mathcal{A}$

GAME $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{PKE}^{\mathsf{G}}}$

$(pk, sk) \leftarrow \mathsf{KG}$
$m \leftarrow \mathcal{A}^{\mathsf{eCO.RO},\mathsf{eCO.Ext}}(sk, pk)$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$
$m' := \mathsf{Dec}^{\mathsf{G}}(sk, c)$
if $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$
  $m' := \perp$
return $[\![m' \neq m]\!]$

Fig. 4. Correctness game $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ for $\mathsf{PKE}^{\mathsf{G}}$ with G modelled as an extractable compressed oracle eCO with oracle interface $\mathsf{eCO.RO}$ and additional extractor interface $\mathsf{eCO.Ext}$ that, intuitively, produces plaintexts for queried ciphertexts. Lines 03-05 are defined relative to the random oracle G that is modelled as an extractable QRO eCO, we stuck with writing G for the sake of simplicity. (Formally, G represents oracle interface $\mathsf{eCO.RO}$.)

be an $\mathsf{IND}\text{-}\mathsf{CCA}\text{-}\mathsf{KEM}$ *adversary (in the QROM) against* $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$*, 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 *be the query depth of the combined queries to* G *and* H*. Then there exists an* $\mathsf{IND}\text{-}\mathsf{CPA}$ *adversary* $\mathcal{B}$ *against* PKE *such that*

$$\mathrm{Adv}^{\mathsf{IND}\text{-}\mathsf{CCA}\text{-}\mathsf{KEM}}_{\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]}(\mathcal{A}) \leq \mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} + 10(q+1)^2\delta_{\mathrm{wc}} + \varepsilon_\gamma \ ,$$

with

$$\mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} = 4 \cdot \sqrt{(d + q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{PKE}}(\mathcal{B})} + \frac{8\,(q + q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} \ ,$$

and the additive spreadness term ε_γ *being defined by*

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

The running time of $\mathcal{B}$ *is bounded by* $\mathrm{Time}(\mathcal{B}) \leq \mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q + q_{\mathsf{D}}, q_{\mathsf{D}}) + O(q_{\mathsf{D}})$ *and* $\mathcal{B}$ *requires quantum memory bounded by* $\mathrm{QMem}(\mathcal{B}) \leq \mathrm{QMem}(\mathcal{A}) + \mathrm{QMem}(\mathsf{eCO}, q + q_{\mathsf{D}}, q_{\mathsf{D}})$*, where* $\mathrm{Time}(\mathsf{eCO}, q, q_E)$*, and* $\mathrm{QMem}(\mathsf{eCO}, q, q_E)$*, denote the time, and quantum memory, necessary to simulate the extractable QROM for* q *many queries to* $\mathsf{eCO.RO}$ *and* q_E *many queries to* $\mathsf{eCO.Ext}$*.*

Proof. We begin by stating an implicit result of [HHM22] as Theorem 2 (below) that relates $\mathsf{IND}\text{-}\mathsf{CCA}$ security of $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ to $\mathsf{IND}\text{-}\mathsf{CPA}$ security of PKE and $\mathsf{FFP}\text{-}\mathsf{CCA}$ security of $\mathsf{PKE}^{\mathsf{G}}$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$.

Theorem 1 is obtained by bounding the $\mathsf{FFP}\text{-}\mathsf{CCA}$ term in Eq. (1) of Theorem 2 in terms of δ_{wc}, which we will do in Section 4: Theorem 3 states that the $\mathsf{FFP}\text{-}\mathsf{CCA}$ term can be bounded by $10(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}} + 1)^2\delta_{\mathrm{wc}}$. Here, we identified $\mathcal{C}$'s number of $\mathsf{eCO.RO}$ queries in Theorem 3 with $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ as indicated by Theorem 2.

For completeness, we show that Theorem 2 indeed follows straightforwardly from the results in [HHM22] in Appendix C. ☐

Theorem 2. *[*$\mathsf{PKE}^{\mathsf{G}}$ $\mathsf{FFP}\text{-}\mathsf{CCA}$ *and* PKE $\mathsf{IND}\text{-}\mathsf{CPA}$ *secure* $\Rightarrow$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND}\text{-}\mathsf{CCA}$*] Let* PKE *be a (randomized)* PKE *scheme that is* γ*-spread, and let* $\mathcal{A}$ *be an* $\mathsf{IND}\text{-}\mathsf{CCA}\text{-}\mathsf{KEM}$ *adversary (in the QROM) against* $\mathsf{FO}_m^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$*, 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 *be the query depth of the combined queries to* G *and* H*. Then there exist an* $\mathsf{IND}\text{-}\mathsf{CPA}$ *adversary* $\mathcal{B}$ *against* PKE *and an* $\mathrm{eQROM}_{\mathsf{Enc}}$ $\mathsf{FFP}\text{-}\mathsf{CPA}$ *adversary* $\mathcal{C}$ *against* $\mathsf{PKE}^{\mathsf{G}}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND}\text{-}\mathsf{CCA}\text{-}\mathsf{KEM}}_{\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]}(\mathcal{A}) \leq \mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} + \mathrm{Adv}^{\mathsf{FFP}\text{-}\mathsf{CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) + \varepsilon_\gamma \ , \tag{1}$$

with

$$\mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} = 4 \cdot \sqrt{(d + q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND}\text{-}\mathsf{CPA}}_{\mathsf{PKE}}(\mathcal{B})} + \frac{8\,(q + q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} \ ,$$

and the additive spreadness term ε_γ *being defined by*

$$\varepsilon_\gamma = 12 q_{\mathsf{D}}(q_{\mathsf{G}} + 4q_{\mathsf{D}}) 2^{-\gamma/2} \ .$$

The running time of $\mathcal{B}$ is bounded by $\mathrm{Time}(\mathcal{B}) \leq \mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q + q_{\mathsf{D}}, q_{\mathsf{D}}) + O(q_{\mathsf{D}})$ *and $\mathcal{B}$ requires quantum memory bounded by* $\mathrm{QMem}(\mathcal{B}) \leq \mathrm{QMem}(\mathcal{A}) + \mathrm{QMem}(\mathsf{eCO}, q + q_{\mathsf{D}}, q_{\mathsf{D}})$*, where* $\mathrm{Time}/\mathrm{QMem}(\mathsf{eCO}, q, q_E)$ *denotes the time/quantum memory necessary to simulate the extractable QROM for q many queries to* eCO.RO *and q_E many queries to* eCO.Ext*. $\mathcal{C}$ makes $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ queries to* eCO.RO*.*

4 Bounding FFP-CCA in the eQROM via worst-case correctness

We now give the alternative analysis of FFP-CCA in the $\mathrm{eQROM}_{\mathsf{Enc}}$ that allows us to replace the FFP-CCA term in Theorem 2 by $10(q+1)^2\delta_{\mathrm{wc}}$.

Theorem 3 **(PKE δ_{wc}-worst-case-correct $\Rightarrow$ $\mathsf{PKE}^{\mathsf{G}}$ FFP-CCA).** *Let* PKE *be a (randomized)* PKE *scheme that is δ_{wc}-worst-case-correct, and let $\mathcal{C}$ be an* FFP-CCA *adversary $\mathcal{C}$ against $\mathsf{PKE}^{\mathsf{G}}$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$, issuing at most q_D decryption queries and q many queries to its extQROM oracle interface* eCO.RO*. Then*

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) \leq 10(q + q_D + 1)^2 \delta_{\mathrm{wc}} \quad . \tag{2}$$

Proof. The proof proceeds in two steps.

1. Use FFP-CCA adversary $\mathcal{C}$ to construct a COR-eQROM adversary $\hat{\mathcal{C}}$ against $\mathsf{PKE}^{\mathsf{G}}$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$ that has the same advantage as $\mathcal{C}$ and makes $\hat{q} := q + q_D$ many queries to eCO.RO.
2. Prove that any such $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{PKE}^{\mathsf{G}},\mathsf{Enc}}$ adversary $\mathcal{D}$, making $\hat{q}$ many queries to the oracle interface eCO.RO that models G, has advantage at most $10(\hat{q}+1)^2\delta_{\mathrm{wc}}$.

$$\begin{array}{c}\delta_{\mathrm{wc}}\\ \text{small for PKE}\end{array} \xrightarrow{\text{Step 2}} \begin{array}{c}\mathsf{PKE}^{\mathsf{G}}\\ \text{COR-eQROM}\\ \text{in the}\\ \text{eQROM}\end{array} \xrightarrow{\text{Step 1}} \begin{array}{c}\mathsf{PKE}^{\mathsf{G}}\\ \text{FFP-CCA in}\\ \text{the eQROM}\end{array}$$

For step 1, we note that COR-eQROM adversaries get the full key pair (sk, pk) (as specified by game COR-eQROM, see Fig. 4) and can hence simulate the decryption oracle on their own. In more detail, we construct COR-eQROM adversary $\hat{\mathcal{C}}$ against $\mathsf{PKE}^{\mathsf{G}}$ as follows: $\hat{\mathcal{C}}$ runs $\mathcal{C}$, forwards all eCO.RO/eCO.Ext queries to its own extractable oracle interfaces, and simulates $\mathcal{C}$'s Dec oracle using the secret key. To perform the re-encryption check during the simulation of Dec, $\hat{\mathcal{C}}$ has to make one additional query to eCO.RO per Dec call. Once $\mathcal{C}$ finishes, $\hat{\mathcal{C}}$ simply forwards $\mathcal{C}$'s output m. $\hat{\mathcal{C}}$ perfectly simulates the FFP-CCA game for $\mathcal{C}$ and wins iff $\mathcal{C}$ wins, hence

$$\mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) \leq \mathrm{Adv}^{\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}}_{\mathsf{PKE}^{\mathsf{G}}}(\hat{\mathcal{C}}) \quad .$$

To begin with step 2 (analysing the $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ advantage), we first slightly simplify the winning condition of the $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ game for $\mathsf{PKE}^{\mathsf{G}}$: We introduce game 1 that only differs from game 0, the original $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ game for $\mathsf{PKE}^{\mathsf{G}}$, by dropping the re-encryption check from the winning condition. It is easy to verify that the $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ advantage is exactly the advantage against game 1:

- The winning condition in game 1 implies the winning condition in game 0.
- To show the other direction, we notice that $\mathcal{A}$ wins game 0 by producing a message m such that either its encryption fails to decrypt (which is the winning condition in game 1) or such that the re-encryption check fails. But if the the re-encryption check fails, then $\mathsf{Dec}(sk, c)$ cannot yield m (and $\mathcal{A}$ again wins in game 1).

$$\mathrm{Adv}^{\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}}_{\mathsf{PKE}^{\mathsf{G}}}(\hat{\mathcal{C}}) = \Pr[\hat{\mathcal{C}} \text{ wins in } G_1] \quad .$$

We proceed by analysing the $\mathsf{COR}\text{-}\mathrm{eQROM}_{\mathsf{Enc}}$ advantage with this simplified winning condition. More concretely, we would like to bound the maximal advantage in game 1 of any adversary that makes at most $\hat{q}$ many queries. To that end, we fix the key pair and define a predicate $P_{\mathrm{fail},\mathsf{PKE}^{\mathsf{G}}}$ by

$$P_{\mathrm{fail},\mathsf{PKE}^{\mathsf{G}}}(m) \Leftrightarrow \mathsf{Dec}_{sk}(\mathsf{Enc}^{\mathsf{G}}_{pk}(m)) \neq m.$$

GAMES 0 - 1

$(pk, sk) \leftarrow \mathsf{KG}$
$m \leftarrow \mathcal{A}^{\mathsf{eCO.RO},\mathsf{eCO.Ext}}(sk, pk)$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$
$m' := \mathsf{Dec}^{\mathsf{G}}(sk, c)$
if $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$ //Game G_0
  $m' := \bot$ //Game G_0
return $[\![m' \neq m]\!]$

Fig. 5. Game G_0, the correctness game $\mathsf{COR}\text{-eQROM}_{\mathsf{Enc}}$ for $\mathsf{PKE}^{\mathsf{G}}$, and Game G_1 with slightly simplified winning condition.

We use the predicate to rewrite the winning condition in game 1:

$$\Pr[\hat{\mathcal{C}} \text{ wins in } G_1] = \mathbf{E}_{\mathsf{KG}} \Pr_{m \leftarrow \hat{\mathcal{C}}^{\mathsf{eCO.RO},\mathsf{eCO.Ext}}(sk,pk)}[P_{\text{fail},\mathsf{PKE}^{\mathsf{G}}}(m)] \ .$$

We will now bound the right-hand side, i.e., the probability that $\hat{\mathcal{C}}$ returns a message satisfying the predicate, for any fixed key pair. To that end, we give a helper Lemma 1 below which relates $\hat{\mathcal{C}}$'s success probability to a sum of square roots of probabilities ("amplitudes"). The sum is taken over all random oracle queries (including an implicit one to check the predicate). In the sum, the k–th summand intuitively represents the following: Consider the oracle query database D for eCO to contain up to k many entries, meaning up to k many queries to $\mathsf{eCO.RO}$ were made so far, without satisfying the predicate. We consider the maximal probability that picking a random output value u for some oracle input value m leads to (m, u) satisfying the predicate. (In the lemma's notation, $\mathrm{Found}(D[m \mapsto u])$, where we define Found like in Lemma 1, using our predicate $P_{\text{fail},\mathsf{PKE}^{\mathsf{G}}}$ on the message space.) The maximum is taken over all possible oracle input values m and all query databases D such that the predicate was not yet satisfied ($\neg\mathrm{Found}(D)$).

We continue by giving a formal argument. Note that the predicate $P_{\text{fail},\mathsf{PKE}^{\mathsf{G}}}$ can be computed using a single query to G, we can therefore identify variable $q_{\mathcal{P}}$ in Lemma 1 with 1. Applying Lemma 1, we thus obtain

$$\begin{aligned}
\sqrt{\Pr_{m \leftarrow \hat{\mathcal{C}}^{\mathsf{eCO.RO},\mathsf{eCO.Ext}}(sk,pk)}[P_{\text{fail},\mathsf{PKE}^{\mathsf{G}}}(m)]} &\leq \sum_{k=1}^{\hat{q}+1} \max_{\substack{m,D: \\ |D| \leq k \\ \neg\mathrm{Found}(D)}} \sqrt{10 \Pr_{u \leftarrow \mathcal{Y}}[\mathrm{Found}(D[m \mapsto u])]} \\
&\leq (\hat{q}+1) \max_{\substack{m,D: \\ |D| \leq q+1 \\ \neg\mathrm{Found}(D)}} \sqrt{10 \Pr_{u \leftarrow \mathcal{Y}}[\mathrm{Found}(D[m \mapsto u])]}
\end{aligned}$$

where the second inequality holds because any database with $\ell < q+1$ entries fulfilling the predicate can be completed to a database with $q+1$ entries still fulfilling the predicate.

To translate the summands back into terms concerning decryption failure, we note the following: If $\neg\mathrm{Found}(D)$, but $\mathrm{Found}(D[x \mapsto u])$, then it must be specifically the entry (x, u) that satisfies the predicate. Thus, assuming the database D before was in a state such that $\neg\mathrm{Found}(D)$, we find

$$\mathrm{Found}(D[x \mapsto u]) \Leftrightarrow \mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(x; u)) \neq x \ .$$

Using this fact and squaring both sides of the above inequality yields

$$\Pr_{m \leftarrow \hat{\mathcal{C}}^{\mathsf{eCO.RO},\mathsf{eCO.Ext}}(sk,pk)}[P_{\text{fail},\mathsf{PKE}^{\mathsf{G}}}(m)] \leq 10(\hat{q}+1)^2 \max_m \Pr_{u \leftarrow \mathcal{Y}}[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m; u)) \neq x]$$

for any fixed key pair (sk, pk). Taking the expectation over KG hence yields

$$\begin{aligned}
\Pr[\hat{\mathcal{C}} \text{ wins in } G_1] &\leq \mathbf{E}_{\mathsf{KG}} 10(\hat{q}+1)^2 \max_m \Pr_{u \leftarrow \mathcal{Y}}[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m; u)) \neq x] \\
&= 10(\hat{q}+1)^2 \delta_{\text{wc}}.
\end{aligned}$$

$\square$

In the above proof, we used the following

Lemma 1 (Variant of Lemma 1 in [AMHJ+23]). *Let $\mathsf{G} : \mathcal{X} \to \mathcal{Y}$ be a random oracle and let $\mathcal{P}^{\mathsf{G}}$ be a predicate on some set $\mathcal{Z}$ that can be computed using at most $q_{\mathcal{P}}$ classical queries to G. Let further $\mathcal{A}^{\mathsf{G}}$ be an algorithm in the* eQRO$_f$ *(for an arbitrary f), making at most q quantum queries to* eCO.RO *and outputing $z \in \mathcal{Z}$. Then*

$$\sqrt{\Pr_{z \leftarrow \mathcal{A}^{\mathsf{G}}}[P(z)]} \leq \sum_{k=1}^{q+q_{\mathcal{P}}} \max_{\substack{x,D:\\ |D| \leq k\\ \neg\mathrm{Found}(D)}} \sqrt{10 \Pr_{u \leftarrow \mathcal{Y}}[\mathrm{Found}_{\mathcal{P}}(D[x \mapsto u])]} \tag{3}$$

where Found$_{\mathcal{P}}$ *is the database property*

$$\mathrm{Found}_{\mathcal{P}} = (\exists z \in \mathcal{Z} : \mathcal{P}^D(z)) \tag{4}$$

and $\mathcal{P}^D$ is the algorithm that computes $\mathcal{P}$ but makes queries to D instead of G, and if any query returns $\perp$, $\mathcal{P}^D$ ouputs 'false'.

Before we give a proof of Lemma 1, we need to prepare some ingredients. In particular, the proof uses the concept of *transition capacities* from [CFHL21], we now recall the required notation from that paper.

A *database property* P is a predicate on the set of partial functions with the same input and output space as G. Overloading notation, we also denote by P the projector acting on a compressed oracle database register with support spanned by the computational basis states corresponding to partial functions fulfilling P. For any database property P we define the database property P_i such that f fulfils P_i iff it fulfils P and is defined on at most i inputs.

We now define the quantum transition capacity, following [CFHL21]. The quantum transition capacity $[\![P \to P']\!]$ is the quantum analogue of the maximum probability that a query transcript has a property P' after an input together with a freshly lazy-sampled output has been added to the transcript, given that the transcript has property P before. In addition, we define a q-query variant that considers q adaptively chosen inputs.

Definition 4 (Quantum transition capacity). *Let P, P' be two database properties. Then, the* quantum transition capacity *is defined as*

$$[\![P \xrightarrow{q} P']\!] := \sup_{U_1,\ldots,U_{q-1}} \|P' O\, U_{q-1}\, O \cdots O\, U_1\, O\, P\| .$$

where the supremum is over all adversary register sizes and all unitaries $U_1, \ldots, U_{q-1}$ acting on the adversary's registers. We write

$$[\![P \to P']\!] := [\![P \xrightarrow{1} P']\!] = \|P'OP\|$$

To bound the power of the eQROM$_f$ for search tasks, we strengthen the model slightly by having the interface eCO.Ext apply the purified version (the *Stinespring dilation*) of $\mathcal{M}_t$ on input t, and return the (quantum) output register. This generalization is not strictly necessary for our proof, but is convenient as it allows us to model an algorithm with query access to eQROM$_f$ as unitary. Concretely, the purified measurement is the isometry

$$V_{TD \to TDO} = \sum_t |t\rangle\langle t|_T \otimes V^{(t)}_{D \to DO}, \text{ with}$$
$$V^{(t)}_{D \to DO} = \sum_{x \in \{0,1\}^m} \Sigma^{t,x}_D \otimes |x\rangle_O .$$

Let us call this model the eQROM$^*_f$ and the strengthened extraction interfacte eCO.Ext*. Any algorithm in the eQROM$_f$ can be simulated in the eQROM$^*_f$ by submitting any eCO.Ext queries to eCO.Ext*, measuring the output and returning the result.

In the following we prove that for query bounds for oracle search problems (like, e.g., preimage search, collision search) proven using the compressed oracle framework, the same bound holds for algorithms with eQROM$^*_f$-access, irrespective of the number of queries made to the interface eCO.Ext*. On a high level, this is due to the fact that the operator that facilitates a query to eCO.Ext* and the projector checking the database property commute. The argument is similar to the one made in Appendix B of [AMHJ+23]. We define the *decorated* transition capacity as

$$[\![P \to P']\!]_V = \|P'VOP\|.$$

We have the following

Lemma 2. *Let V_{DE} be a controlled unitary with control register the database register D, and acting on an arbitrary additional register E. Then*
$$[\![P \to P']\!]_V = [\![P \to P']\!].$$

Proof. As V is a controlled unitary with control register D, and P' is an operator that is diagonal in the computational basis, we have $V_{DE}P'_D = P'_D V_{DE}$. We thus get

$$[\![P \to P']\!]_V = \|P'VOP\| = \|VP'OP\| = \|P'OP\| = [\![P \to P']\!].$$

Here, the second equality follows because V and P' commute, and the third equality is due to the unitary invariance of the operator norm. ◻

This lemma can be used to show that the framework for query bounds developed in [CFHL21] works essentially unchanged for the decorated transition capacity $[\![P \to P']\!]_V$ with a controlled unitary V as in Lemma 2 as well.[3]

Now, any algorithm $\mathcal{A}$ in the eQROM$^*_f$ proceeds without loss of generality by applying the unitary

$$U_{\mathcal{A}} = U_q O U_{q-1} O \dots O U_0$$

to a quantum register initialized in the all-0 state, where the U_i have the form

$$U_i = U_{i,\ell} V U_{i,\ell-1} V \dots V U_{i,0},$$

where the unitaries $U_{i,j}$ do not act on the compressed oracle database.

Using the prepared ingredients, we can conclude that Lemma 1 from [AMHJ+23] holds in the eQROM$^*_f$, with a bound depending on the number of $\mathsf{eCO.RO}$ queries only:

Proof (of Lemma 1). The proof is identical to the proof of Lemma 1 in [AMHJ+23], with one difference: If we denote the adversary's unitary (we can purify/Stinespring-dilate any adversary for this mathematical argument) between the ith and the $(i+1)$st query to $\mathsf{eCO.RO}$ by U_i, we obtain the decorated transition capacity $[\![\neg\text{Found} \wedge (|D| \le k-1) \to \text{Found}]\!]_{U_i}$ instead of the 'non-decorated' capacity $[\![\neg\text{Found} \wedge (|D| \le k-1) \to \text{Found}]\!]$. (Note that U_i includes any $\mathsf{eCO.Ext}$ queries made by the adversary between the ith and the $(i+1)$st query to $\mathsf{eCO.RO}$, which are controlled unitaries with control register D.) Due to Lemma 2, however, this does not make any difference and the proof proceeds as in [AMHJ+23]. ◻

References

AMHJ+23. Carlos Aguilar-Melchor, Andreas Hülsing, David Joseph, Christian Majenz, Eyal Ronen, and Dongze Yue. Sdith in the qrom. Cryptology ePrint Archive, Paper 2023/756, 2023. https://eprint.iacr.org/2023/756.

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.

CFHL21. Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part II*, volume 12697 of *Lecture Notes in Computer Science*, pages 598–629, Zagreb, Croatia, October 17–21, 2021. 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.

[3] Here we have only defined and characterized the decorated transition capacity as needed for analyses that don't distinguish sequential and parallel queries, which suffices for our purposes.

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.

DGJ+19. Jan-Pieter DAnvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. Decryption failure attacks on ind-cca secure lattice-based schemes. In *Public-Key Cryptography PKC 2019*, volume 11443 of *Lecture Notes in Computer Science*, pages 565–598. Springer, 2019.

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.

FKK+22. Michael Fahr, Hunter Kippen, Andrew Kwong, Thinh Dang, Jacob Lichtinger, Dana Dachman-Soled, Daniel Genkin, Alexander Nelson, Ray Perlner, Arkady Yerukhimovich, and Daniel Apon. When frodo flips: End-to-end key recovery on frodokem via rowhammer. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 979993, New York, NY, USA, 2022. Association for Computing Machinery.

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.

HHM22. Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Failing gracefully: Decryption failures and the fujisaki-okamoto transform. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022, Part IV*, volume 13794 of *Lecture Notes in Computer Science*, pages 414–443, Taipei, Taiwan, December 5–9, 2022. 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.

MX23. Varun Maram and Keita Xagawa. Post-quantum anonymity of Kyber. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023: 26th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 13940 of *Lecture Notes in Computer Science*, pages 3–35, Atlanta, GA, USA, May 7–10, 2023. Springer, Heidelberg, Germany.

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 Definitions for Public-Key Encryption (PKE) and Key Encapsulation Mechanisms (KEMs)

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 Appendix D.)

A.1 Definitions for PKE

For convenience, we start by recalling the formal definition of γ-spreadness.

Definition 5 (γ-spreadness). *We say that* PKE *is γ-spread iff for all key pairs* $(pk, sk) \in \operatorname{supp}(\mathsf{KG})$ *and all messages* $m \in \mathcal{M}$ *it holds that*

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

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 (OW-CPA) and Indistinguishability under Chosen-Plaintext Attacks (IND-CPA).

Definition 6 (OW-CPA, IND-CPA). *Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with message space* $\mathcal{M}$. *We define the* OW-CPA *game as in Fig. 6 and the* OW-CPA *advantage function of an adversary* $\mathcal{A}$ *against* PKE *as*

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

Furthermore, we define the 'left-or-right' version of IND-CPA *by defining games* $\mathsf{IND\text{-}CPA}_b$*, where* $b \in \{0, 1\}$ *(also in Fig. 6), and the* IND-CPA *advantage function of an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *against* PKE *(where* $\mathcal{A}_2$ *has binary output) as*

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

Game OW-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. 6. Games OW-CPA and $\mathsf{IND\text{-}CPA}_b$ for PKE.

A.2 Standard notions for KEM

We now recall Indistinguishability under Chosen-Plaintext Attacks (IND-CPA) and under Chosen-Ciphertext Attacks (IND-CCA).

Definition 7 (IND-CPA, IND-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* IND-ATK-KEM *games as in Fig. 7, where*

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

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

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

Game IND-ATK-KEM

$(pk, sk) \leftarrow \mathsf{KG}$

$b \leftarrow_\$ \{0,1\}$

$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$

$K_1^* \leftarrow_\$ \mathcal{K}$

$b' \leftarrow \mathcal{A}^{\mathsf{O}_{\mathsf{ATK}}}(pk, c^*, K_b^*)$

return $[\![b' = b]\!]$

oDecaps$(c \neq c^*)$

$K := \mathsf{Decaps}(sk, c)$

return K

Fig. 7. Game IND-ATK-KEM for KEM, where $\mathsf{ATK} \in \{\mathsf{CPA}, \mathsf{CCA}\}$ and $\mathsf{O}_{\mathsf{ATK}}$ is defined in Definition 7.

B 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. 8. $\mathsf{FO}_m^\perp$ 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. 9.

$\mathsf{Encaps}(pk)$

01 $m \leftarrow_\$ \mathcal{M}$

02 $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$

03 $K := \mathsf{H}(m)$

04 **return** (K, c)

$\mathsf{Decaps}(sk, c)$

05 $m' := \mathsf{Dec}(sk, c)$

06 **if** $m' = \perp$ **or** $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$

07 **return** $\perp$

08 **else**

09 **return** $K := \mathsf{H}(m')$

Fig. 8. Key encapsulation mechanism $\mathsf{KEM}_m^\perp = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$, obtained from $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ by setting $\mathsf{KEM}_m^\perp := \mathsf{FO}_m^\perp[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$.

$\mathsf{Enc}^{\mathsf{G}}(pk)$

$m \leftarrow_\$ \mathcal{M}$

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

return c

$\mathsf{Dec}^{\mathsf{G}}(sk, c)$

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

if $m' = \perp$ **or** $c \neq \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$

return $\perp$

else

return m'

Fig. 9. 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 $\mathsf{Dec}^{\mathsf{G}}$.

C Obtaining Theorem 2 from [HHM22]

For the reader's convenience, we begin by restating Theorem 2.

Theorem 2. *[$\mathsf{PKE}^{\mathsf{G}}$ $\mathsf{FFP\text{-}CCA}$ and PKE $\mathsf{IND\text{-}CPA}$ secure $\Rightarrow$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CCA}$] Let PKE be a (randomized) PKE scheme that is γ-spread, and let $\mathcal{A}$ be an $\mathsf{IND\text{-}CCA\text{-}KEM}$ adversary (in the QROM) against $\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, 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 be the query depth of the combined queries to G and H. Then there exist an $\mathsf{IND\text{-}CPA}$ adversary $\mathcal{B}$ against PKE and an $\mathrm{eQROM}_{\mathsf{Enc}}$ $\mathsf{FFP\text{-}CPA}$ adversary $\mathcal{C}$ against $\mathsf{PKE}^{\mathsf{G}}$ such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA\text{-}KEM}}_{\mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]}(\mathcal{A}) \leq \mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} + \mathrm{Adv}^{\mathsf{FFP\text{-}CCA}}_{\mathsf{PKE}^{\mathsf{G}}}(\mathcal{C}) + \varepsilon_\gamma \ , \tag{1}$$

with

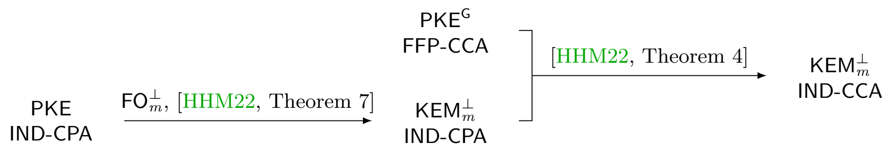
$$\mathrm{Adv}_{\mathsf{PKE},\mathcal{B}} = 4 \cdot \sqrt{(d+q_{\mathsf{D}}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B})} + \frac{8\,(q+q_{\mathsf{D}})}{\sqrt{|\mathcal{M}|}} \ ,$$

and the additive spreadness term ε_γ being defined by

$$\varepsilon_\gamma = 12 q_{\mathsf{D}}(q_{\mathsf{G}} + 4q_{\mathsf{D}})2^{-\gamma/2} \ .$$

The running time of $\mathcal{B}$ is bounded by $\mathrm{Time}(\mathcal{B}) \leq \mathrm{Time}(A) + \mathrm{Time}(\mathsf{eCO}, q+q_{\mathsf{D}}, q_{\mathsf{D}}) + O(q_{\mathsf{D}})$ and $\mathcal{B}$ requires quantum memory bounded by $\mathrm{QMem}(\mathcal{B}) \leq \mathrm{QMem}(\mathcal{A}) + \mathrm{QMem}(\mathsf{eCO}, q+q_{\mathsf{D}}, q_{\mathsf{D}})$, where $\mathrm{Time}/\mathrm{QMem}(\mathsf{eCO}, q, q_E)$ denotes the time/quantum memory necessary to simulate the extractable QROM for q many queries to $\mathsf{eCO.RO}$ and q_E many queries to $\mathsf{eCO.Ext}$. $\mathcal{C}$ makes $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ queries to $\mathsf{eCO.RO}$.

The corollary is obtained in a straightforward manner by combining Theorems 4 and 7 from [HHM22] as indicated in the figure below.



We begin by repeating [HHM22, Theorem 3].

Theorem 4 **($\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CPA}$ and $\mathsf{PKE}^{\mathsf{G}}$ $\mathsf{FFP\text{-}CCA}$ $\overset{\mathrm{eQROM}_{\mathsf{Enc}}}{\Rightarrow}$ $\mathsf{FO}_m^{\perp}[\mathsf{PKE}]$ $\mathsf{IND\text{-}CCA}$).** *Let PKE be a (randomized) PKE that is γ-spread, and $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$. Let $\mathcal{A}$ be an $\mathsf{IND\text{-}CCA\text{-}KEM}$-adversary (in the QROM) against $\mathsf{KEM}_m^{\perp}$, 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 $\mathsf{IND\text{-}CPA\text{-}KEM}$ adversary $\tilde{\mathcal{A}}$ and an $\mathsf{FFP\text{-}CCA}$ adversary $\mathcal{B}$ against $\mathsf{PKE}^{\mathsf{G}}$, both in the $\mathrm{eQROM}_{\mathsf{Enc}}$, such that*

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

The adversary $\tilde{\mathcal{A}}$ makes $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$ queries to $\mathsf{eCO.RO}$ with a combined depth of $d + q_{\mathsf{D}}$, and q_{D} queries to $\mathsf{eCO.Ext}$. Here, $\mathsf{eCO.RO}$ simulates $\mathsf{G} \times \mathsf{H}$. Adversary $\mathcal{C}$ makes q_{D} many queries to ODECRYPT *and $\mathsf{eCO.Ext}$ and q_{G} queries to $\mathsf{eCO.RO}$. Neither $\tilde{\mathcal{A}}$ nor $\mathcal{C}$ query $\mathsf{eCO.Ext}$ on the challenge ciphertext. The running times of the adversaries $\tilde{\mathcal{A}}$ and $\mathcal{C}$ are bounded by $\mathrm{Time}(\tilde{\mathcal{A}}), \mathrm{Time}(\mathcal{C}) \leq \mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{D}})$.*

We proceed by repeating [HHM22, Theorem 7]. The bound in Theorem 2 is obtained by plugging [HHM22, Theorem 7] into [HHM22, Theorem 3] and identifying $\tilde{q}$ with $q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{D}}$, $\tilde{d}$ with $d + q_{\mathsf{D}}$, and $\tilde{q_E}$ with q_{D}.

Theorem 5. *Let $\tilde{\mathcal{A}}$ be an $\mathsf{IND\text{-}CPA\text{-}KEM}$ adversary against $\mathsf{KEM}_m^{\perp} := \mathsf{FO}_m^{\perp}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$ in the $\mathrm{eQROM}_{\mathsf{Enc}}$, issuing $\tilde{q}$ many queries to $\mathsf{eCO.RO}$ in total, with a query depth of $\tilde{d}$, and $\tilde{q_E}$ many queries to $\mathsf{eCO.Ext}$, where none of them is with its challenge ciphertext. Then there exists an $\mathsf{IND\text{-}CPA}$ adversary $\mathcal{B}$ against PKE such that*

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

The running time and quantum memory footprint of $\mathcal{B}$ satisfy $\mathrm{Time}(\mathcal{B}) = \mathrm{Time}(\tilde{\mathcal{A}}) + \mathrm{Time}(\mathsf{eCO}, \tilde{q}, \tilde{q_E})$ and $\mathrm{QMem}(\mathcal{B}) = \mathrm{QMem}(\tilde{\mathcal{A}}) + \mathrm{QMem}(\mathsf{eCO}, \tilde{q}, \tilde{q_E})$.

D Compressed oracles and extraction

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 $\mathsf{eCO.Ext}_f$, defined relative to a function $f : X \times Y \to T$. Informally, $\mathsf{eCO.Ext}_f$ 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 $f(x, y) = t$. $\mathsf{eCO.Ext}_f$ 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 $\mathsf{eCO.Ext}$ instead of $\mathsf{eCO.Ext}_f$.

In general, $\mathsf{eCO.Ext}_f$ 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 $\mathsf{eCO.Ext}$ 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 defined by
$$O\,|x\rangle_X = |x\rangle_X \otimes \left(F_{D_x} \mathsf{CNOT}^{\otimes n}_{D_x:Y} F_{D_x}\right), \tag{5}$$
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.

We now make our description of the extraction interface $\mathsf{eCO.Ext}$ 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.
$$\Sigma^{t,x} = \bigotimes_{x'<x} \bar{\Pi}^{t,x'}_{D'_x} \otimes \Pi^{t,x}_{D_x}, \quad \text{with} \quad \Pi^{t,x} = \sum_{\substack{y\in\{0,1\}^n:\\ f(x,y)=t}} |y\rangle\langle y| \tag{6}$$
and $\bar{\Pi} = \mathbb{1} - \Pi$. The remaining projector corresponds to the case where no register contains such a y, i.e.
$$\Sigma^{t,\emptyset} = \bigotimes_{x'\in\{0,1\}^m} \bar{\Pi}^{t,x'}_{D'_x}. \tag{7}$$
As an example, say we model a random oracle H as such an eQRO_f. Using $f(x, y) := [\![\mathsf{H}(x) = y]\!]$, $\mathcal{M}^1$ 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. $\mathsf{eCO.Ext}$ 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], given below.These characterisations are in terms of the quantity

$$\begin{aligned} \varGamma(f) = \max_t \varGamma_{R_{f,t}}, \text{ with} \\ R_{f,t}(x,y) :\Leftrightarrow f(x,y) = t \text{ and} \\ \varGamma_R := \max_x |\{y \mid R(x,y)\}|. \end{aligned} \tag{8}$$

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 $\varGamma(f) = 2^{-\gamma}|\mathcal{R}|$ if PKE is γ-spread. In this case, $\mathsf{eCO.Ext}(c)$ 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.