Chasing QuOCCAs in a Quantum World: Type-2 Oracles for CCA-Secure PKE

Barbara Jiabao Benedikt[1], Tommaso Gagliardoni[2,3], and Patrick Struck[4]

[1] Technische Universität Darmstadt, Germany
[2] Horizen Labs, USA
[3] Lucumo Security, Switzerland
[4] Universität Konstanz, Germany

Abstract. In the context of PKE schemes, Gagliardoni et al. proposed at PQCrypto 2021 a qIND-qCPA security notion (a superposition-based analogue of the classical IND-CPA security notion), by using the theory of so-called *type-2 unitary operators*. On one hand, this notion is very natural, closely mirrors the classical intuition, and can be handled without relying on complex techniques such as Zhandry's compressed oracles. On the other hand, it is restricted to a certain class of PKE schemes (so-called *isometric*). Moreover, it is not immediately clear how to extend the definition to chosen-ciphertext attack (CCA) scenarios, mainly due to the possibility of decryption failures – something that is entailed by most quantum-resistant PKE schemes.

In this work, we use the theory of type-2 operators to extend superposition-based security notions to *any* PKE schemes, in the CPA and CCA setting, *without* Zhandry's compressed oracle technique. We start first by showing that a trivial extension of Gagliardoni et al.'s techniques to the general case is not possible, even for the CCA1 case, by identifying barriers preventing the realization of a 'natural' type-2 decryption operator. Then we define a subclass of PKE schemes (which we call *'strongly decryptable'*), for which it is easy to circumvent the aforementioned barriers and to define superposition-based CCA1 and CCA2 notions.

Further, we introduce a novel transformation (that we call *'purification'*) which applies to any PKE scheme, producing a 'quasi-PKE' scheme, for which it is possible to define properties that mimic the security notions defined for strongly decryptable schemes; we can thus 'unload' the security definitions for an arbitrary PKE scheme on its purification. Finally, we show implications and separations between our security notions, as well as constructions.

Keywords: Quantum Security Notions · Type-2 Operators · Minimal Oracles · Public Key Encryption · qIND-qCPA· qIND-qCCA1· qIND-qCCA2

* Barbara Jiabao Benedikt was supported by the National Research Center for Applied Cybersecurity ATHENE. Patrick Struck acknowledges funding by the Hector Foundation II.

1 Introduction

Quantum-resistant, or post-quantum, cryptography (PQC) emerged as a response to the threat that future large-scale quantum computers will pose to the security of cryptographic schemes. Quantum algorithms like Shor's [Sho94] are known to have the potential of breaking most public key cryptography in use today such as RSA and ECC. Initially, research in PQC was focused on finding different hardness assumptions conjectured to hold even against quantum computers, e.g., lattice problems such as the shortest vector problem [Reg05].

Eventually, PQC research shifted towards explicitly considering reductions from quantum adversaries when proving security. In some cases, this becomes challenging, as in the case of proofs that rely on random oracles. Classically, random oracles are meant to model hash functions in real-world protocols. But since the description of a hash function is public, one should assume that a quantum adversary should be able to implement the circuit of the function as a *quantum circuit*, therefore being able to evaluate the (classical) function over a superposition of different inputs, or even over mixed states. This, in certain cases, might introduce new attack vectors. Thus, in the case of quantum adversaries, random oracle proofs need to be lifted to the *quantum random oracle model (QROM)* [BDF+11]. While initially very challenging due to the lack of proof techniques, many classical results have been restored in the QROM, such as Fiat-Shamir [DFMS19] and Fujisaki-Okamoto [HKSU20, DFMS22].

The QROM models a setting, where adversaries have local quantum computing power, while honest users still only have classical computing power. In the literature, this model is commonly referred to as Q1 (or QS1, or 'post-quantum'), while the non-quantum model is then called Q0 (or QS0, or 'classical'). The Q1 model is arguably relevant in a near future, where quantum computers are available to powerful adversaries, but not widespread and accessible to normal users, so that most cryptography is still executed on classical computing devices. However, in recent years, quantum security research also considered the stronger Q2 setting (or QS2, or 'superposition-based'), where, in addition to local quantum computing power, adversaries are granted quantum access to certain oracles that are normally meant to be only classical, such as signature or decryption oracles. For encryption and signature schemes, this research area was first considered by Boneh and Zhandry [BZ13]. Subsequently, many works provided new results in the Q2 model [GHS16, GYZ17, AMRS20, CETU21, GKS21, KS22, CEV22, AM24].

When working in the Q2 model, the challenge is not merely proving security, but defining proper security notions that lift well-understood properties from the Q0 or Q1 model (say, IND-CPA for encryption) to the Q2 model. For example, in their initial work, Boneh and Zhandry showed an inherent limitation when lifting security notions for encryption schemes like IND-CPA and IND-CCA to the Q2 model: A straightforward adaptation of the challenge phase to the quantum setting results in an unachievable security notion. Later works showed how to circumvent this problem using mainly two different approaches.

Mossayebi and Schack [MS16] provided an IND-CPA security notion for symmetric encryption that relies on a "real-or-permuted" game, where the adversary

receives the encryption of either the original message, or of a random permutation of it; Chevalier et al. [CEV22] later extended on that approach and provided also IND-CCA security notions in the Q2 model, handling cheating adversaries – those that try to ask for a decryption of the challenge ciphertext – using the compressed oracle technique by Zhandry [Zha19].

Gagliardoni et al. [GHS16] used a very different approach to lift IND-CPA security for symmetric encryption to the Q2 model. At the heart of their work are so-called *type-2* operators, or minimal oracles as initially introduced in [KKVB02]. These are quantum oracles that process plaintext and ciphertext 'on the fly', without relying on ancilla registers. This approach was later extended to the case of public key encryption (PKE) schemes in [GKS21].

In this work, we follow the same line of research using type-2 operators, extending the results for PKE scheme in the Q2 model. The main limitations of [GKS21] are that (1) their notions only apply to a restricted class of PKE schemes, called *isometric*; and (2) even in that restriction, security is only defined up to chosen-plaintext attacks, while there are no security notions proposed for the generally stronger chosen-ciphertext attack scenarios. This is a clear gap in the Q2 model which simply does not exist in the Q0 and Q1 model, where defining chosen-ciphertext attack security is somewhat trivial.[5] Therefore, the question we ask is:

Can we lift the qIND-qCPA *security notion for PKE schemes from [GKS21] to a* qIND-qCCA *security notion, possibly to a larger set of schemes?*

1.1 Our Contribution

In this work, we answer the aforementioned question in the affirmative by studying qIND-qCCA security for PKE schemes using the theory of type-2 operators.

Traditionally, in the quantum setting, the biggest obstacle in defining CCA security lies in handling replay detection mechanisms in the post-challenge decryption queries, e.g., the *adaptive*, or *CCA2* case, while the *non-adaptive*, or *CCA1* case is relatively straightforward. Classically, IND-CCA2 is defined by imposing that the adversary is not allowed to query the decryption oracle on the challenge ciphertext (which would otherwise result in trivial break of security). But enforcing this condition when the query is quantum is usually tricky, due to no-cloning and the destructive nature of quantum measurements. This problem has been solved in [AGM18], in the context of symmetric-key encryption in the *fully quantum (Q3)* scenario[6], by using a quantum replay detection technique based on a 'real-vs-ideal' approach. In the Q2 case, looking at [CEV22], defining qIND-qCCA1 for the 'real-or-permuted' setting is fairly straightforward, while defining qIND-qCCA2 requires the (much more involved) compressed oracle technique [Zha19].

[5] Bellare et al. [BHK15] showed subtle details in defining security but defining chosen-ciphertext security is not part of current research anymore.

[6] The Q3 model is not concerned with quantum security of classical cryptography but deals with quantum cryptography itself.

In our 'type-2 oracle' case, instead, we are faced with a more immediate problem. Surprisingly, it turns out that the main obstacle is lifting Q2 security even just to qIND-qCCA1, i.e., lunchtime attacks [NY90] where the adversary has access to a decryption oracle only *before* the challenge. This is mainly due to two reasons: on one hand, we must assume that, in general, information on the randomness used during encryption of a certain message is lost, and cannot be recovered from the ciphertext alone. On the other hand, many quantum-resistant PKE schemes suffer from a small rate of *decryption failures*. Both issues, albeit not problematic in the real-world implementation of PKE schemes, are a barrier to defining a properly invertible decryption quantum oracle. As a first contribution, in this work we describe what a 'natural' form of this type-2 decryption oracle should look like, and prove that it is impossible to build.

To work around this problem, we start first by considering a restricted class of PKE schemes (which we call *strongly decryptable*) which have a "very good decryption property"; namely, for every honestly generated ciphertext, it is always possible to (deterministically and efficiently) decrypt that ciphertext *and* at the same time recover the *unique* randomness which generated that ciphertext. This property, in particular, implies that the mapping from message-randomness pair to ciphertext is injective. We show that strongly decryptable schemes exist (e.g., the RSA-OAEP construction [FOPS01]), and characterize relations and separations to other properties introduced in [GKS21], such as *isometry, recoverability, and perfect correctness*. Next, we show how the strong decryptability property allows to define suitable type-2 encryption and decryption oracles, which in turn allow us to define the qIND-qCCA1 security notion, thereby breaking the main barrier in advancing the state of the art.

The downside of this approach is that it is limited to strongly decryptable PKE scheme, which are somewhat rare (e.g., lattice-based encryption schemes typically lose information in the process of encryption which prevent the receiver to recover the randomness during decryption). So, as a next step, we introduce a transformation that we call *purification* of a PKE scheme. Given *any* PKE scheme, its *purification* (or *purified scheme*) is a variant which is *not* a PKE scheme itself according to the definition, but which 'approximates' the behavior of the original scheme, except that no information is lost during encryption. Using the framework previously built for strongly decryptable schemes, we then define ad-hoc security notions for purified schemes, which mimic the notion that we would like to express *if the scheme were a PKE*. We can then unload the security definition for a PKE on the corresponding security notion for its purification. This allows us to extend qIND-qCPA and qIND-qCCA1 security notions to *any* PKE scheme. We also consider the extension to qIND-qCCA2, for which we rely on the Q3 model technique of "replay detection" from [AGM18]; this is made possible by the use of type-2 operators, which 'bridge' classical constructions to quantum ones. Again, considering the purification of a PKE scheme allows us to extend qIND-qCCA2 to *any* PKE scheme.

Finally, we show separating results and we also prove that the standard public key / symmetric key hybrid construction achieves our security notions under standard properties of the underlying composing schemes.

1.2 On the Motivations for Q2 Security

There are various reasons for considering Q2 security over the more conservative Q1. At the very minimum, Q2 security notions are stronger, and allow to aim for potentially higher security. On a theoretical level, many Q2 notions allow to bridge in a natural way security notions and techniques in the classical (Q0 and Q1) world, and others that are natively meant to be used in a scenario where every party, not just the adversary, has access to quantum computing and communication resources (the so called Q3, or QS3, or 'fully quantum' model). On the practical level, Q2 security models scenarios where an adversary can 'trick' a classical hardware in exhibiting (at least partial) quantum behavior.

A common misconception is that such scenarios would imply that the adversary is able to 'turn a classical device into a quantum computer'; this is not the case. What the Q2 notion captures is the possibility that the adversary, having physical control of a classical computing device, manages to expose the device to non-standard physical conditions which, in turn, cause some unexpected quantum behavior (cf. the *'frozen smart card'* mind experiment in [GHS16]). So, it is more akin to considering a quantum version of *side-channel attacks*: the scenario is not *'the smart card chip turns into a quantum computer'*, but rather *'pins nr. 3 and 7 of the smart card chip show a partial quantum correlation with some probability'*, which is a much weaker condition, and yet all Q0 and Q1 security models collapse under such condition. Since we do not have, as of today, a meaningful model of which types of quantum computation should be possible or not depending on the hardware and the adversary's power, we opt for the conservative approach of granting quantum oracle access to the adversary.

Besides that, though, there are other reasons as to why the Q2 model is an interesting model to study. Lifting security notions to the Q2 model can sometimes reveal details that are somewhat "hidden" when restricting to the Q1 model. An example can be seen when looking at the qIND-qCPA security notions by Gagliardoni et al. [GKS21]: To define their security notion, they relied on a property called *recoverability* which seems to be widespread in public key encryption schemes. This property was exploited to attack several submissions to the NIST PQC standardization process, exploiting bad domain separations of random oracles [BDG20].[7]

Finally, a well established theory in the Q2 model can lead to a more natural extension into the Q3 model, that is, quantum cryptography.

[7] Note though that [BDG20] developed the recoverability property concurrently and independently of [GKS21].

1.3 On the Motivation for Type-2 Operators

When considering existing work about quantum security notions, one can find a variety of design choices. This is already true for CPA security: The initial notion by Boneh and Zhandry [BZ13] show that a notion using type-1 operators in the left-or-right sense leads to an unachievable security notion – their workaround is a notion that uses type-1 operators but only in the learning phase, keeping the challenge phase classical. Later, Gagliardoni et al. [GHS16] developed a security notion using type-2 operators in the left-or-right approach while Mossayebi and Schack [MS16] developed a notion using type-1 operators in the real-or-permuted sense. Both of those approaches successfully avoid the problem identified by Boneh and Zhandry. Carstens et al. [CETU21] investigated the relations between 72 different IND-CPA security notions (for symmetric encryption) and proved (non-)equivalences as well as (non-)implications. Among these security notions are the three aforementioned notions. A core results of [CETU21] is that the notions from [GHS16] and [MS16] are incomparable, i.e., they do not imply each other. This leaves us with the open question: which notions are the "best" ones – type-1 or type-2?

The answer is not straightforward. If we were looking at security models for *classical schemes implemented on a quantum computer* one could argue that, since traditionally these schemes have been described in literature mostly as type-1 operators, type-1 security would be more natural. However, this is an oversimplification, because it does not take into account the fact that, when actually *realizing* a quantum circuit, the actual implementation could be optimized for the classical function at hand, and not follow the didactical blueprint. For example, consider a classical function that computes the boolean NOT on a bit b. One could represent it as a type-1 operator as $|b\rangle|0\rangle \mapsto |b\rangle|1-b\rangle$. However, in this case, and depending on the concrete physical architecture of the computer, it would probably be more efficient to implement it as a 1-qubit Pauli X operation as $|b\rangle \mapsto X|b\rangle$, i.e. a type-2. Moreover, when looking at *quantum* side-channel attacks on *classical* hardware, whether the most realistic model follows the type-1 or type-2 paradigm certainly depends on the *physical realization* of the classical circuit in exam. For example, one could argue that circuits that implement operations on separate 'source' and 'target' registers (e.g. a `"add eax, ebx"` instruction in X86 Assembly) would be better modeled as type-1 operators, while operations on single registers (e.g. a `"inc eax"` instruction) are more in line with the type-2 approach. Considering that even a simple function could be composed of many instructions of both types, even just for the CPA case it is not immediately clear whether the "right" model is type-1 or type-2, that is why it is important to study both.

Furthermore, [GKS21] describes other advantages that type-2 CPA security notions have in comparison to type-1 notions like the ones in [CEV22]. These advantages basically boil down to the fact that type-2 notions more closely mirror the established cryptographic framework both for the classical case (where they allow to switch smoothly from *Left-or-Right* to *Real-or-Random* notions), and for the fully quantum case (where they allow to easily prove equivalent notions

of *semantic security*, or to lift the security of a classical scheme to that of a quantum scheme that uses the classical one as building block).

If we shift our focus from CPA to CCA security, one could assume that we have a similar situation between *type-1* and *type-2* qIND-qCCA. Clearly, both approaches are important to investigate in order to find a good notion. Chevalier et al. [CEV22] used type-1 operators and the real-or-permuted approach as [MS16], and succeeded to define qIND-qCCA notions for PKE schemes. In this work we will define an alternative qIND-qCCA notion which relies on type-2 operators (as [GHS16]). Our qIND-qCCA2 notion follows the 'real-vs-ideal' approach from [AGM18] and does not require the compressed oracle technique [Zha19] as [CEV22] does.

2 Preliminaries

In this section, we briefly recall basic quantum notation and results from [GKS21] which we will use in the rest of our paper. We refer to [NC02] for more details.

2.1 Quantum Computation and Notation

A quantum system A is represented by a complex Hilbert space denoted by $\mathfrak{H}_A$. Pure states are (equivalence classes of) elements in a Hilbert space $\mathfrak{H}$, whereas mixed states are elements of a density matrix operator space $\mathfrak{D}(\mathfrak{H})$ over $\mathfrak{H}$, where $\mathfrak{D}$ is Hermitian positive semi-definite linear operator of trace 1. We denote with $\mathfrak{M}$ the measurement in the canonical computational basis. For pure states we use the ket notation, e.g., $|\varphi\rangle$, while we use lowercase Greek letters, e.g., ρ, for mixed states. With abuse of notation, when the separation is clear from the context, we denote a bi- (or multi-)partite state over two (or more) quantum systems as multiple concatenated lowercase Greek letters; for example, for registers A and B, we might denote $\rho\sigma \in \mathfrak{D}(\mathfrak{H}_A \otimes \mathfrak{H}_B)$, where $\mathrm{Tr}_A(\rho\sigma) = \sigma \in \mathfrak{D}(\mathfrak{H}_B)$ (and symmetrically on the other register, where Tr_A denotes *partial trace*). Operations from a quantum system A to a quantum system B are performed by unitary operators or super-operators, e.g., $U : \mathfrak{H}_A \to \mathfrak{H}_B$ for operations on pure states and $U : \mathfrak{D}(\mathfrak{H}_A) \to \mathfrak{D}(\mathfrak{H}_B)$ for mixed states.

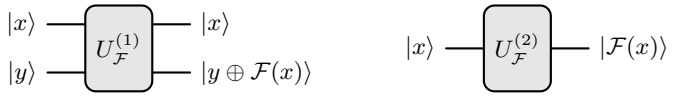

Fig. 1: Type-1 (left) and type-2 (right) operators for function $\mathcal{F}$.

We distinguish two ways to compute a classical function on a superposition of inputs: either using a *type-1* or a *type-2* operator. This distinction was first introduced by Kashefi et al. [KKVB02]:

Definition 1 *Let $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ be a function, then we call the unitaries*

$$U_{\mathcal{F}}^{(1)} : |x, y\rangle \mapsto |x, y \oplus \mathcal{F}(x)\rangle\,,$$
$$U_{\mathcal{F}}^{(2)} : |x\rangle \mapsto |\mathcal{F}(x)\rangle\,,$$

type-1 resp. type-2 operator, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ (cf. Figure 1).

A classical function $\mathcal{F}$ can always be implemented efficiently as a type-1 operator whenever $\mathcal{F}$ is efficient [NC02], whereas we cannot always realize a type-2 implementation of a function $\mathcal{F}$. This is, in general, only possible if $\mathcal{F}$ is (efficiently) *invertible*.

2.2 PKE Schemes in the Q2 World

In the following, we consider (families of) sets for representing different objects, such as messages, keys, randomness, and ciphertexts. When looking at CCA notions, we also have to handle the case of *decryption failures* through a special *rejection symbol* $\perp$ that lies outside the message space.

Definition 2 (PKE) *A* public key encryption (PKE) *scheme is a tuple of three efficient algorithms $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$.*

$\mathtt{KGen} : \mathbb{N} \times \mathcal{R} \to \mathcal{PK} \times \mathcal{SK}$ *takes a security parameter λ and randomness r, and return a keypair $(\mathsf{pk}, \mathsf{sk})$.*

$\mathtt{Enc} : \mathcal{PK} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ *maps a public key pk, message m, and randomness r to a ciphertext c.*

$\mathtt{Dec} : \mathcal{SK} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$ *maps a secret key sk and ciphertext c to a message m (or rejects with $\perp$).*

We call Π perfectly correct*, if for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, $m \in \mathcal{M}$ and $r \in \mathcal{R}$ it holds that $\mathtt{Dec}_{\mathsf{sk}}(\mathtt{Enc}_{\mathsf{pk}}(m; r)) = m$.*

We call Π recoverable *(cf. [GKS21]), if there exists an efficient algorithm $\mathtt{Rec} : \mathcal{PK} \times \mathcal{R} \times \mathcal{C} \to \mathcal{M}$ such that for any $\mathsf{pk} \in \mathcal{PK}$, $r \in \mathcal{R}$, $m \in \mathcal{M}$ it holds that $\mathtt{Rec}_{\mathsf{pk}}(r, \mathtt{Enc}_{\mathsf{pk}}(m; r)) = m$.*

Note, that a PKE consists of efficient and classical algorithms which can be represented as a classical circuit. Same is true for $\mathtt{Rec}$, if it exists. Since we consider the Q2 setting, we want to represent these algorithms as quantum circuits *or operators*. Obviously, as the mentioned algorithms are efficient, the following type-1 operators exist:

Definition 3 (Type-1 Operators for PKE**)** *Let $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a recoverable* PKE *scheme and let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. The type-1 encryption, decryption, and recovery operator are the unitaries defined as follows*

$$U_{\mathtt{Enc}_{\mathsf{pk}}}^{(1)} : |m, r, y\rangle \mapsto |m, r, y \oplus \mathtt{Enc}_{\mathsf{pk}}(m; r)\rangle\,,$$
$$U_{\mathtt{Dec}_{\mathsf{sk}}}^{(1)} : |c, y\rangle \mapsto |c, y \oplus \mathtt{Dec}_{\mathsf{sk}}(c)\rangle\,,$$
$$U_{\mathtt{Rec}_{\mathsf{pk}}}^{(1)} : |r, c, y\rangle \mapsto |r, c, y \oplus \mathtt{Rec}_{\mathsf{pk}}(r, c)\rangle\,.$$

Following [GKS21], we classify PKE schemes according to whether it is possible to define a unitary that behaves as a type-2 encryption operator. These schemes we call *(efficiently) isometric* PKE scheme, depending on whether the corresponding unitary can be efficiently realized.

Definition 4 ((Efficiently) Isometric Schemes) *Let Π be a* PKE *scheme and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. *We call Π* ***isometric****, if there exists a unitary U such that for any $m \in \mathcal{M}$ and $r \in \mathcal{R}$:*

$$\Pr\left[\mathfrak{M}\big(U\,|m, 0, \ldots, 0, r\rangle\,\big) \mapsto (\mathtt{Enc}_{\mathsf{pk}}(m; r), r)\right] = 1.$$

Furthermore, we say Π is ***efficiently isometric****, if the scheme is isometric and the corresponding unitary U can be efficiently realized.*

Thus, given an efficiently isometric PKE scheme the corresponding type-2 encryption operator exists:

Definition 5 (Type-2 Encryption for PKE) *Let Π be an efficiently isometric PKE scheme, and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. *A type-2 encryption operator for Π is an efficiently computable unitary in the family defined by*

$$U^{(2)}_{(\mathtt{Enc},\mathsf{pk},\mathsf{sk})} : |m, 0^\ell, r\rangle \mapsto |\mathtt{Enc}_{\mathsf{pk}}(m; r), r\rangle\,,$$

where $\ell \in \mathbb{N}$ is chosen such that the message $m||0^\ell$ has the same bit length as the encrypted message $\mathtt{Enc}_{\mathsf{pk}}(m; r)$.

Gagliardoni et al. [GKS21] proved that, if a PKE scheme is perfectly correct or recoverable, there exists a canonical efficient implementation of $U^{(2)}_{\mathtt{Enc}}$:

Theorem 6 (Thm. 7 and Thm. 11 in [GKS21]) *Let Π be PKE scheme. If Π is perfectly correct or recoverable, then it is efficiently isometric.*

2.3 qIND-qCPA Security of PKE Schemes

Type-2 encryption operators allow us to define the qIND-qCPA security notion, since the challenger can use these operators to answer encryption queries of the adversary in superposition without creating extra entanglement.

Oracle $O^{(2)}_{\mathtt{Enc}}(\varphi)$ on input $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$

1: $\chi \otimes |r\rangle\langle r| \coloneqq U^{(2)}_{\mathtt{Enc}}(\mu \otimes |0 \ldots 0\rangle\langle 0 \ldots 0| \otimes |r\rangle\langle r|)$, where $r \leftarrow^{\$} \mathcal{R}$
2: Trace out $|r\rangle\langle r|$
3: **return** χ

Fig. 2: Type-2 encryption oracle as in [GKS21], where $U^{(2)}_{\mathtt{Enc}}$ is a type-2 encryption operator. Note that $|r\rangle\langle r|$ can be traced out since it is not entangled with χ.

More in detail (cf. Fig. 2): when the challenger receives a (quantum) plaintext μ from the adversary, she samples a classical randomness r and feeds r together

with the received μ (and zero ancillas) into the encryption operator $U^{(2)}_{\mathtt{Enc}}$. Accordingly, the output of this operator will be: a register containing the (quantum) ciphertext χ, and a copy register containing the same classical randomness r that was used as input, which is thus not entangled and can be decoupled from the ciphertext register and traced out. This allows the challenger to only send the ciphertext register χ to the adversary.

Having a type-2 encryption oracle defined, it is easy to define qIND-qCPA security for PKE schemes. Once an adversary decided the challenge messages, she sends them over to the challenger who encrypts one of the messages using her type-2 encryption oracle (cf. [GKS21]).

Definition 7 (qIND-qCPA Security for Isometric PKE (LoR version)) *Let Π be an efficiently isometric* PKE *scheme. Then Π has quantum ciphertext indistinguishability under quantum chosen plaintext attack, or is* qIND-qCPA *secure, iff the advantage of any QPT adversary $\mathcal{A}$ to win the experiment* $\text{qIND-qCPA}_{LoR}(\Pi, \mathcal{A}, \lambda)$ *(defined as in Fig. 3) is negligible, i.e.,*

$$\mathbf{Adv}^{\text{qIND-qCPA}_{LoR}}_{\Pi}(\mathcal{A}) \coloneqq \left|\Pr[\text{qIND-qCPA}_{LoR}(\Pi, \mathcal{A}, \lambda) \to \mathsf{win}] - \frac{1}{2}\right| \leq \text{negl}(\lambda).$$

$\text{qIND-qCPA}_{LoR}(\Pi, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda)$

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, implements $O^{(2)}_{\mathtt{Enc}}$, and flips a coin $b \leftarrow^{\$} \{0,1\}$
2: $\mathcal{A}_1^{O^{(2)}_{\mathtt{Enc}}}(\mathsf{pk}, \lambda)$ prepares a memory state σ, and sends to $\mathfrak{C}$ challenge states μ_0 and μ_1
3: $\mathfrak{C}$ samples fresh (classical) randomness r, traces out μ_{1-b}, sends μ_b to $O^{(2)}_{\mathtt{Enc}}$, and sends the response χ to $\mathcal{A}_2$
4: $\mathcal{A}_2$ receives states χ and σ
5: $\mathcal{A}_2^{O^{(2)}_{\mathtt{Enc}}}$ outputs a bit b'
6: **if** $b = b'$, **then return** win; **else return** reject

Fig. 3: The qIND-qCPA game (LoR, or Left-or-Right, version) for an efficiently isometric PKE scheme Π, where $O^{(2)}_{\mathtt{Enc}}$ is defined as in Fig. 2.

RSA-OAEP. Throughout this work, we sometimes refer to RSA-OAEP as it exhibits relevant properties for our work. RSA-OAEP combines the RSA trapdoor permutation with the *optimal asymmetric encryption padding* [BR95], which works as follows. Let π_{pk} be the RSA trapdoor permutation, and G, H random oracles. To encrypt a message m using randomness r, compute $s \leftarrow m \oplus G(r)$ and $t \leftarrow H(s) \oplus r$, and set the ciphertext to be $c \leftarrow \pi_{\mathsf{pk}}(s \parallel t)$. Decryption works in a straightforward way: Apply the inverse permutation to c, obtaining $(s \parallel t) \leftarrow \pi^{-1}_{\mathsf{sk}}(c)$, from which first r is recovered as $r \leftarrow H(s) \oplus t$ and, subsequently, the message m as $m \leftarrow s \oplus G(r)$, which is then returned. To achieve CCA security, the message is padded with a number of zeros which are verified during decryption.

3 Obstacles in Defining Reversible Decryption

In PKE schemes, type-2 encryption operators can be defined mainly for the reason that the challenger maintains control of the randomness used to answer the encryption queries of the adversary. The situation changes completely if we think about how the challenger could answer to decryption queries instead. In that case, we are facing several problems.

First, and most importantly, there is the issue of decryption failures, which are an inherent possibility in many real-world PKE schemes. As long as such failures only happen with low probability, this should not be an obstacle for defining suitable qIND-qCCA security notions. But from a formal point of view, this is a problem if we want to leverage type-2 operators for the security definition, because we need to specify the behavior of these operators (which should be, in an intuitive sense, "always reversible") when these failures happen. We will ignore this problem at first, by initially considering only schemes where such failures never happen. We will address the general case in Section 5.

Second, under the assumption of no decryption failures, we need to implement the encryption in a reversible way in order to build a type-2 decryption operator. But since a single plaintext maps to many different ciphertexts (because of CPA security), we need to recover the randomness that was used to generate the ciphertext if we want to disambiguate the mapping and have any hope of fully reverting the operator. Unfortunately, it is not obvious how we can provide this randomness: both challenger and adversary do not necessarily know which randomness was used for the encryption of a given ciphertext state during a decryption query. In some cases, we may assume that the randomness may be somehow recovered from the ciphertext alone, but in general we should assume that the original randomness was either lost or withheld by the encrypting party.

Third, during decryption, we cannot hope to keep a copy of the ciphertext register as output of the operator: except for the case of purely classical queries, this would introduce entanglement between that register and the returned plaintext. If we withheld that register, this would be equivalent to measuring it from the adversary's point of view (cf. [GHS16]), which would result in a disturbed returned plaintext (because of the entanglement), also potentially allowing the adversary to distinguish ciphertexts and breaking security. The idea of type-2 operators, instead, is that the input (including the ciphertext in our case) should be completely "uncomputed" and transformed into the output (the plaintext), without copying any information.

Moreover, it would be useful to have a blueprint to obtain type-2 operators from more elementary building blocks that we know to be always efficiently implementable, i.e., type-1 operators. This is even more important for those operators, like our decryption case, which aim at translating quantumly a procedure rooted in classical hardware: in this case, building such type-2 translation using "purely quantum" gates such as Hadamards would be more difficult to justify.

Thus, the preferred type-2 decryption operator for PKE would ideally be built from elementary type-1 operators, and output neither ciphertext nor randomness

register, due to the reasons just described. Unfortunately, this is impossible to achieve, as the next section shows.

3.1 Impossibility of Practical Type-2 Decryption for PKE

As we have seen, type-1 operators can be composed and used as building blocks to realize a certain type-2 operator. It is possible to know which type-2 operators can be built this way and which cannot by only looking at the input and output registers of the available type-1 operators used as components.

To explain the reason, we look at a simple example: let U_1 be a type-1 operator with input domains $I_1, \ldots, I_k$ and output domains $O_1, \ldots, O_l$, and U^* a operator built from U_1. Any reasonable operator U^* takes the U_1's input variables and l-many output registers initialized with $|0\rangle$ and contains one or more copies of U_1. This leads, if the number of copies is even, to a "back calculation" of the output registers $|o_1\rangle, \ldots, |o_l\rangle$ to $|0\rangle$. However, in any case I_1 is a subset of the output variables of U^* (cf. Fig. 4).

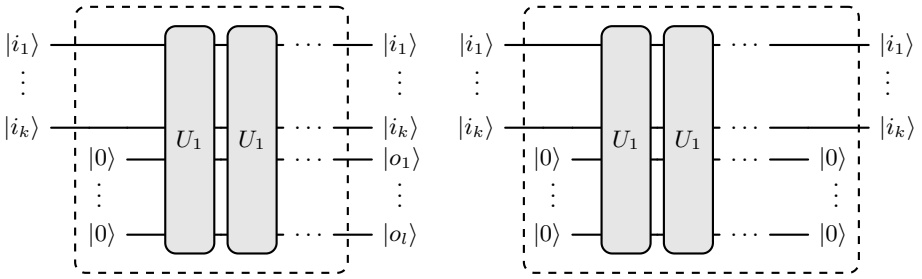

Fig. 4: General form of operators U^* built from U_1. The operators have either $I_1 \cup O_1$ as output registers (left), or only I_1 (right).

The picture looks similarly for all other operators U^* built from type-1 operators: there exists a "last" type-1 operator in U^*, for which we cannot uncompute the input registers anymore, since it is the "last" operation. This idea is reflected in the following theorem, whose proof can be found in Appendix D.1:

Theorem 8. *Let $U_1, \ldots, U_n$ be unitaries which operate on a shared set of variables $\{x_1, \ldots, x_m\}$ (with domains $X_1 \times \cdots \times X_m$), where each application of a unitary U_i corresponds to a (type-1)* conversion *of the input variables $I_i \coloneqq \{i_1^{(i)}, \ldots\}$ to the output variables $O_i \coloneqq \{o_1^{(i)}, \ldots\}$. For instance, if U_1 converts x_1 into x_2 and U_2 converts x_2 into x_1, then $U_2U_1(x_1) = x_1$ for any $x_1 \in X_1$. Let U^* be an operator which is constructed by a* conversion-respecting *composition (i.e., the composition will not feed a variable $x_j \neq x_i$ to a unitary as x_i) from the unitaries $U_1, \ldots, U_n$, and let $S^* \subset \{x_1, \ldots, x_m\}$ be the set of output variables of U^*. Then there exists a unitary U_i – with input variables I_i – , such that $I_i \subset S^*$.*

This implies, e.g., if all unitaries $U_1, \ldots, U_n$ have either x_1 or x_2 (or both) as input variable, then we can *not* construct a type-2 operator which outputs neither x_1 nor x_2. In particular, it follows an impossibility result:

Proposition 9 *It is impossible to construct a type-2 decryption operator from type-1 operators only, which outputs neither r nor c registers.*

Proof (Sketch; the full proof can be found in App. D.2). According to Thm. 8, a type-2 decryption operator which outputs neither c nor r can only exist if there is a unitary U with input variables $I \subset \{m, m', t\}$ (cf. Fig. 5), where t denotes auxiliary information that could help to recover the randomness, and m' denotes the decryption of $\mathtt{Enc}_{\mathsf{pk}}(m)$ which may differ from m in case of a decryption failure.

Intuitively, given only m, m', and possibly some auxiliary tag t, there is in general not enough information to reconstruct the original ciphertext c or the encryption randomness r. Moreover, such a mapping is not well-defined: for correctly decrypting ciphertexts we typically have $m = m'$, while most pairs (m, m') with $m \neq m'$ do not correspond to any meaningful encryption/decryption process. Hence, the desired unitary cannot exist. □

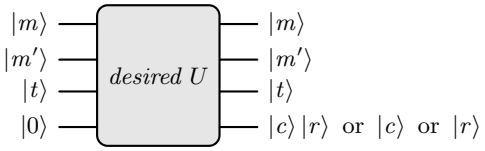

Fig. 5: The desired type-1 operator which would allow us to build a type-2 decryption operator.

4 Strongly Decryptable PKE

As we have seen, defining a reversible type-2 decryption operator brings many challenges. To overcome these challenges, we first consider a restricted class of PKE schemes which have a "very good decryption property" that makes many of the above problems disappear. Namely, what we want is that, for every honestly generated ciphertext, it is always possible to (deterministically and efficiently) decrypt that ciphertext *and* also recover the *unique* randomness which generated that ciphertext. This property, in particular, implies that every message-randomness pair maps to a different ciphertext, and hence the ciphertext space is at least as large as the product of plaintext and randomness spaces. We call this property *strong decryptability* defined below:

Definition 10 (Strong Decryptability) *Let Π be a* PKE *scheme. We call Π* strongly decryptable *if there exists a (deterministic and efficient[8]) algorithm*

[8] We could, in theory, make a distinction on whether `sDec` exists but is efficient or not. This would yield a similar distinction as 'isometric' vs. 'efficiently isometric' in [GKS21]. We will not look at this refinement, however, as it would not bring us any particular insight, and we will stick to the efficient case.

$\mathtt{sDec} : \mathcal{SK} \times \mathcal{C} \to (\mathcal{M} \times \mathcal{R}) \cup \{\perp\}$ *(for **s**trong **dec**ryption) such that, for any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$ *and any* $(m, r) \in \mathcal{M} \times \mathcal{R}$*, it holds that:*

$$\mathtt{Enc}_{\mathsf{pk}}(m; r) = c \iff \mathtt{sDec}_{\mathsf{sk}}(c) = (m, r). \tag{1}$$

In particular, strong decryptability implies that `Enc` is injective, and therefore strongly decryptable schemes are *isometric*. We will see in the next subsection how this isometric property is actually efficient, i.e., strongly decryptable schemes are *efficiently isometric*. However, the inclusion is strict: there exist efficiently isometric schemes which are not strongly decryptable. In fact, it is easy to show the following (the proof can be found in Appendix D.3):

Proposition 11 *Strong decryptability is* independent *of both recoverability and perfect correctness, i.e., strong decryptability does not imply either of the other two properties, nor does either of them imply strong decryptability.*

How realistic is strong decryptability as a property? Do strongly decryptable PKE schemes exist at all? The RSA trapdoor permutation in combination with OAEP is an example – even if this scheme is not quantum-resistant, it still shows that the concept is not purely theoretical. For sure, not every real-world PKE scheme is strongly decryptable, e.g., the canonical LWE scheme loses information about the used randomness during encryption. One might argue that, if a scheme is strongly decryptable, then there would be no reason to prefer the (possibly more error-prone) normal `Dec` algorithm over the more informative `sDec` algorithm during decryption. However, this is not necessarily the case: maybe `sDec` exists but is much slower or wasteful than the normal decryption algorithm.

Do PKE schemes exist which are at the same time perfectly correct *and* recoverable *and* strongly decryptable? Yes: an example is again the RSA-OAEP construction. However, there are also PKE schemes which are both perfectly correct and recoverable, but *not* strongly decryptable, and there are efficiently isometric PKE schemes which have neither of the three properties. This is stated in the following proposition whose proof is given in Appendix D.4.

Proposition 12 *Among* PKE *schemes which are both perfectly correct and recoverable, some are strongly decryptable, and others are not. Among efficiently isometric* PKE *schemes there are schemes which are neither recoverable, nor perfectly correct, nor strongly decryptable.*

The situation is depicted in Fig. 6: we can see strong decryptability as a (strict) "refinement" of the efficiently isometric property for PKE schemes:

4.1 Type-2 Operators and Oracles for Strongly Decryptable PKE

Since the strong decryption algorithm `sDec` is failure-free (*for honestly generated ciphertexts*), we can convert it to a type-2 unitary operator which takes as input a ciphertext register, and outputs (possibly entangled) plaintext and randomness registers (and zero ancillas to make input and output space size match).

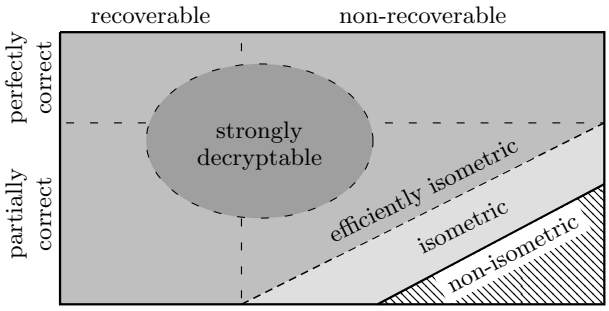

Fig. 6: Classification of different PKE schemes (updated from [GKS21]).

This type-2 $\mathtt{sDec}$ operator will be the main ingredient to define decryption oracles in our qIND-{qCCA1,qCCA2} security notion for strongly decryptable PKE. We denote this operator by $U^{(2)}_{\mathtt{sDec}}$, and we note that knowledge of the secret key is required for its efficient realization (cf. Fig. 7 for the explicit construction).

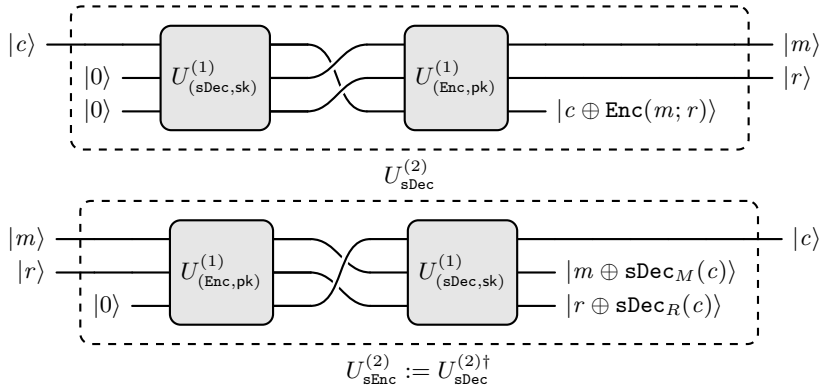

Fig. 7: Type-2 *strong* decryption operator, and its adjoint. $\mathtt{sDec}_M$ and $\mathtt{sDec}_R$ denote the partial functions of $\mathtt{sDec}$ which compute the first resp. second part of the tuple, i.e., $\mathtt{sDec}(c) =: (\mathtt{sDec}_M(c), \mathtt{sDec}_R(c))$. Note that the output of $U^{(2)}_{\mathtt{sDec}}$ *only* corresponds to the actual output of $\mathtt{sDec}$ if the input ciphertext decrypts correctly. For any other ciphertext, the output ancilla register $|c \oplus \mathtt{Enc}(m;r)\rangle$ is not the zero-register and, thus, $U^{(2)}_{\mathtt{sDec}}$ does not behave as a unitary. We identify as $\perp$ any output for which this is the case.

How does its adjoint operator $U^{(2)\dagger}_{\mathtt{sDec}}$ look like? It can be seen as a particularly powerful type-2 encryption operator, which (requiring knowledge of the secret key) takes as input message and randomness registers, and only outputs a ciphertext register (unlike the "regular" $U^{(2)}_{\mathtt{Enc}}$, which outputs two, possibly entangled, ciphertext *and* randomness registers). This is also shown in Fig. 7 (notice that all type-1 operators are self-adjoint). Sometimes we will abuse notation and write the unitary operator $U^{(2)\dagger}_{\mathtt{sDec}}$ as simply $U^{(2)}_{\mathtt{sEnc}}$ (where $\mathtt{sEnc}$ stands for **s**trong **enc**ryption, although there is no classical definition for an algorithm $\mathtt{sEnc}$).

Given the operator $U^{(2)}_{\mathtt{sEnc}}$, we can finally show that strongly decryptable schemes are *efficiently* isometric. In order to do so, it is sufficient to show an explicit, efficient construction of the operator $U^{(2)}_{\mathtt{Enc}}$. But this is trivial, since now we have the (more powerful) strong encryption operator $U^{(2)}_{\mathtt{sEnc}}$.

Proposition 13 *Strongly decryptable PKE schemes are efficiently isometric.*

Proof. The construction of $U^{(2)}_{\mathtt{Enc}}$ is depicted in Fig. 8. □

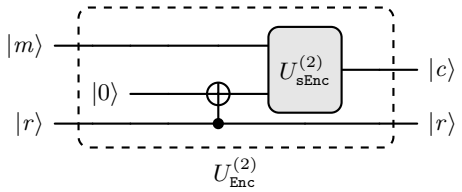

Fig. 8: Strongly decryptable PKE schemes are efficiently isometric (cf. Def. 4).

Given the operators $U^{(2)}_{\mathtt{sEnc}}$ and $U^{(2)}_{\mathtt{sDec}}$, we also can define the corresponding *oracles*, as interfaces offered by the challenger to the adversary.

Definition 14 (Type-2 Strong Oracles) *Let* $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{sDec})$ *be a strongly decryptable* PKE *scheme. The* type-2 strong encryption and decryption oracle*,* $O^{(2)}_{\mathtt{sEnc}}$ *and* $O^{(2)}_{\mathtt{sDec}}$*, are defined by the procedures in Fig. 9.*

Oracle $O^{(2)}_{\mathtt{sEnc}}(\mu)$ on input a plaintext quantum state $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$

1: $r \leftarrow^{\$} \mathcal{R}$
2: $\chi := U^{(2)}_{\mathtt{sEnc}}(\mu \otimes |r\rangle\langle r| \otimes |0\ldots0\rangle\langle 0\ldots0|)$
3: **return** $\chi \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$

Oracle $O^{(2)}_{\mathtt{sDec}}(\chi)$ on input a ciphertext quantum state $\chi \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$

1: $\mu\upsilon\zeta := U^{(2)}_{\mathtt{sDec}}(\chi \otimes |0\ldots0\rangle\langle 0\ldots0|)$, where $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$, $\upsilon \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$ and $\zeta \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$
2: Measure ζ in the computational basis
3: **if** outcome is not $0\ldots0$, **then return** $|\bot\rangle\langle\bot|$; **else return** $\mu\upsilon$

Fig. 9: Type-2 *strong* encryption and decryption oracle.

Note that, in Def. 14, using $U^{(2)}_{\mathtt{Enc}}$ instead of $U^{(2)}_{\mathtt{sEnc}}$ and tracing out the classical randomness to obtain χ as in the definition of $O^{(2)}_{\mathtt{Enc}}$ in Fig. 2, leads to an equivalent definition of $O^{(2)}_{\mathtt{sEnc}}$ as in Fig. 9. In other words, $O^{(2)}_{\mathtt{Enc}} = O^{(2)}_{\mathtt{sEnc}}$.

4.2 Security for Strongly Decryptable PKE

First, notice that (due to Prop. 13) we can define qIND-qCPA security for strongly decryptable schemes (and all the related separations and constructions)

exactly as in [GKS21], using the $O^{(2)}_{\mathtt{Enc}}$ oracle implemented through the $\mathtt{sDec}$ algorithm, as in Fig. 9, and define qIND-qCPA accordingly. Because it will be more convenient for us in the next steps, we consider an equivalent formulation of qIND-qCPA (best suited for the exposition of the rest of our work) which uses a RoR (Real-or-Random) approach instead of the LoR (Left-or-Right) approach from Def. 7. We adapt this definition from the Q3 scenario from [AGM18], by using a fully mixed quantum state $\tau_{\mathcal{M}}$ (i.e., the uniform density matrix in $\mathfrak{D}(\mathcal{M})$) as a "random" plaintext state: $\tau_{\mathcal{M}}$ is the quantum analogue of a classical random message, as it can be evaluated to any outcome with equal probability regardless of the chosen observable measurement.

---

qIND-qATK$_{\mathtt{sDec}}(\Pi, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda)$, where ATK is either CPA or CCA 1

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, implements $O^{(2)}_{\mathtt{sEnc}}$ and $O^{(2)}_{\mathtt{sDec}}$, and flips a coin $b \leftarrow^{\$} \{0,1\}$

2: $\mathcal{A}_1^{O^{(2)}_{\mathtt{sEnc}}, O^{(2)}_{\mathtt{sDec}}}(\mathsf{pk}, \lambda)$ prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ

3: $\mathfrak{C}$ samples fresh (classical) randomness r, encrypts through $O^{(2)}_{\mathtt{sEnc}}$ either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$), and sends the response χ to $\mathcal{A}_2$

4: $\mathcal{A}_2^{O^{(2)}_{\mathtt{sEnc}}}$ receives states χ and σ, and outputs a bit b'

5: **if** $b = b'$, **then return** win; **else return** reject

---

Fig. 10: The qIND-qCPA and qIND-qCCA1 experiment (RoR version) – without resp. with gray text – for strongly decryptable PKE schemes.

It is trivial to show that the LoR and RoR formulations are equivalent, as in the classical case. From now on, we will always consider the RoR version of qIND-qCPA. Stepping up to qIND-qCCA1 security, it is sufficient to modify the game above by giving access to $O^{(2)}_{\mathtt{sDec}}$ to the pre-challenge adversary $\mathcal{A}_1$.

Definition 15 (qIND-{qCPA,qCCA1} Security for Str. Decr. PKE) *Let Π be a strongly decryptable* PKE *scheme, and let* ATK *be either* CPA *or* CCA1*. Then Π is* qIND-qATK *secure, iff the advantage of any QPT adversary $\mathcal{A}$ to win* qIND-qATK$_{\mathtt{sDec}}(\Pi, \mathcal{A}, \lambda)$ *(defined as in Fig. 10) is negligible, i.e.,*

$$\mathbf{Adv}^{\text{qIND-qATK}_{\mathtt{sDec}}}_{\Pi}(\mathcal{A}) \coloneqq \left|\Pr[\text{qIND-qATK}_{\mathtt{sDec}}(\Pi, \mathcal{A}, \lambda) \to \mathsf{win}] - \frac{1}{2}\right| \leq \text{negl}(\lambda).$$

For defining qIND-qCCA2, we follow the approach of Alagic et al. [AGM18] of comparing so-called *Test* and *Fake* games. In the Test game, the adversary can send *any* decryption query in the pre- *and* post-challenge phase, including the *challenge ciphertext*, whereas the only goal of the Fake game is to detect a replay of the challenge ciphertext. However, we face a problem if we give the adversary access only to the oracles we previously defined. Consider the following PKE scheme:

Construction 16 *Let* $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{sDec})$ *be a strongly decryptable* PKE *scheme with ciphertext space* $\mathcal{C}$*. Then we define* $\overline{\Pi} = (\overline{\mathtt{KGen}}, \overline{\mathtt{Enc}}, \overline{\mathtt{Dec}}, \overline{\mathtt{sDec}})$ *as a modification of* Π *with ciphertext space* $\overline{\mathcal{C}} := \mathcal{C} \times \{0,1\}$*,* $\overline{\mathtt{KGen}} := \mathtt{KGen}$ *and the other algorithms are as defined in Fig. 11.*

$\overline{\mathtt{Enc}}_{\mathsf{pk}}(m; r)$

1: **return** $\mathtt{Enc}_{\mathsf{pk}}(m; r)||0$

$\overline{\mathtt{Dec}}_{\mathsf{sk}}(c||\overline{c})$

1: **return** $\mathtt{Dec}_{\mathsf{sk}}(c)$

$\overline{\mathtt{sDec}}_{\mathsf{sk}}(c||\overline{c})$

1: **if** $\overline{c} = 0$ **then**
2: **return** $\mathtt{sDec}_{\mathsf{sk}}(c)$
3: **return** $\perp$

Fig. 11: A modification $\overline{\Pi}$ of the scheme Π. It is easy to see that $\overline{\Pi}$ is strongly decryptable if the underlying scheme Π already is. Any ciphertext in $\overline{\mathcal{C}}$, we denote as $c||\overline{c}$, where $c \in \mathcal{C}$ and $\overline{c}$ is the extra bit.

This scheme is clearly not classical IND-CCA2 secure. Once the adversary obtains the challenge ciphertext of form $c^*||0$, she can easily win the game by flipping the last bit and sending $c^*||1$ as a valid query to her decryption oracle. As $\overline{\mathtt{Dec}}_{\mathsf{sk}}(c^*||0) = \overline{\mathtt{Dec}}_{\mathsf{sk}}(c^*||1)$, this query allows her to trivially distinguish the real-or-random case.

However, it turns out that this scheme becomes CCA2 secure in the quantum setting if the adversary has only access to the oracles $O^{(2)}_{\mathtt{sEnc}}$ and $O^{(2)}_{\mathtt{sDec}}$. The qIND-qCCA2 security of $\overline{\Pi}$ reduces to the qIND-qCCA2 security of the underlying scheme, and, in particular, the previous described attack is no longer possible in this setting. Consider an adversary who receives a challenge ciphertext of form $\chi^* \otimes |0\rangle\langle 0|$ and sends $\chi^* \otimes |1\rangle\langle 1|$ to her *strong* decryption oracle $O^{(2)}_{\mathtt{sDec}}$. As the latter is a ciphertext, which cannot be generated by any message-randomness pair by definition, the measurement of the ancilla register ζ in Fig. 9 is non-zero and $|\perp\rangle\langle\perp|$ is outputted as response. Thus, the decryption query which allowed the adversary to win the game classically does not give any information to the adversary if the adversary has only access to $O^{(2)}_{\mathtt{sDec}}$ as decryption oracle. We circumvent this issue by providing access to a *classical decryption oracle* $\mathtt{Dec}_{\mathsf{sk}}$ as well, which outputs $\mathtt{Dec}_{\mathsf{sk}}(c)$ on any input classical ciphertext $c \in \mathcal{C}$. [9]

The Test game looks like the previously defined experiments for qIND-qCPA and qIND-qCCA1, except that the qIND-qCCA2 adversary has access to both encryption and decryption oracles in the pre- *and* post-challenge phase, and is also provided with the additional classical decryption oracle.

Experiment 17 (qCCA2 Test) *The experiment* qCCA2-Test$(\Pi, \mathcal{A}, \lambda)$ *for a strongly decryptable PKE scheme* Π *is defined in Fig. 12.*

[9] We give a formal proof of "qIND-qATK ⇒ IND-ATK" in Appendix C.3 for a larger class of PKE schemes. It is easy to see that the given proofs there easily translate to the the setting of strongly decryptable PKE schemes.

qCCA2-Test$(\Pi, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda)$

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, implements $O^{(2)}_{\mathtt{sEnc}}$, $O^{(2)}_{\mathtt{sDec}}$ and $\mathtt{Dec}_{\mathsf{sk}}$, and samples $b \leftarrow^{\$} \{0,1\}$
2: $\mathcal{A}_1^{O^{(2)}_{\mathtt{sEnc}}, O^{(2)}_{\mathtt{sDec}}, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ
3: $\mathfrak{C}$ prepares fresh (classical) randomness r, encrypts through $O^{(2)}_{\mathtt{sEnc}}$ either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) and sends the response χ to $\mathcal{A}_2$
4: $\mathcal{A}_2^{O^{(2)}_{\mathtt{sEnc}}, O^{(2)}_{\mathtt{sDec}}, \mathtt{Dec}_{\mathsf{sk}}}$ receives χ and σ, and outputs a bit b'
5: **if** $b = b'$, **then return** win; **else return** reject

Fig. 12: The qCCA2-Test game for strongly decryptable PKE schemes Π.

In the Fake game, instead, the goal is to detect replay attacks of the adversary. This is done through the replay detection technique developed in [AGM18] for the Q3 case: the challenger discards the challenge plaintext μ, replacing it by half of a maximally-entangled state $|\varphi^+\rangle_{\mu'\mu''}$ on two registers μ' and μ''. She sends the encryption of μ' (with some randomness r) to the adversary, whereas she stores the other half (μ'') together with r. This enables the challenger to detect cheating: For any (valid, i.e., successful) decryption query in the post-challenge phase, the challenger recovers (and measures) the randomness and checks if it matches the stored randomness. If the randomness is different, the challenger returns the plaintext state. Otherwise, she checks for entanglement between the decrypted message and the stored half (μ''). To do so, she uses the two-outcome projective measurement $\{\Pi^+, \mathbb{1} - \Pi^+\}$ which tests if a state has the same or different support as $|+\rangle$. We denote the first outcome by 0, and the second outcome by 1. Accordingly if the outcome is 0, the challenger succeeds to detect a replay attack, and the game ends by outputting cheat. If the outcome is 1, instead, the challenger returns the recovered decrypted message[10]. When the adversary terminates and outputs a bit, that bit is ignored, and the game terminates with either cheat or reject (which is – from the adversary's perspective – equivalent to *winning*) with probability $\frac{1}{2}$.

Experiment 18 (qCCA2 Fake) *The experiment* qCCA2-Fake$(\Pi, \mathcal{A}, \lambda)$ *for a strongly decryptable PKE scheme Π is defined in Fig. 13.*

Having the Test game and Fake game defined, we can define qIND-qCCA2 security for strongly decryptable PKE schemes:

Definition 19 (qIND-qCCA2 Security) *Let Π be a strongly decryptable PKE. Then Π is* qIND-qCCA2 *secure, iff for any QPT adversary $\mathcal{A}$ it holds:*

$$\Pr[\text{qCCA2-Test}(\Pi, \mathcal{A}, \lambda) \to \mathsf{win}] - \Pr[\text{qCCA2-Fake}(\Pi, \mathcal{A}, \lambda) \to \mathsf{cheat}] \leq \text{negl}(\lambda).$$

Note that, as in [AGM18], we compare the "winning" probability of an adversary with the "cheat-detection" probability *without* absolute value. This is

[10] Note that the measurement of the randomness might have affected this message. However, this is not a problem in the context of the *Fake* game, since here we are only interested in detecting replay attacks.

qCCA2-Fake($\Pi, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda$)

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$ and implements $O^{(2)}_{\mathtt{sEnc}}$, $O^{(2)}_{\mathtt{sDec}}$ and $\mathtt{Dec}_{\mathsf{sk}}$
2: $\mathcal{A}_1^{O^{(2)}_{\mathtt{sEnc}}, O^{(2)}_{\mathtt{sDec}}, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ
3: $\mathfrak{C}$ discards μ, prepares $|\varphi^+\rangle_{\mu'\mu''}$ and fresh randomness r, and stores (μ'', r); then $\mathfrak{C}$ encrypts through $O^{(2)}_{\mathtt{sEnc}}$ (using r) μ' and sends the response χ to $\mathcal{A}_2$
4: $\mathcal{A}_2^{O^{(2)}_{\mathtt{sEnc}}, D_{|\mathsf{fake}\rangle}, D_{\mathsf{fake}}}$ receives χ and σ, where $D_{|\mathsf{fake}\rangle}$, D_{fake} are defined as in Fig. 14
5: After $\mathcal{A}_2$ terminates, $\mathfrak{C}$ draws a bit $b \leftarrow^{\$} \{0, 1\}$ at random
6: **if** $b = 1$, **then return** cheat; **else return** reject

Fig. 13: The qCCA2-Fake game for strongly decryptable PKE schemes Π.

$D_{|\mathsf{fake}\rangle}$ on input $\chi \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$

1: $\mu\upsilon\zeta \coloneqq U^{(2)}_{\mathtt{sDec}}(\chi \otimes |0\ldots0\rangle\langle 0\ldots0|)$, where $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$, $\upsilon \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$ and $\zeta \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$
2: Measure ζ in the computational basis
3: **if** outcome is not $0\ldots0$, **then** Send $|\bot\rangle\langle\bot|$ to $\mathcal{A}_2$
4: **else do**
5: Measure υ in the computational basis, and store $r' \leftarrow \mathfrak{M}(\upsilon)$
6: **if** outcome (r') is r, **then**
7: Apply $\{\Pi^+, \mathbb{1} - \Pi^+\}$ to $\mu\mu''$
8: **if** outcome is 0, **then**
9: Abort game and **return** cheat; // replay detected
10: Send $\mu \otimes |r'\rangle\langle r'|$ to $\mathcal{A}_2$

D_{fake} on classical input $c \in \mathcal{C}$

1: Prepare the quantum state $\chi \coloneqq |c\rangle\langle c| \otimes |0\ldots0\rangle\langle 0\ldots0|$, and query $D_{|\mathsf{fake}\rangle}(\chi)$
// If challenge ciphertext was measured and then replayed, this step will detect cheating.
2: Discard response of $D_{|\mathsf{fake}\rangle}$
3: Send $\mathtt{Dec}_{\mathsf{sk}}(c)$ to $\mathcal{A}_2$

Fig. 14: $D_{|\mathsf{fake}\rangle}$ and D_{fake} are used as the decryption oracle in the post-challenge phase of the qCCA2-Fake game for strongly decryptable schemes Π. Note, that $D_{|\mathsf{fake}\rangle}$ *without* lines 5-9 corresponds to $O^{(2)}_{\mathtt{sDec}}$ and line 3 of D_{fake} to $\mathtt{Dec}_{\mathsf{sk}}$.

due to the fact that an adversary can always artificially inflate the absolute difference between these two probabilities by replaying the challenge ciphertext (thereby winning the *Fake* game), but purposedly ignoring this information (e.g., outputting a random bit). What the adversary should not be able to do (if the scheme is secure) is the other way around: winning the *Test* game without replaying the challenge. In other words, qIND-qCCA2 security would be an unachievable notion if we kept the absolute value. These security definitions above are specific to strongly decryptable schemes, but show the strategy that we are going to adopt in the general case. In the following sections, we will investigate achievability and separations for a larger class of PKE schemes.

5 Purification of PKE Schemes

If a PKE scheme is *not* strongly decryptable, that means that there is not always enough information in a produced ciphertext to recover both m and r (at least, not in any efficient way). In other words, some information *gets lost* during encryption, and therefore no strong decryption algorithm $\mathtt{sDec}$ can exist. In the classical setting, this leads to two possible outcomes: either this "information loss" only impacts the randomness recovery aspect, or it also impacts the message recovery (only in this case, the scheme will be subject to decryption failures). But in a quantum setting, and specifically for type-2 operators, there is no such distinction: since the *whole* computation needs to be reversible, efficient recovery of both m and r is necessary. In other words, this information loss for non-strongly decryptable schemes becomes a problem when defining a type-2 decryption, which take as input only the ciphertext *without the original randomness* used during the encryption.

To overcome this problem, we introduce a workaround: Given a (generally non-strongly decryptable) PKE scheme Π, we associate it to a family of "quasi-PKE schemes" which *are* strongly decryptable (in some sense), and yet they behave "as much as possible" like the original scheme. More precisely, we define a *purification* of Π as a tuple of algorithms which "mimic" the original PKE, but *do not* erase information during encryption, rather "moving it" (storing it) into an *auxiliary tag* t. This tag acts as a "backup" that allows, together with the ciphertext, to recover the original message and randomness (at least in theory). From a quantum perspective, this approach models the fact that information loss during encryption is incompatible with reversible computation (unless a measurement is performed). Purification can thus be viewed as a mechanism to explicitly account for and reverse this loss.

Clearly, this definition is too vague. Instead of refining it, we proceed by directly defining a *constructive and efficient* tuple of algorithms which behaves like desired. We call this tuple *the purification* of a given PKE scheme.

Definition 20 (Purified Schemes) *Let* $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a* PKE *scheme. We call* $\Pi^{\diamondsuit} = (\mathtt{KGen}^{\diamondsuit}, \mathtt{Enc}^{\diamondsuit}, \mathtt{Dec}^{\diamondsuit})$ *the* purification *of* Π *(or the* purified scheme *in short), which is defined as in Fig. 15.*

$\mathtt{KGen}^{\Diamond}(\lambda; r)$

1: **return** $\mathtt{KGen}(\lambda; r)$

$\mathtt{Enc}^{\Diamond}_{\mathsf{pk},\mathsf{sk}}(m; r)$

1: $c \leftarrow \mathtt{Enc}_{\mathsf{pk}}(m; r)$
2: $e \leftarrow m \oplus \mathtt{Dec}_{\mathsf{sk}}(\mathtt{Enc}_{\mathsf{pk}}(m; r))$
3: $t := (e, r)$
4: **return** (c, t)

$\mathtt{Dec}^{\Diamond}_{\mathsf{sk}}(c, t)$

1: $m' \leftarrow \mathtt{Dec}_{\mathsf{sk}}(c)$
2: $(e, r) \leftarrow t$
3: $m \leftarrow m' \oplus e$
4: **return** (m, r)

Fig. 15: The Purification $\Pi^{\Diamond}$ of a PKE scheme Π.

Notice that $\Pi^{\Diamond}$ is *not* a PKE as from Def. 2: the purified encryption $\mathtt{Enc}^{\Diamond}$ requires the secret key sk, and outputs a tag t along with the ciphertext c, while the purified decryption $\mathtt{Dec}^{\Diamond}$ also gets the t as input. The tag is of the form $t = (e, r)$, consisting of a value $e \in \mathcal{M}$ (the "error") and of the originally used randomness r. Value e is the *difference* between the original message m and the encrypted and decrypted message $\mathtt{Dec}_{\mathsf{sk}}(\mathtt{Enc}_{\mathsf{pk}}(m))$. Clearly, e vanishes if there is no decryption failure. For ease of notation, we do not consider the case that $\mathtt{Dec}$ returns an extra symbol $\perp$ (*explicit rejection*). Instead, we consider a rejection (decryption failure) to be a valid (but wrong) return value in the message space (*implicit rejection*)[11]. This way, given the right tag, even if a rejection occurs it is always possible to recover the original message which was encrypted.

A purification is just an abstraction that "approximates" the original PKE scheme while at the same time preserving all information during encryption. As such, we cannot translate directly the security notions for strongly decryptable schemes from Sec. 4 to this new case (we do not even have a definition of strong decryptability for purified schemes). We can, however, define ad-hoc security notions for purified schemes, which mimic the notion that we would like to express *if the scheme were a* PKE. We can then unload the security definition for a PKE on the corresponding security notion for its purification. This is the plan that we are going to develop in the next sections.

5.1 Quantum Operators and Oracles for Purified Schemes

Given Def. 20, we can implement $\mathtt{Enc}^{\Diamond}$ and $\mathtt{Dec}^{\Diamond}$ as type-1 unitary operators $U^{(1)}_{\mathtt{Enc}^{\Diamond}}$ and $U^{(1)}_{\mathtt{Dec}^{\Diamond}}$, respectively, for any purified scheme $\Pi^{\Diamond}$. Thanks to their structure, this in turn allows us to easily implement operators $\mathcal{U}_{\mathtt{Enc}^{\Diamond}}$ and $\mathcal{U}_{\mathtt{Dec}^{\Diamond}}$[12] (cf. Fig. 16) which at least behave like type-2 operators, as long as no decryption failure occurs and the tag is valid. Notice that the efficient implementation of all these operators requires knowledge of the secret key sk.

Having those operators constructed, we can now define the corresponding oracles for the security games. For the decryption operator $\mathcal{U}_{\mathtt{Dec}^{\Diamond}}$ we need to be particularly cautious: In case the tag/ciphertext pair was invalid, the output states α and β are entangled with the other registers, which therefore cannot be

[11] It is always possible to adapt the description to explicit rejection, by adding an extra bit to the message space, but that would lead to unnecessary bookkeeping.

[12] Notice that $\mathcal{U}_{\mathtt{Dec}^{\Diamond}}$ is not unitary, but rather defined as a generic quantum channel, that's why we use the notation $\mathcal{U}$ rather than U.

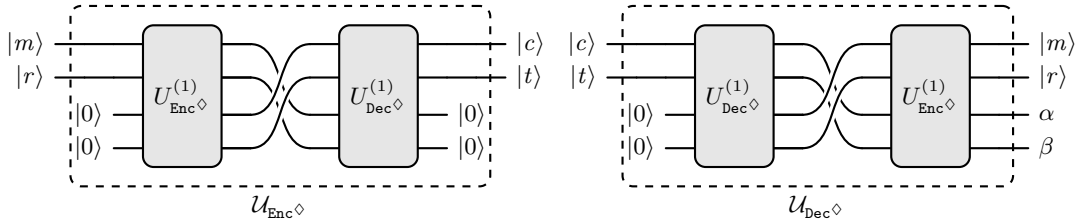

Fig. 16: Purified operators $\mathcal{U}_{\mathtt{Enc}^\diamond}$ and $\mathcal{U}_{\mathtt{Dec}^\diamond}$. Note that if the input $(|c\rangle, |t\rangle)$ is a valid ciphertext-tag pair, the output quantum states α, β are both $|0\rangle$.

sent to the adversary. For avoiding this entanglement, the oracle measures both states α, β before returning, and only outputs the other registers – which are supposed to contain message and randomness – if the result of the measurement is 0, otherwise it outputs $|\bot\rangle\langle\bot|$. Note that, if α, β are both 0, the tag was (with overwhelming probability) valid and the decryption successful. In the end, even if α, β were not zero states but were measured to be 0, this measurement assures that sent quantum state to the adversary is not entangled anymore.

There is, however, another challenge. It turns out that, if we define the oracles as previously described, even the resulting qIND-qCPA security notion is unachievable. The reason is that an adversary who can see the randomness used to generate a ciphertext, trivially wins the game: Once she receives the challenge ciphertext together with the corresponding tag, she can extract the randomness and re-encrypt the challenge messages using this randomness and compare with the challenge ciphertext.

For this reason, we do not define the oracles in the naive way: instead, the oracles output tags where the randomness is *permuted* via a post-quantum pseudo-random permutation $\pi : \mathcal{R} \to \mathcal{R}$, which is an additional parameter to the oracle[13]. In practice one can use, e.g., `AES` [AES01] – which is (believed to be) a post-quantum secure SKE – but any other family of quantum-resistant pseudo-random permutations that can be efficiently sampled is a possible candidate.

Definition 21 (Purified Quantum Oracles) *Let Π be a* PKE *scheme. Let $\pi : \mathcal{R} \to \mathcal{R}$ be a post-quantum pseudo-random permutation and U^π (resp. $U^{\pi^{-1}}$) a unitary implementation of π (resp. π^{-1}). The* purified quantum oracles $O^\diamond_{\mathtt{Enc},\pi}$ *and $O^\diamond_{\mathtt{Dec},\pi}$ are defined as in Fig. 17.*

Note that an adversary querying the decryption oracle must provide a valid tag register for the ciphertext to be decrypted. We emphasize that this requirement is not restrictive. For ciphertexts generated by the challenger, the adversary can always supply valid tags since these are returned by the challenger as part of the encryption. Moreover, with overwhelming probability, the adversary can also provide valid tags for ciphertexts she generated herself by setting the error

[13] Note that prior works [MS16, CETU21] likewise rely on (pseudo)random permutations. Importantly, this does not strengthen the underlying assumptions of our security definition: PKE exists under *Cryptomania* assumptions [Imp95], while PRPs can be constructed in *Minicrypt* from one-way functions (cf. [HILL99, GGM86, LR88]).

Oracle $O^{\Diamond}_{\mathtt{Enc},\pi}(\mu)$ on input quantum state $(\mu) \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$

1: $r \leftarrow^{\$} \mathcal{R}$
2: $\chi\varepsilon\rho \otimes |0\ldots0\rangle\langle 0\ldots0| \coloneqq \mathcal{U}_{\mathtt{Enc}\Diamond}(\mu \otimes |r\rangle\langle r| \otimes |0\ldots0\rangle\langle 0\ldots0|)$,
where $\chi \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$, $\varepsilon \in \mathfrak{D}(\mathfrak{H}_{\mathcal{E}})$ and $\rho \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$
3: **return** $\chi\varepsilon(U^{\pi}\rho)$

Oracle $O^{\Diamond}_{\mathtt{Dec},\pi}(\chi\varepsilon\upsilon)$ on input quantum state $(\chi\varepsilon\upsilon) \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$

1: $\mu\rho\zeta \leftarrow \mathcal{U}_{\mathtt{Dec}\Diamond}(\chi\varepsilon(U^{\pi^{-1}}\upsilon) \otimes |0\ldots0\rangle\langle 0\ldots0|)$,
where $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$, $\rho \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$ and $\zeta \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$
2: Measure ζ in the computational basis
3: **if** outcome is not $0\ldots0$, **then return** $|\bot\rangle\langle\bot|$; **else return** $\mu(U^{\pi}\rho)$

Fig. 17: Purified encryption and decryption oracle.

of the tag to the all-zero string (and the randomness of the tag as the previously used randomness). An error value different than the all-zero string only occurs when there is a decryption failure, which happens with negligible probability.

Since decryption failures occur only with negligible probability, the adversary can generate a tag for a locally encrypted plaintext $c \coloneqq \mathtt{Enc}_{\mathsf{pk}}(m; r)$ by setting $t = (0, r)$. This tag is *valid* with overwhelming probability, and therefore the resulting ciphertext-tag pair decrypts to m and r under the purified scheme with overwhelming probability. Note that this mirrors the classical setting, where an adversary may likewise construct ciphertexts that do not decrypt correctly.

However, the decryption oracle $O^{\Diamond}_{\mathtt{Dec},\pi}$ (cf. Fig. 17) expects tags in which the randomness has been permuted by π. To address this, the adversary is additionally provided with *classical* access to π (but *not* to π^{-1}). This allows her to generate tags which are valid with overwhelming probability for her locally computed ciphertexts: for $c \coloneqq \mathtt{Enc}_{\mathsf{pk}}(m; r)$, she asks the challenger to compute $\pi(r)$ and associates c with the tag $t = (0, \pi(r))$. Note the access to π will not enable the adversary to break the (post-quantum) pseudo-randomness of π, since the access is restricted to be classical.

At this point we want to comment that the use of this secret permutation is not *necessary* for qIND-qCPA and qIND-qCCA1 security. An alternative definition could be: send the randomness in *plain* to the adversary for any learning query, and trace out the randomness for the challenge ciphertext. This also avoids the previously described trivial attack of redoing the encryption locally with the same randomness. Such alternative definition is enticing, as it would avoid the pseudo-random permutation at all and, therefore, another extra oracle in the game. On the other hand, the downside of this approach is that encryptions of learning queries and the encrypted challenge would be of a different form, and one could argue that this would be an unfair disadvantage for the adversary. Furthermore, we require anyway a method for hiding the randomness returned to the adversary in the definition of qIND-qCCA2 security, as we will see later. Therefore, in the next subsection, we adopt the conservative choice of using the permutation-based approach (and offering the corresponding classical oracle to the adversary) for the qIND-qCPA and qIND-qCCA1 experiments already.

6 Quantum Security from Purified Schemes

For purified schemes, we can now use the oracles described in the previous section to define appropriate qIND-{qCPA,qCCA1,qCCA2} security notions.

qIND-qATK($\Pi^\diamondsuit, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda$), where ATK is either CPA or CCA1

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^\diamondsuit$ and flips a coin $b \leftarrow^{\$} \{0,1\}$
2: $\mathfrak{C}$ samples a pseudo-random permutation π and implements game oracles for $\mathcal{A}$
3: $\mathcal{A}_1^{O^\diamondsuit_{\mathtt{Enc},\pi}, O_\pi, O^\diamondsuit_{\mathtt{Dec},\pi}, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends $\mathfrak{C}$ a challenge register μ
4: $\mathfrak{C}$ samples fresh (classical) randomness r, encrypts through $O^\diamondsuit_{\mathtt{Enc},\pi}$ either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) and sends the response $\chi\varepsilon\upsilon$ to $\mathcal{A}_2$
5: $\mathcal{A}_2^{O^\diamondsuit_{\mathtt{Enc},\pi}, O_\pi}$ receives states $\chi\varepsilon\upsilon$ and σ, and outputs a bit b'
6: **if** $b = b'$, **then return** win; **else return** reject

Fig. 18: The qIND-qCPA and qIND-qCCA1 experiment (cf. Exp. 22) – without resp. with gray text – for purified schemes $\Pi^\diamondsuit$.

Experiment 22 (Purified qIND-{qCPA,qCCA1} Experiment) *Let $\Pi^\diamondsuit$ be a purified scheme, and let* ATK *be either* CPA *or* CCA1*. The experiment* qIND-qATK($\Pi^\diamondsuit, \mathcal{A}, \lambda$) *is as defined in Fig. 18, where the adversary has access to the purified encryption oracle $O^\diamondsuit_{\mathtt{Enc},\pi}$, as well as a classical oracle O_π to the secret permutation π. If* ATK = CCA1*, the adversary has also access to the decryption oracles $O^\diamondsuit_{\mathtt{Dec},\pi}$ and $\mathtt{Dec}_{\mathsf{sk}}$ in the pre-challenge phase.*

Having the security games defined for purified schemes allows us to extend the security definitions to arbitrary schemes. This, in particular, allows us to define qIND-qCPA security for *any* PKE scheme (not only isometric ones), leading to an improvement over [GKS21] (and Def. 7).

Definition 23 (qIND-{qCPA,qCCA1} Security for PKE) *Let Π be a* PKE *scheme, and let* ATK *be either* CPA *or* CCA1*. Then Π is* qIND-qATK *secure, iff the advantage for any QPT adversary $\mathcal{A}$ to win the* qIND-qATK($\Pi^\diamondsuit, \mathcal{A}, \lambda$) *game against the purification $\Pi^\diamondsuit$ is negligible, i.e.,*

$$\mathbf{Adv}^{\text{qIND-qATK}}_{\Pi}(\mathcal{A}) := \left|\Pr[\text{qIND-qATK}(\Pi^\diamondsuit, \mathcal{A}, \lambda) \to \mathsf{win}] - \frac{1}{2}\right| \leq \text{negl}(\lambda).$$

It is easy to see that qIND-qCCA1 is a stronger notion than qIND-qCPA (the proof can be found in App. C.1), and that the quantum notions qIND-qCPA and qIND-qCCA1 imply theirs classical analogue IND-CPA and IND-pqCCA1 (the proof can be found in App. C.3). For defining qIND-qCCA2, we again follow the approach of [AGM18] of comparing Test and Fake games:

Experiment 24 (qCCA2 Test) *The experiment* qCCA2-Test($\Pi^\diamondsuit, \mathcal{A}, \lambda$) *for a purified PKE scheme $\Pi^\diamondsuit = (\mathtt{KGen}^\diamondsuit, \mathtt{Enc}^\diamondsuit, \mathtt{Dec}^\diamondsuit)$ is defined in Fig. 19.*

qCCA2-Test($\Pi^\diamondsuit, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda$)

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^\diamondsuit$ and flips a coin $b \leftarrow^{\$} \{0,1\}$
2: $\mathfrak{C}$ chooses a pseudo-random permutation π and implements the oracles for $\mathcal{A}$
3: $\mathcal{A}_1^{O^\diamondsuit_{\mathtt{Enc},\pi}, O^\diamondsuit_{\mathtt{Dec},\pi}, O_\pi, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ
4: $\mathfrak{C}$ prepares fresh (classical) randomness r, encrypts through $O^\diamondsuit_{\mathtt{Enc},\pi}$ (using r) either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) and sends the response $\chi\varepsilon\upsilon$ to $\mathcal{A}_2$
5: $\mathcal{A}_2^{O^\diamondsuit_{\mathtt{Enc},\pi}, O^\diamondsuit_{\mathtt{Dec},\pi}, O_\pi, \mathtt{Dec}_{\mathsf{sk}}}$ receives $\chi\varepsilon\upsilon$ and σ, and outputs a bit b'
6: **if** $b = b'$, **then return** win; **else return** reject

Fig. 19: The qCCA2-Test game for purified schemes $\Pi^\diamondsuit$.

Experiment 25 (qCCA2 Fake) *The experiment* qCCA2-Fake($\Pi^\diamondsuit, \mathcal{A}, \lambda$) *for a purified PKE scheme* $\Pi^\diamondsuit = (\mathtt{KGen}^\diamondsuit, \mathtt{Enc}^\diamondsuit, \mathtt{Dec}^\diamondsuit)$ *is defined in Fig. 20.*

qCCA2-Fake($\Pi^\diamondsuit, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda$)

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^\diamondsuit$
2: $\mathfrak{C}$ chooses a pseudo-random permutation π and implements the oracles for $\mathcal{A}$
3: $\mathcal{A}_1^{O^\diamondsuit_{\mathtt{Enc},\pi}, O^\diamondsuit_{\mathtt{Dec},\pi}, O_\pi, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ
4: $\mathfrak{C}$ discards μ, prepares $|\varphi^+\rangle_{\mu'\mu''}$ and fresh randomness r, and stores (μ'', r); then $\mathfrak{C}$ encrypts through $O^\diamondsuit_{\mathtt{Enc},\pi}$ (using r) μ' and sends the response $\chi\varepsilon\upsilon$ to $\mathcal{A}_2$
5: $\mathcal{A}_2^{O^\diamondsuit_{\mathtt{Enc},\pi}, D^\diamondsuit_{|\mathsf{fake}\rangle}, O_\pi, D^\diamondsuit_{\mathsf{fake}}}$ receives registers $\chi\varepsilon\upsilon$ and σ, where $D^\diamondsuit_{|\mathsf{fake}\rangle}$ and $D^\diamondsuit_{\mathsf{fake}}$ are defined as in Fig. 21
6: After $\mathcal{A}_2$ terminates, $\mathfrak{C}$ draws a bit b at random
7: **if** $b = 1$, **then return** cheat; **else return** reject

Fig. 20: The qCCA2-Fake game for purified schemes $\Pi^\diamondsuit$.

Having the Test game and Fake game for purified schemes defined, we can define qIND-qCCA2 security for *any* PKE scheme:

Definition 26 (qIND-qCCA2 Security for PKE) *Let Π be a PKE scheme. Then Π is* qIND-qCCA2 *secure, iff for any QPT adversary $\mathcal{A}$ (against the purification $\Pi^\diamondsuit$) it holds:*

$$\Pr[\text{qCCA2-Test}(\Pi^\diamondsuit, \mathcal{A}, \lambda) \to \mathsf{win}] - \Pr[\text{qCCA2-Fake}(\Pi^\diamondsuit, \mathcal{A}, \lambda) \to \mathsf{cheat}] \leq \mathrm{negl}(\lambda).$$

It easy to see that qIND-qCCA2 is stronger than qIND-qCCA1, and that qIND-qCCA2 than IND-CCA2 (proofs can found in the App. C.2 and C.3).

7 Constructions

In Sec. 5, we defined qIND-qCPA, qIND-qCCA1 and qIND-qCCA2 security for *any* PKE scheme following the RoR approach. Can PKE schemes achieve these notions? Similar to [GKS21], we consider the canonical hybrid encryption scheme which is built from a SKE scheme S and a PKE scheme P (cf. Fig. 22).

$D^{\Diamond}_{|\mathsf{fake}\rangle}$ on input $(\chi\varepsilon\upsilon) \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$

1: Compute $\mu\rho\zeta \leftarrow \mathcal{U}_{\mathsf{Dec}^{\Diamond}}(\chi\varepsilon(U^{\pi^{-1}}\upsilon) \otimes |0\dots0\rangle\langle 0\dots0|)$,
   where $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$, $\rho \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$ and $\zeta \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$
2: Measure ζ in the computational basis
3: **if** outcome is not $0\dots0$, **then** Send $|\bot\rangle\langle\bot|$ to $\mathcal{A}_2$
4: **else do**
5: Measure ρ in the computational basis, and store $r' \leftarrow \mathfrak{M}(\rho)$
6: **if** outcome (r') is r, **then**
7: Apply $\{\Pi^{+}, \mathbb{1} - \Pi^{+}\}$ to $\mu\mu''$
8: **if** outcome is 0, **then**
9: Abort game and **return** cheat; // replay detected
10: Send $\mu \otimes |\pi(r')\rangle\langle\pi(r')|$ to $\mathcal{A}_2$

$D^{\Diamond}_{\mathsf{fake}}$ on classical input $c \in \mathcal{C}$

1: Prepare $\chi \coloneqq |c||0\dots0||\pi(r)\rangle\langle c||0\dots0||\pi(r)|$, and query $D^{\Diamond}_{|\mathsf{fake}\rangle}(\chi)$
   // If challenge ciphertext was measured and then replayed, this step will detect cheating.
2: Discard response of $D^{\Diamond}_{|\mathsf{fake}\rangle}$
3: Send $\mathtt{Dec}_{\mathsf{sk}}(c)$ to $\mathcal{A}_2$

Fig. 21: $D^{\Diamond}_{|\mathsf{fake}\rangle}$ and $D^{\Diamond}_{\mathsf{fake}}$ are used as decryption oracle in the post-challenge phase of the qCCA2-Fake game for purified schemes $\Pi^{\Diamond}$. Note, that $D^{\Diamond}_{|\mathsf{fake}\rangle}$ *without* lines 5-9 corresponds to $O^{\Diamond}_{\mathsf{Dec},\pi}$ (cf. Fig. 17), and line 3 of $D^{\Diamond}_{\mathsf{fake}}$ to $\mathtt{Dec}_{\mathsf{sk}}$.

The construction encrypts a message m under S using a one-time symmetric key k to obtain the *symmetric* ciphertext component c_S, and encrypts k under P, yielding the *asymmetric* ciphertext component c_P. Decryption works in the natural way, i.e., first decrypt c_P to recover k, which is then used to decrypt c_S.

We then identify which security properties of the underlying SKE and PKE scheme are sufficient for the resulting hybrid scheme to satisfy our notions. Due to space limitations, we present only a condensed overview of the constructions and security proofs in this section. More details and the complete proofs are deferred to Appendix A.

$\mathtt{KGen}(\lambda; r)$

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^{P}(\lambda; r)$
2: **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathtt{Enc}_{\mathsf{pk}}(m; r)$

1: **Parse** r **as** (r_1, r_2, r_3)
2: $\mathsf{k} \leftarrow \mathtt{KGen}^{S}(\lambda; r_1)$
3: $c_S \leftarrow \mathtt{Enc}^{S}_{\mathsf{k}}(m; r_2)$
4: $c_P \leftarrow \mathtt{Enc}^{P}_{\mathsf{pk}}(\mathsf{k}; r_3)$
5: **return** (c_S, c_P)

$\mathtt{Dec}_{\mathsf{sk}}(c)$

1: **Parse** c **as** (c_S, c_P)
2: $\mathsf{k} \leftarrow \mathtt{Dec}^{P}_{\mathsf{sk}}(c_P)$
3: $m \leftarrow \mathtt{Dec}^{S}_{\mathsf{k}}(c_S)$
4: **return** m

Fig. 22: Hybrid encryption scheme Π obtained from a PKE P and a SKE S.

Importantly, in the Q2 setting both ciphertext components c_S and c_P may be in superposition and hence potentially entangled. Consequently, a challenger simulating a decryption oracle must handle such states carefully. Looking ahead, however, the challenger simulating a decryption oracle with cheat detection will

need to verify whether c_S encrypts half of a maximally-entangled state. To do this check, the challenger must measure the asymmetric component c_P.

Intuitively, this measurement is admissible because c_P is never returned to the adversary; only the resulting plaintext is output. Formally, c_P is therefore traced out. By the properties of reduced density operators and the partial trace (cf. [NC02, Box 2.6, Eq. (2.180)]), applying a measurement to the register c_P does not affect the adversary's reduced state, even if c_P is entangled with c_S. A formal statement and proof are deferred to Lemma 29 in the Appendix.

To establish security of the hybrid construction, we follow the approaches of [GKS21] and [AGM21]. Gagliardoni et al. [GKS21] proved, that such a hybrid scheme is qIND-qCPA secure (in their notion) if the SKE is qIND (cf. [GHS16]) and the PKE is *post-quantum* CPA (IND-pqCPA). In the qIND security notion for SKE, the adversary is given access to the encryption oracle, there is no learning phase, and the challenge – potentially in superposition – is of the form LoR. On the other hand, IND-pqATK refers to the standard post-quantum security notion where the adversary is quantum but all interactions with the encryption (and decryption) remain classical.

It turns out that these results also applies to our *general* qIND-qCPA security notion. Moreover, the same approach also yields qIND-qCCA1 security.

For qIND-qCCA2 security, we require the PKE to be IND-pqCCA2 secure, and the SKE to fulfill a suitable property of *quantum authentication*, which can be viewed as a 'natural' type-2 analogue of the Q3 notion of QAE introduced by [AGM21]. Informally, qAE guarantees that a QPT adversary cannot distinguish between interacting with the real encryption and decryption oracles, and an ideal world in which encryptions are replaced by encryptions of halves of EPR pairs while the decryption oracle rejects all fresh ciphertexts. We explicitly formalize this notion in Def. 35 in the Appendix.

Theorem 27 (Security of Hybrid Encryption) *Let Π be the hybrid encryption scheme obtained from a* PKE *scheme P and a* SKE *scheme S (cf. Fig. 22) and let* ATK *be either* CPA *or* CCA1*. For any* qIND-qATK *adversary $\mathcal{A}_1$ and for any* qIND-qCCA2 *adversary $\mathcal{A}_2$ against the purification $\Pi^\Diamond$, there exists adversaries $\mathcal{A}^P$ and $\mathcal{A}^S$ against P and S, respectively, such that*

$$\mathbf{Adv}^{\text{qIND-qATK}}_{\Pi}(\mathcal{A}_1) \leq \mathbf{Adv}^{\text{IND-pqATK}}_{P}(\mathcal{A}^P) + \mathbf{Adv}^{\text{qIND}}_{S}(\mathcal{A}^S),$$
$$\mathbf{Adv}^{\text{qIND-qCCA2}}_{\Pi}(\mathcal{A}_2) \leq \mathbf{Adv}^{\text{IND-pqCCA2}}_{P}(\mathcal{A}^P) + \mathbf{Adv}^{\text{qAE}}_{S}(\mathcal{A}^S).$$

8 Conclusions

In this section we want to briefly recap our contributions and point out some directions for future work. On the negative side, we proved that, in general, it is impossible to build an 'optimal' type-2 decryption oracle for PKE. On the positive side, we identified a new subclass of efficiently isometric PKE schemes, which we called *strongly decryptable*; for these schemes, we then showed that it is possible to define *type-2* encryption and decryption operators, which lead to

suitable notions of qIND-qCPA, qIND-qCCA1 and qIND-qCCA2 security. Furthermore, we developed a new transformation – the *purification* – which leads to the realization of encryption and decryption operators for purified 'quasi-PKE schemes', which in turn allows us to formally define qIND-qCPA, qIND-qCCA1 and qIND-qCCA2 security for *any* PKE scheme, thereby avoiding the prior limitation. Finally, we showed separations between type-2 notions for PKE, as well as constructions (through hybrid PKE/SKE encryption schemes) that achieve our notions. A few open questions remain for future investigation.

First, it would be interesting to look at the relations between our notion and the one from [CEV22]. We conjecture that, as in the case of SKE, they turn out to be incomparable. The idea is to take the separating examples – which are SKE schemes – and turn them into PKE schemes using the hybrid encryption approach. This adds an asymmetric component to the ciphertext (the encryption of the symmetric key) which, for honestly generated ciphertexts, will be classical and unentangled from the rest. By this observation, we believe that the argument from [CETU21], separating the type-1 and type-2 notions, extend to our setting. Still, a formal analysis is necessary to be certain.

Second, while our purification approach is technically convenient, it may appear somewhat artificial from a cryptographic point of view. There might be more elegant ways to account for the loss of reversibility during type-2 encryption, we leave this as an open problem.

Third, in the construction of hybrid qIND-qCCA2 PKE, we used a Q2 version of *quantum authenticated encryption* (which we dubbed qAE) as a requirement for the underlying SKE scheme, mainly because it is of easier formulation. However, just as in the case of [AGM18], we think it should be possible to relax this requirement to some weaker form of one-time authentication.

Fourth, it should be possible to further strengthen our impossibility result for type-2 operators. Currently, we only prove impossibility for type-2 operators built from elementary type-1's. This is enough for our case here, as it discards the most natural route to build type-2 decryption, and it is therefore enough to justify the search for alternative approaches such as our purification. However, we do not think that using different building blocks can circumvent a general impossibility. Intuitively, a type-2 decryption operator that outputs neither ciphertext nor randomness appears fundamentally incompatible with randomized encryption: a single plaintext can correspond to many different ciphertexts, thus there is no unique ciphertext that the inverse could recover from the plaintext alone. The challenge lies in formalizing this intuition, in particular in the presence of decryption failures.

Finally, and most interestingly, one could investigate whether there exist generic transformations – e.g., Fujisaki-Okamoto [FO99], Naor-Yung [NY90], or OAEP [BR95] – which, like in the classical case, succeed in lifting the security of a PKE scheme from qIND-qCPA to qIND-qCCA2.

References

ABF+16. Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. Computational security of quantum encryption. In Anderson C. A. Nascimento and Paulo Barreto, editors, *ICITS 16*, volume 10015 of *LNCS*, pages 47–71. Springer, Cham, August 2016.

AES01. Advanced Encryption Standard (AES). National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce, November 2001.

AGM18. Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 489–519. Springer, Cham, April / May 2018.

AGM21. Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Can you sign a quantum state? *Quantum*, 5:603, 2021.

AM17. Gorjan Alagic and Christian Majenz. Quantum non-malleability and authentication. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 310–341. Springer, Cham, August 2017.

AM24. Navid Alamati and Varun Maram. Quantum CCA-secure PKE, revisited. In Qiang Tang and Vanessa Teague, editors, *PKC 2024, Part II*, volume 14604 of *LNCS*, pages 193–226. Springer, Cham, April 2024.

AMRS20. Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 788–817. Springer, Cham, May 2020.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Berlin, Heidelberg, December 2011.

BDG20. Mihir Bellare, Hannah Davis, and Felix Günther. Separate your domains: NIST PQC KEMs, oracle cloning and read-only indifferentiability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 3–32. Springer, Cham, May 2020.

BHK15. Mihir Bellare, Dennis Hofheinz, and Eike Kiltz. Subtleties in the definition of IND-CCA: When and how should challenge decryption be disallowed? *Journal of Cryptology*, 28(1):29–48, January 2015.

BR95. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Berlin, Heidelberg, May 1995.

BZ13. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Berlin, Heidelberg, August 2013.

CETU21. Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. Relationships between quantum IND-CPA notions. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 240–272. Springer, Cham, November 2021.

CEV22. Céline Chevalier, Ehsan Ebrahimi, and Quoc-Huy Vu. On security notions for encryption in a quantum world. In Takanori Isobe and Santanu Sarkar, editors, *INDOCRYPT 2022*, volume 13774 of *LNCS*, pages 592–613. Springer, Cham, December 2022.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 356–383. Springer, Cham, August 2019.

DFMS22. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 677–706. Springer, Cham, May / June 2022.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Berlin, Heidelberg, August 1999.

FOPS01. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 260–274. Springer, Berlin, Heidelberg, August 2001.

GGM86. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, October 1986.

GHS16. Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 60–89. Springer, Berlin, Heidelberg, August 2016.

GKS21. Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. Quantum indistinguishability for public key encryption. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 12th International Workshop, PQCrypto 2021*, pages 463–482. Springer, Cham, 2021.

GYZ17. Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 342–371. Springer, Cham, August 2017.

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

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, Part II*, volume 12111 of *LNCS*, pages 389–422. Springer, Cham, May 2020.

Imp95. Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995.

KKVB02. Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5):050304, 2002.

KS22. Juliane Krämer and Patrick Struck. On quantum ciphertext indistinguishability, recoverability, and OAEP. In Jung Hee Cheon and Thomas Johans-

son, editors, *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022*, pages 286–306. Springer, Cham, September 2022.

LR88. Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2), 1988.

MS16. Shahram Mossayebi and Rüdiger Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *arXiv preprint arXiv:1609.03780*, 2016.

NC02. Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

Shr04. Tom Shrimpton. A characterization of authenticated-encryption as a form of chosen-ciphertext security. Cryptology ePrint Archive, Report 2004/272, 2004.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Cham, August 2019.

A Constructions (Full Version of Sec. 7)

In Sec. 5, we defined qIND-qCPA, qIND-qCCA1 and qIND-qCCA2 security for *any* PKE scheme following the RoR approach. Can PKE schemes achieve these notions? Similar to [GKS21], we consider the canonical hybrid encryption scheme which is built from a SKE and a PKE scheme. We then identify which security properties of the underlying SKE and PKE scheme are sufficient for the resulting hybrid scheme to satisfy our notions.

Construction 28 (Hybrid Encryption) *Consider the following construction* $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *obtained from a* PKE *scheme* $P = (\mathtt{KGen}^P, \mathtt{Enc}^P, \mathtt{Dec}^P)$ *and a* SKE *scheme* $S = (\mathtt{KGen}^S, \mathtt{Enc}^S, \mathtt{Dec}^S)$ *(cf. Fig. 23).*

Our construction encrypts a message m *under* S *using a one-time symmetric key* k *to obtain the* symmetric *ciphertext component* c_S*, and encrypts* k *under* P*, yielding the* asymmetric *ciphertext component* c_P*. Decryption works in the natural way, i.e., first decrypt* c_P *to recover* k*, which is then used to decrypt* c_S*.*

$\mathtt{KGen}(\lambda; r)$

1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^P(\lambda; r)$
2: **return** $(\mathsf{pk}, \mathsf{sk})$

$\mathtt{Enc}_{\mathsf{pk}}(m; r)$

1: **Parse** r **as** (r_1, r_2, r_3)
2: $\mathsf{k} \leftarrow \mathtt{KGen}^S(\lambda; r_1)$
3: $c_S \leftarrow \mathtt{Enc}^S_{\mathsf{k}}(m; r_2)$
4: $c_P \leftarrow \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}; r_3)$
5: **return** (c_S, c_P)

$\mathtt{Dec}_{\mathsf{sk}}(c)$

1: **Parse** c **as** (c_S, c_P)
2: $\mathsf{k} \leftarrow \mathtt{Dec}^P_{\mathsf{sk}}(c_P)$
3: $m \leftarrow \mathtt{Dec}^S_{\mathsf{k}}(c_S)$
4: **return** m

Fig. 23: Hybrid encryption scheme Π obtained from a PKE P and a SKE S.

Importantly, in the Q2 setting both ciphertext components c_S and c_P may be in superposition and hence potentially entangled. Consequently, a challenger simulating a decryption oracle must, in general, handle such states carefully. Looking ahead, however, the challenger simulating a decryption oracle with cheat detection will need to verify whether c_S encrypts half of a maximally-entangled state. To perform this check, the challenger must measure the asymmetric ciphertext component c_P. Note that this does not contradict the Q2 setting: the challenger measures only an internal register that is never returned to the adversary, while the externally visible interface of the oracle remains coherent.

To illustrate why this measurement is harmless, consider for example an adversary submitting a superposition of two ciphertexts $(c_S, c_P) = (\mathtt{Enc}^S_{\mathsf{k}_0}(m_0), \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}_0))$ and $(c'_S, c'_P) = (\mathtt{Enc}^S_{\mathsf{k}_1}(m_1), \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}_1))$ for two random keys $\mathsf{k}_0, \mathsf{k}_1$ and messages m_0, m_1, i.e., the state

$$\chi := \frac{1}{\sqrt{2}}(|c_S, c_P\rangle + |c'_S, c'_P\rangle).$$

If no measurement is performed, decryption produces a superposition of the plaintexts m_0 and m_1. However, the asymmetric ciphertext component – and any intermediate state derived from it – is never returned to the adversary. Thus, from the adversary's point of view, this register is effectively traced out.

If, instead, the challenger first measures the asymmetric component of the ciphertext, the state χ collapses either to the hybrid encryption of m_0 or m_1. Nevertheless, since the measured register is never returned, the adversary only observes the resulting reduced state on the plaintext register (and any reference registers it may hold). By the properties of the partial trace, this reduced state is identical in both cases: a measurement of the plaintext register returns either m_0 or m_1. Hence, measuring the asymmetric component does *not* affect the adversary's view. This is formalized in the following lemma:

Lemma 29 *Let $\mathfrak{C}$ be a challenger simulating a* type-2 *decryption oracle (e.g., as in Fig. 17 or 21) for the hybrid encryption scheme (Constr. 28) in the Q2 setting, and let $\chi = (\chi_S, \chi_P)$ be a (potentially quantum) ciphertext produced by the hybrid encryption scheme, where χ_S and χ_P denote the symmetric and asymmetric ciphertext components, respectively.*

Then, from the adversary's point of view, the following two decryption procedures are indistinguishable:

- *$\mathfrak{C}$ applies $\mathtt{Dec}^P$ to χ_P to obtain a (possibly quantum) key κ, uses κ to decrypt χ_S, and outputs the plaintext μ,*
- *$\mathfrak{C}$ first measures χ_P in the computational basis to obtain $c_P \leftarrow \mathfrak{M}(\chi_P)$, decrypts c_P to obtain a* classical *key k, uses k to decrypt χ_S, and finally outputs the plaintext μ.*

Proof. In both procedures, the decryption oracle ultimately returns only the plaintext μ, and neither the asymmetric ciphertext component χ_P nor any intermediate decryption state is given to the adversary. As a result, the adversary's view is fully determined by the reduced state on its own register together with the output μ.

Formally, the register containing χ_P is – since it is never returned to the adversary – traced out. By the properties of reduced density operators and the partial trace (cf. [NC02, Box 2.6, Eq. (2.180)]), applying a measurement to the register χ_P does not affect the reduced state (in our case μ) on the adversary's system. This remains true even if χ_P is initially entangled with χ_S.

Consequently, measuring χ_P prior to decryption does not change the adversary's view, and the two procedures are indistinguishable. □

A.1 Construction of qIND-qCPA and qIND-qCCA1 PKE

Gagliardoni et al. [GKS21] proved, that such a hybrid scheme is qIND-qCPA secure (in their notion) if the SKE is qIND (cf. [GHS16]) and the PKE is *post-quantum* CPA (IND-pqCPA). The qIND security notion for SKE schemes works as follows: the adversary is given type-2 oracle access to encryption, there is no learning phase, and the challenge – potentially in superposition – is of the form LoR. We explicitly define this notion in Appendix B (cf. Def. 32). On the other hand, IND-pqCPA essentially corresponds to Def. 7 (and Fig. 3) except that the challenge is restricted to be classical.

It turns out that these results also applies to our *general* qIND-qCPA security notion. Furthermore, we can similarly construct a qIND-qCCA1 secure PKE scheme allowing us to generalize the claim to "a hybrid scheme is qIND-qATK secure (in our notion) if the PKE scheme is IND-pqATK and the SKE scheme is qIND", *where* ATK *is either* CPA *or* CCA1. Again, IND-pqATK refers to the post-quantum ATK security notion, i.e., any decryption query is restricted to be classical, and the *classical* challenge is of the form LoR. Note that this is sufficient, since the asymmetric scheme is only used to encrypt the *classical* one-time symmetric key. In particular, we do not require the stronger notion IND-qATK of Boneh and Zhandry (where the adversary has superposition access to the decryption oracle).

Theorem 30 (qIND-{qCPA,qCCA1} Security of Hybrid Encryption) *Let Π be the hybrid encryption scheme obtained from a* PKE *scheme P and a* SKE *scheme S (cf. Fig. 22) and let* ATK *be either* CPA *or* CCA1. *For any* qIND-qATK *adversary $\mathcal{A}$ against the purification $\Pi^{\diamondsuit}$, there exists adversaries $\mathcal{A}^P$ and $\mathcal{A}^S$ against P and S, respectively, such that*

$$\mathbf{Adv}^{\text{qIND-qATK}}_{\Pi}(\mathcal{A}) \leq \mathbf{Adv}^{\text{IND-pqATK}}_{P}(\mathcal{A}^P) + \mathbf{Adv}^{\text{qIND}}_{S}(\mathcal{A}^S).$$

Proof. Let $\mathcal{A}$ be the adversary playing the qIND-qATK. We prove the claim via game hops:

Game 0: G_0 is just qIND-qATK (cf. Fig. 18).

Game 1: G_1 is as G_0, but the challenger $\mathfrak{C}$ "blinds" the asymmetric part c_p of the challenge ciphertext, replacing it with an encryption of a randomly chosen k' rather than k. This substitution is undetectable for $\mathcal{A}$. Otherwise we could build a reduction $\mathcal{A}^P$ against the IND-pqATK security of P as follows:

1. $\mathcal{A}^P$ plays the IND-pqATK game against $\mathfrak{C}^{\text{IND-pqATK}}$ and simulates $\mathfrak{C}$ for $\mathcal{A}$. $\mathcal{A}^P$ flips a bit $b \leftarrow^{\$} \{0,1\}$, chooses a random pseudo-random permutation π and implements in unitaries $U^{\pi}, U^{\pi^{-1}}$. The public key pk, $\mathcal{A}^P$ receives from $\mathfrak{C}^{\text{IND-pqATK}}$, she forwards to $\mathcal{A}$.
2. For every encryption query μ of $\mathcal{A}$, $\mathcal{A}^P$ proceeds as follows: First, she samples a (classical) one-time key $\mathsf{k} \leftarrow \mathtt{KGen}^S(\lambda; r_1)$ (for some r_1) and encrypts the key to $\chi_P \leftarrow \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}; r_3)$ (for some r_3). If available, she sends a copy of χ_P – this is possible, as χ_P is a classical value – to her (classical) decryption oracle in order to compute the corresponding error value $\varepsilon_P \leftarrow \mathsf{k} \oplus O_{\mathtt{Dec}^P}(\chi_P)$. Otherwise, she guesses $\varepsilon_P \leftarrow |0\rangle\langle 0|$. Next, she encrypts μ (using k) to $\chi_S \leftarrow \mathtt{Enc}^S_{\mathsf{k}}(\mu; r_2)$ (for some r_2) and computes the corresponding error value $\varepsilon \leftarrow \mu \oplus \mathtt{Dec}^S_{\mathsf{k}\oplus\varepsilon_P}(\mathtt{Enc}^S_{\mathsf{k}}(\mu; r_2))$. Finally $\mathcal{A}^P$ sends $\chi_S\chi_P\varepsilon\upsilon$ to $\mathcal{A}$, where $\upsilon \leftarrow U^{\pi}\,|r_1||r_2||r_3\rangle\langle r_1||r_2||r_3|$.
3. If ATK = CCA1, $\mathcal{A}$ is allowed to send decryption queries to $\mathcal{A}^P$. Note, that any classical decryption query is easy to decrypt. For every quantum decryption query $\chi\varepsilon\upsilon$ of $\mathcal{A}$, $\mathcal{A}^P$ proceeds as follows: She parses χ as $\chi_S\chi_P$. Next, she measures the asymmetric ciphertext component $c_P \leftarrow \mathfrak{M}(\chi_P)$ in the computational basis[14], sends c_P to her (classi-

[14] Again, by Lemma 29, this does not change $\mathcal{A}$'s point of view.

cal) decryption oracle to receive the one-time key k and decrypts χ_S to $\mu \leftarrow \mathtt{Dec}^S_{\mathsf{k}}(\chi_S) \oplus \varepsilon$. Finally $\mathcal{A}^P$ sends $\mu\upsilon$ to $\mathcal{A}$.

4. Once $\mathcal{A}^P$ receives the challenge μ from $\mathcal{A}$, she samples two fresh (classical) one-time keys k (with some r_1) and k' and sends $(\mathsf{k}, \mathsf{k}')$ as challenge to $\mathfrak{C}^{\text{IND-pqATK}}$ to obtain the challenge ciphertext χ_P. Note, that χ_P is either the encryption of k (if $b_{\mathfrak{C}} = 0$) or k' (if $b_{\mathfrak{C}} = 1$), depending on $\mathfrak{C}^{\text{IND-pqATK}}$'s bit $b_{\mathfrak{C}}$. $\mathcal{A}^P$ obtains χ_S via encryption of either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) under S using the key k and some r_2. The corresponding tag $\varepsilon\upsilon$ to $\chi_S\,\chi_P$ she computes as before (2.) except she guesses the unknown randomness $r_3 \leftarrow^{\$} \mathcal{R}_3$ and the (intermediate) error value $\varepsilon_P \leftarrow |0\rangle\langle 0|$. Then, she sends $\chi_S\,\chi_P\,\varepsilon\upsilon$ to $\mathcal{A}$.
5. For every encryption query of $\mathcal{A}$ in the post-challenge phase, $\mathcal{A}^P$ proceeds as in (2.).
6. If $\mathcal{A}$ guesses b correctly, then $\mathcal{A}^P$ outputs $b_{\mathfrak{C}} = 0$, otherwise $b_{\mathfrak{C}} = 1$.

If ATK = CCA1, it holds that $\mathcal{A}^P$ (almost) perfectly simulates G_0 (if $b_{\mathfrak{C}} = 0$) or G_1 (if $b_{\mathfrak{C}} = 1$), depending on her own challenge. The critical part of the simulation is that $\mathcal{A}^P$ guesses parts of the tag. However, the guessed error ε_P was guessed correctly with overwhelming probability and the randomness register υ with guessed randomness r_3 is indistinguishable from υ where r_3 is replaced with the *used randomness*, since π is a pseudo-random permutation. Therefore, we have

$$|\Pr[\mathcal{A} \text{ wins } \mathsf{G}_0] - \frac{1}{2}| \leq |\Pr[\mathcal{A} \text{ wins } \mathsf{G}_1] - \frac{1}{2}| + \underbrace{\mathbf{Adv}_P^{\text{IND-pqATK}}(\mathcal{A}^P)}_{=\text{negl}(\lambda)}.$$

Next, we prove that $\mathcal{A}$ cannot win G_1 with non-negligible advantage over guessing. Otherwise we could build a reduction $\mathcal{A}^S$ against the qIND security of S as follows:

1. $\mathcal{A}^S$ plays the qIND game against $\mathfrak{C}^{\text{qIND}}$ and simulates $\mathfrak{C}$ for $\mathcal{A}$. $\mathcal{A}^S$ chooses a random pseudo-random permutation π, implements in unitaries $U^{\pi}, U^{\pi^{-1}}$, generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}(\lambda)$ and forwards pk to $\mathcal{A}$.
2. For any encryption and decryption query, $\mathcal{A}^S$ implements a purified encryption oracle and – if ATK = CCA1 – a purified decryption and an additional classical oracle for Π and computes everything locally. In particular, the challenger $\mathfrak{C}^{\text{qIND}}$ is not involved in the pre-challenge phase.
3. Once $\mathcal{A}$ sends a challenge register μ, adversary $\mathcal{A}^S$ forwards $(\mu, \tau_{\mathcal{M}})$ as her own challenge to $\mathfrak{C}^{\text{qIND}}$ to obtain χ_S. Next, she samples a fresh (classical) one-time key k', encrypts it to $\chi_P \leftarrow \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}'; r_3)$ (for some r_3). She takes both ciphertext, guesses the unknown parts of the tag – $r_1 \leftarrow^{\$} \mathcal{R}_1$, $r_2 \leftarrow^{\$} \mathcal{R}_2$ and $\varepsilon \leftarrow |0\ldots0\rangle\langle 0\ldots0|$ – , and sends $\chi_S\,\chi_P\,\varepsilon U^{\pi}\,|r\rangle\langle r|$ to $\mathcal{A}$, where $r = r_1||r_2||r_3$.
4. For every encryption query of $\mathcal{A}$ in the post-challenge phase, $\mathcal{A}^S$ proceeds as in (2.).
5. When $\mathcal{A}$ outputs her guess, $\mathcal{A}^S$ forwards it as her own output.

It holds that $\mathcal{A}^S$ (almost) perfectly simulates game G_1 for $\mathcal{A}$, with the same challenge bit as $\mathfrak{C}^{\text{qIND}}$. The critical part of the simulation is that $\mathcal{A}^S$ guesses parts of the tag. However, the *guessed error* ε was guessed correctly with overwhelming probability and the "randomness register" $\upsilon = U^\pi |r_1||r_2||r_3\rangle\langle r_1||r_2||r_3|$ with *guessed randomness* r_1 *and* r_2 is indistinguishable from υ where r_1, r_2 are replaced with the *used randomness*, since π is a pseudo-random permutation. Thus,

$$|\Pr[\mathcal{A} \text{ wins } \mathsf{G}_1] - \frac{1}{2}| \leq \mathbf{Adv}_S^{\text{qIND}}(\mathcal{A}^S) \leq \text{negl}(\lambda).$$

Collecting the bounds above proves the statement. □

A.2 Construction of qIND-qCCA2 PKE

Following a similar approach as [AGM21], we prove qIND-qCCA2 security of the hybrid scheme under suitable properties of the underlying PKE and SKE components, by transforming the Test game (cf. Fig. 19) to the Fake game (cf. Fig. 20) via game hops. We require the PKE to be IND-pqCCA2 secure, and the SKE to fulfill a suitable property of *quantum authentication*. For the latter, there are different possible choices (for example, in [AGM21], the Q3 notion of *one-time ciphertext authentication* is used). To make things easier for us, because it is conceptually simpler, we introduce a stronger notion for SKE: *quantum authenticated encryption* (qAE), which is a 'natural' type-2 translation to the Q2 realm of the Q3 notion of QAE introduced by [AGM21]. We explicitly define this notion in Appendix B (cf. Def. 35). In a nutshell: qAE states that it should not be possible, for any QPT adversary, to distinguish between a *Real* world (where she has access to type-2 encryption and decryption oracles), and an *Ideal* world (where the encryption oracle always ignores the query and encrypts half of an EPR pair, while the decryption oracle always returns $\perp$, except on non-fresh ciphertexts which are managed through the usual detection replay techniques).

Theorem 31 (qIND-qCCA2 Security of Hybrid Encryption) *Let Π be the hybrid encryption obtained from a* PKE *scheme P and a* SKE *scheme S (cf. Fig. 22). For any* qIND-qCCA2 *adversary $\mathcal{A}$ against the purification $\Pi^\diamondsuit$, there exists adversaries $\mathcal{A}^P$ and $\mathcal{A}^S$ against P and S respectively, such that:*

$$\mathbf{Adv}_\Pi^{\text{qIND-qCCA2}}(\mathcal{A}) \leq \mathbf{Adv}_P^{\text{IND-pqCCA2}}(\mathcal{A}^P) + \mathbf{Adv}_S^{\text{qAE}}(\mathcal{A}^S).$$

Proof. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be the adversary playing the qCCA2-Test game against Π. We do a game hop leading from qCCA2-Test to qCCA2-Fake.

Game 0: G_0 is just qCCA2-Test (cf. Fig. 19).

Game 1: G_1 is as G_0, but the challenger $\mathfrak{C}$ "blinds" the asymmetric part (c_P) of the challenge ciphertext, which is the encrypted classical one-time key that was used to encrypt the plaintext under S, replacing it with a different

(classical) one (cf. Fig. 24). Furthermore, we modify the decryption oracle in order to undo the blinding whenever queried on the "fake" (replaced) ciphertext, making the substitution undetectable to the adversary, i.e., whenever the adversary sends a decryption query with the fake asymmetric part, the challenger substitutes it with the correct one (cf. Fig. 25).
This substitution is possible as, for the hybrid encryption scheme (cf. Fig. 22), measuring vs. not-measuring the asymmetric ciphertext component (χ_P) of χ – which is not given to $\mathcal{A}$ – is indistinguishable for $\mathcal{A}$ (cf. Lemma 29), but allows $\mathfrak{C}$ to reliably detect the fake asymmetric part. As the measurement is undetectable, this measurement does not disturb $\mathcal{A}$'s behavior. Moreover, the substitution of χ_P with a different χ'_P is undetectable for $\mathcal{A}$. Otherwise we could build a reduction $\mathcal{A}^P$ against the IND-pqCCA2 security of P as follows:

1. $\mathcal{A}^P$ plays the IND-pqCCA2 game against $\mathfrak{C}^{\text{IND-pqCCA2}}$ and simulates $\mathfrak{C}$ for $\mathcal{A}$. $\mathcal{A}^P$ flips a bit $b \leftarrow^{\$} \{0,1\}$, chooses a random pseudo-random permutation π and implements in unitaries $U^\pi, U^{\pi^{-1}}$. The public key pk, $\mathcal{A}^P$ receives from $\mathfrak{C}^{\text{IND-pqCCA2}}$, she forwards to $\mathcal{A}$.
2. For every encryption query μ of $\mathcal{A}$, $\mathcal{A}^P$ proceeds as follows: First, she samples a (classical) one-time key $\mathsf{k} \leftarrow \mathtt{KGen}^S(\lambda; r_1)$ (for some r_1) and encrypts the key to $\chi_P \leftarrow \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k}; r_3)$ (for some r_3). She sends a copy of χ_P – this is possible, as χ_P is a classical value – to her (classical) decryption oracle in order to compute the corresponding error value $\varepsilon_P \leftarrow \mathsf{k} \oplus O_{\mathtt{Dec}^P}(\chi_P)$. Next, she encrypts μ (using k) to $\chi_S \leftarrow \mathtt{Enc}^S_{\mathsf{k}}(\mu; r_2)$ (for some r_2) and computes the corresponding error value $\varepsilon \leftarrow \mu \oplus \mathtt{Dec}^S_{\mathsf{k}\oplus\varepsilon_P}(\mathtt{Enc}^S_{\mathsf{k}}(\mu; r_2))$. Finally $\mathcal{A}^P$ sends $\chi_S\,\chi_P\,\varepsilon\upsilon$ to $\mathcal{A}$, where $\upsilon \leftarrow U^\pi\,|r_1||r_2||r_3\rangle\langle r_1||r_2||r_3|$.
3. For every classical decryption query c, it is easy for $\mathcal{A}^P$ to decrypt the ciphertext. For every quantum decryption query $\chi\varepsilon\upsilon$ of $\mathcal{A}$, $\mathcal{A}^P$ proceeds as follows: She parses χ as $\chi_S\,\chi_P$. Next, she measures the asymmetric ciphertext component $c_P \leftarrow \mathfrak{M}(\chi_P)$ in the computational basis[15], sends c_P to her (classical) decryption oracle to receive the one-time key k and decrypts χ_S to $\mu \leftarrow \mathtt{Dec}^S_{\mathsf{k}}(\chi_S) \oplus \varepsilon$. Finally $\mathcal{A}^P$ sends $\mu\upsilon$ to $\mathcal{A}$.
4. Once $\mathcal{A}^P$ receives the challenge from $\mathcal{A}$, she generates two fresh (classical) one-time keys k and k', and sends $(\mathsf{k}, \mathsf{k}')$ as challenge to $\mathfrak{C}^{\text{IND-pqCCA2}}$. The response ($\chi_P$) is either the encryption of k (if $b_{\mathfrak{C}} = 0$) or k' (if $b_{\mathfrak{C}} = 1$), depending on the $\mathfrak{C}^{\text{IND-pqCCA2}}$'s challenge bit $b_{\mathfrak{C}}$. Anyway, she stores a copy of χ_P – again, this is possible, as χ_P is a classical value.
5. $\mathcal{A}^P$ obtains χ_S via encryption of either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) under S using the key k and some randomness r_2. The corresponding tag $\varepsilon\upsilon$ to $\chi_S\,\chi_P$ she computes as before (2.) except she guesses the unknown randomness $r_3 \leftarrow^{\$} \mathcal{R}_3$ and the (intermediate) error value $\varepsilon_P \leftarrow |0\rangle\langle 0|$. Then, she sends $\chi_P\,\chi_S\,\varepsilon\upsilon$ to $\mathcal{A}$.
6. After the challenge phase, she simulates any encryption query as previous and any decryption query as follows: if the corresponding asymmetric

[15] Again, by Lemma 29, this does not change $\mathcal{A}$'s point of view.

Game $\mathsf{G}_1(\Pi^{\Diamond}, \mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2), \lambda)$

1: $\mathfrak{C}$ runs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}^{\Diamond}$ and flips a coin $b \leftarrow^{\$} \{0,1\}$
2: $\mathfrak{C}$ chooses a random pseudo-random permutation π
3: $\mathcal{A}_1^{O^{\Diamond}_{\mathtt{Enc},\pi}, O^{\Diamond}_{\mathtt{Dec},\pi}, O_{\pi}, \mathtt{Dec}_{\mathsf{sk}}}$ receives pk and λ, prepares a memory state σ, and sends to $\mathfrak{C}$ a challenge state μ
4: $\mathfrak{C}$ prepares fresh (classical) randomness r, encrypts through $O^{\Diamond}_{\mathtt{Enc},\pi}$ (using r) either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) and places the result in $\chi\varepsilon\upsilon$
5: $\mathfrak{C}$ records the asymmetric part χ_P (the encrypted one-time key k) of χ
6: $\mathfrak{C}$ generates a fresh one-time key $\mathsf{k}' \leftarrow \mathtt{KGen}^{S}(\lambda; r_1')$, encrypts $\chi_P' \leftarrow \mathtt{Enc}^{P}_{\mathsf{pk}}(\mathsf{k}'; r_2')$ and replaces the classical χ_P with χ_P' in $\chi\varepsilon\upsilon$
7: $\mathcal{A}_2^{O^{\Diamond}_{\mathtt{Enc},\pi}, D_{|\mathsf{fake}\rangle}, O_{\pi}, D_{\mathsf{fake}}}$ receives registers $\chi\varepsilon\upsilon$ and σ, where $D_{|\mathsf{fake}\rangle}$ and D_{fake} are defined as in Fig. 25
8: $\mathcal{A}_2$ outputs a bit b'
9: **if** $b = b'$, **then return** win; **else return** reject

Fig. 24: The G_1 experiment.

$D_{|\mathsf{fake}\rangle}$ on input $(\chi\varepsilon\upsilon) \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$

1: Measure the asymmetric subsystem (χ_P) of χ
2: **if** outcome is χ_P', **then**
3: Replace the asymmetric subsystem of χ with $|\chi_P\rangle\langle\chi_P|$
4: Apply $O^{\Diamond}_{\mathtt{Dec},\pi}$ on input $(\chi\varepsilon\upsilon)$

D_{fake} on input $c \in \mathcal{C}$

1: Parse c as $c_S || c_P$, where c_P is the asymmetric part
2: **if** $c_P = \chi_P'$, **then** // Recall, that χ_P' is the encrypted one-time key and, thus, classical.
3: Replace the asymmetric part of c with χ_P
4: Apply $\mathtt{Dec}_{\mathsf{sk}}$ on input c

Fig. 25: The modified decryption oracle $D_{|\mathsf{fake}\rangle}$ and D_{fake} in G_1. Note that, by Lemma 29, the measurement in line 1 of $D_{|\mathsf{fake}\rangle}$ does not change the adversaries point of view.

$D'_{|\mathsf{fake}\rangle}$ and $D''_{|\mathsf{fake}\rangle}$ on input $(\chi\varepsilon\upsilon) \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$

1: Measure the asymmetric subsystem of χ
2: **if** outcome is χ'_P, **then**
3: Replace the asymmetric subsystem of χ with $|\chi_P\rangle\langle\chi_P|$
4: Compute $\mu\rho\zeta \leftarrow \mathcal{U}_{\mathsf{Dec}^\diamond}(\chi\varepsilon(U^{\pi^{-1}}\upsilon) \otimes |0\dots0\rangle\langle 0\dots0|)$,
where $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$, $\rho \in \mathfrak{D}(\mathfrak{H}_{\mathcal{R}})$ and $\zeta \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}})$
5: Measure ζ in the computational basis
6: **if** outcome is not $0\dots0$, **then** Send $|\bot\rangle\langle\bot|$ to $\mathcal{A}_2$
7: **else do**
8: Measure ρ in the computational basis
9: **if** outcome is r, **then**
10: Apply $\{\Pi^+, \mathbb{1} - \Pi^+\}$ to $\mu\mu''$
11: **if** outcome is 0 **then**
12: Abort game and **return** win // replay detected
13: Send $\mu \otimes |\pi(r)\rangle\langle\pi(r)|$ to $\mathcal{A}_2$

D'_{fake} and D''_{fake} on input $c \in \mathcal{C}$

1: Parse c as $c_S||c_P$, where c_P is the asymmetric part
2: **if** $c_P = \chi'_P$, **then**
3: Replace the asymmetric part of c with χ_P
4: Prepare $\chi \coloneqq |c||0\dots0||\pi(r)\rangle\langle c||0\dots0||\pi(r)|$, and query $D'_{|\mathsf{fake}\rangle}(\chi)$ resp. $D''_{|\mathsf{fake}\rangle}(\chi)$
// If challenge ciphertext was measured and then replayed, this step will detect cheating.
5: Discard response of $D'_{|\mathsf{fake}\rangle}(\chi)$ resp. $D''_{|\mathsf{fake}\rangle}(\chi)$
6: Send $\mathsf{Dec}_{\mathsf{sk}}(c)$ to $\mathcal{A}_2$

Fig. 26: The modified decryption oracles $D'_{|\mathsf{fake}\rangle}$ and D'_{fake} in G_4 and $D''_{|\mathsf{fake}\rangle}$ and D''_{fake} in G_5. Note that, by Lemma 29, the measurement in line 1 of each of the quantum oracle does not change the adversaries point of view.

part is χ_P, she defines the decryption as k. Otherwise she sends the asymmetric part (which is $\neq \chi_P$) to her decryption oracle to obtain the used symmetric one-time key. In both cases, she decrypts c_S resp. χ_S with the obtained key by herself.

4. If $\mathcal{A}$ guesses b correctly, then $\mathcal{A}^P$ outputs $b_{\mathfrak{C}} = 0$, otherwise $b_{\mathfrak{C}} = 1$.

If $b_{\mathfrak{C}} = 1$, then $\mathcal{A}^P$ does with its simulation of decryption queries (after the challenge phase) the same as the modified decryption oracle. In particular, both ($\mathcal{A}^P$, $D_{|\mathsf{fake}\rangle}$ and D_{fake}, cf. Fig. 25) takes the asymmetric part χ_P – which is the encryption of k' – and defines it to be the encryption of k. In case $b_{\mathfrak{C}} = 0$, the reduction (almost) perfectly simulates G_0 or G_1 – in that case χ_P is actually the encryption of k. The critical part of the simulation is that – for the quantum decryption queries – $\mathcal{A}^P$ guesses parts of the tag. However, the guessed error ε_P was guessed correctly with overwhelming probability and the randomness register v with guessed randomness r_3 is indistinguishable from v where r_3 is replaced with the *used randomness*, since π is a pseudo-random permutation. Therefore, we have

$$\Pr[\mathcal{A} \text{ wins } \mathsf{G}_0] \leq \Pr[\mathcal{A} \text{ wins } \mathsf{G}_1] + \underbrace{\mathbf{Adv}_P^{\text{IND-pqCCA2}}(\mathcal{A}^P)}_{=\text{negl}(\lambda)}.$$

Game 2 G_2 is as G_1, but this time $\mathfrak{C}$ *always* replaces the challenge plaintext register μ with the fully mixed state $\tau_{\mathcal{M}}$, regardless of a bit b. If an adversary playing G_2 terminates and outputs a bit, the game G_2 terminates as well outputting either win or reject (with probability $\frac{1}{2}$).
Clearly, if $b = 1$ nothing changes for $\mathcal{A}$. If $b = 0$ instead, the symmetric part of the challenge ciphertext received back by $\mathcal{A}$ is now an encryption (through $\mathtt{Enc}_{\mathsf{k}}^S$) of $\tau_{\mathcal{M}}$ for a secret key k unknown to $\mathcal{A}$ by the reasoning made in G_1. Therefore the substitution of μ with $\tau_{\mathcal{M}}$ is undetectable for $\mathcal{A}$. Otherwise we could build a reduction $\mathcal{A}^S$ against the qAE security (cf. Def. 35) of S as follows:

1. $\mathcal{A}^S$ plays the qAE game against $\mathfrak{C}^{\text{qAE}}$ and simulates $\mathfrak{C}$ for $\mathcal{A}$. $\mathcal{A}^S$ chooses a random pseudo-random permutation π, implements in unitaries $U^{\pi}, U^{\pi^{-1}}$, generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}(\lambda)$ and forwards pk to $\mathcal{A}$.
2. For any encryption and decryption query, $\mathcal{A}^S$ implements a purified encryption and decryption oracle, and an additional classical decryption oracle for Π and computes everything locally. In particular, the challenger $\mathfrak{C}^{\text{qAE}}$ is not involved in the pre-challenge phase.
3. Once $\mathcal{A}^S$ receives the challenge μ from $\mathcal{A}$, she flips a coin $b \leftarrow^{\$} \{0,1\}$ and sends either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$) to her encryption oracle. The response (χ_S) is either the encryption of the sent message (if $b_{\mathfrak{C}} = 0$) or the encryption of μ' (if $b_{\mathfrak{C}} = 1$) – half of an maximally entangled register $|\varphi^+\rangle_{\mu'\mu''}$ prepared by $\mathfrak{C}^{qAE}$ – , depending on $\mathfrak{C}^{\text{qAE}}$'s challenge bit $b_{\mathfrak{C}}$. If $b_{\mathfrak{C}} = 0$, then the challenger runs the qAE-*Real* experiment (cf. Fig. 28) and, otherwise, the qAE-*Ideal* (cf. Fig. 29) experiment.
3. Next, she samples a fresh (classical) one-time key k', encrypts it to $\chi_P \leftarrow \mathtt{Enc}_{\mathsf{pk}}^P(\mathsf{k}'; r_3)$ (for some r_3). Again, she stores a copy of χ_P – this

is possible, as χ_P is a classical value. She guesses the unknown parts of the tag – $r_1 \leftarrow^{\$} \mathcal{R}_1$, $r_2 \leftarrow^{\$} \mathcal{R}_2$ and $\varepsilon \leftarrow |0\dots0\rangle\langle 0\dots0|$ – , and sends $\chi_S\, \chi_P\, \varepsilon(U^\pi\, |r\rangle\langle r|)$ to $\mathcal{A}$, where $r = r_1||r_2||r_3$.

4. After the challenge phase, she simulates any encryption query as previous and any classical resp. quantum decryption query as follows: if the corresponding asymmetric part – which she obtains for quantum queries by simply measuring the asymmetric part of the ciphertext[16] – equals the stored fake ciphertext χ_P, she sends c_s resp. χ_S to her decryption oracle. Otherwise, she decrypts the queried ciphertext by herself.
5. If $\mathcal{A}$ guesses b correctly, then $\mathcal{A}^P$ outputs $b_{\mathfrak{C}} = 0$, otherwise she outputs $b_{\mathfrak{C}} = 1$.

Note that the reduction (almost) perfectly simulates G_1 and G_2, since $\mathcal{A}$ cannot distinguish $\tau_{\mathcal{M}}$ from μ' without access to μ''. The critical part of the simulation is that $\mathcal{A}^P$ guesses parts of the tag. However, this is not detectable for the adversary for the same reason as in the previous game hop. Therefore it follows:

$$\Pr[\mathcal{A} \text{ wins } \mathsf{G}_1] \leq \Pr[\mathcal{A} \text{ wins } \mathsf{G}_2] + \underbrace{\mathbf{Adv}_S^{\mathrm{qAE}}(\mathcal{A}^S)}_{=\mathrm{negl}(\lambda)}.$$

Game 3: G_3 is as G_2, except that $\mathfrak{C}$ prepares two entangled registers $|\varphi^+\rangle_{\mu'\mu''}$. Then $\mathfrak{C}$ returns the encryption of μ' (instead of $\tau_{\mathcal{M}}$) to $\mathcal{A}$. Clearly the winning probability of $\mathcal{A}$ is unaffected by this, otherwise $\mathcal{A}$ could distinguish $\tau_{\mathcal{M}}$ from μ' without access to μ'':

$$\Pr[\mathcal{A} \text{ wins } \mathsf{G}_3] = \Pr[\mathcal{A} \text{ wins } \mathsf{G}_2].$$

Game 4: G_4 is as G_3, except that $\mathfrak{C}$ stores μ'' and the randomness r, which was used to encrypt μ'. Furthermore, $\mathcal{A}_2$ gets access to $D'_{|\mathsf{fake}\rangle}$ and D'_{fake} (cf. Fig. 26).

The difference compared to the previous oracles is that the new oracles aborts with win, whenever it detects a challenge plaintext replay. Clearly, the winning probability of $\mathcal{A}$ does not decrease in respect to the previous game, hence so far we have:

$$\Pr[\mathcal{A} \text{ wins } \mathsf{G}_3] \leq \Pr[\mathcal{A} \text{ wins } \mathsf{G}_4].$$

Game 5 This is as G_4, but we remove the "χ_P-blinding" (line 6 in Fig. 24). Furthermore, $\mathcal{A}_2$ gets access to other decryption oracles $D''_{|\mathsf{fake}\rangle}$ and D''_{fake} where we removed the "χ_P-unblinding" (cf. Fig. 26). Note, that this decryption oracles $D''_{|\mathsf{fake}\rangle}$ and D''_{fake} are the same as $D_{|\mathsf{fake}\rangle}$ and D_{fake} in the qCCA2-Fake game (cf. Fig. 21), except it aborts (in line 12 of $D''_{|\mathsf{fake}\rangle}$) with win instead of cheat whenever it detects a challenge plaintext replay. This replacement is undetectable for the same reasoning as in G_1.

Game 6 This is as G_5, except that:

[16] Again, by Lemma 29, this does not change $\mathcal{A}$'s point of view.

1. $D''_{|\mathsf{fake}\rangle}$ and D''_{fake} are replaced by $D^{\Diamond}_{|\mathsf{fake}\rangle}$ and $D^{\Diamond}_{\mathsf{fake}}$, i.e., aborts with cheat instead of win.
2. $\mathfrak{C}$ does not flip the random bit b.
3. As soon as $\mathcal{A}$ terminates, $\mathfrak{C}$ outputs cheat or reject with probability $\frac{1}{2}$.

This final game is equivalent to qCCA2-Fake. In fact, by now $\mathcal{A}$'s output is completely unrelated to b. Combining all inequalities, we have shown:

$$\begin{aligned}\mathbf{Adv}^{\text{qIND-qCCA2}}_{\Pi}(\mathcal{A}) &= \Pr[\mathcal{A} \text{ wins qCCA2-Test}] \\ &\quad - \Pr[\mathcal{A} \text{ cheats in qCCA2-Fake}] \\ &\leq \mathbf{Adv}^{\text{IND-pqCCA2}}_{P}(\mathcal{A}^P) + \mathbf{Adv}^{\text{qAE}}_{S}(\mathcal{A}^S) \leq \text{negl}(\lambda),\end{aligned}$$

which proves the statement. □

B Q2 Security of SKE

In this section we look at Q2 notions of security for SKE. Let us first note that, since we only consider SKE schemes, we can treat encryption and decryption as perfectly correct[17], and therefore the implementation of the $O^{(2)}_{\mathtt{Enc}^S}$ and $O^{(2)}_{\mathtt{Dec}^S}$ oracles for a SKE scheme $S = (\mathtt{KGen}^S, \mathtt{Enc}^S, \mathtt{Dec}^S)$ follows the usual blueprint – notice, they are exactly the inverse of each other, as here we only have a single symmetric key k.

But we can say more: the underlying unitary $U^{(2)}_{\mathtt{Enc}^S}$, when applied to arbitrary states in $\mathfrak{D}(\mathfrak{H}_{\mathcal{M}} \otimes \mathfrak{H}_{\mathcal{R}})$, behaves like a *symmetric key quantum encryption scheme (SKQE)* in the Q3 sense, as defined in [ABF+16]; accordingly, we recall a characterization of SKQE from [AM17], which states that *any* encryption operator in a SKQE scheme can be decomposed in the following form: first, attach a (possibly key-dependent) auxiliary state to the plaintext state, and then apply a unitary – this characterization is a consequence of correctness only, so it applies to all SKQE. In our case (because we are in the Q2 case built on top of classical SKE schemes), this decomposition is always efficient: the 'auxiliary state' is just the (classical) randomness, plus (optionally) a (classical) authentication tag[18], while the 'unitary' is $U^{(2)}_{\mathtt{Enc}^S}$. This implies that the inverse operator, $U^{(2)}_{\mathtt{Dec}^S}$, when applied to a correctly generated ciphertext, produces a state which we can see as a separable product of: the original quantum plaintext, and the used randomness encoded as a computational basis element.

Then, let us first recall the *one-time quantum indistinguishability* (qIND) notion for classical SKE from [GHS16] (where it was denoted as *gqIND*). This was originally given in the *Left-or-Right* flavour, but we use here an equivalent *Real-or-Random* formulation:

[17] We do not consider 'exotic' SKE schemes which are not perfectly correct.

[18] To be more precise: the 'auxiliary state' could be a (classical) *function* of the (classical) randomness and the key; for example, some bits of the original randomness could be discarded. We ignore this aspect here, as it would not change the situation for us, adding just the necessity for extra bookkeeping.

qIND$(S, \mathcal{A}, \lambda)$

1: $\mathfrak{C}$ runs $\mathsf{k} \leftarrow \mathtt{KGen}^S$, and flips a coin $b \leftarrow^{\$} \{0,1\}$
2: $\mathcal{A}$ receives λ, and sends to $\mathfrak{C}$ a challenge state μ
3: $\mathfrak{C}$ encrypts through $O^{(2)}_{\mathtt{Enc}^S}$ either μ (if $b = 0$) or $\tau_{\mathcal{M}}$ (if $b = 1$), and sends the response χ to $\mathcal{A}$
4: $\mathcal{A}$ receives state χ and outputs a bit b'
5: **if** $b = b'$, **then return** win; **else return** reject

Fig. 27: The qIND experiment (RoR version) for a classical SKE scheme S.

Definition 32 (qIND Security for SKE) *An* SKE *scheme S has* one-time ciphertext quantum indistinguishability*, or is qIND-secure, iff the advantage of any QPT adversary $\mathcal{A}$ to win* qIND$(S, \mathcal{A}, \lambda)$ *(cf. Fig. 27) is negligible, i.e.,*

$$\mathbf{Adv}^{\mathrm{qIND}}_S(\mathcal{A}) \coloneqq \left|\Pr[\mathrm{qIND}(S, \mathcal{A}, \lambda) \to \mathsf{win}] - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda).$$

Stepping up to Q2 notions of *ciphertext authentication* for SKE, the situation gets more complex. For defining a Q2 notion of authenticated encryption, we translate in a natural way the Q3 notion of *quantum authenticated encryption* (QAE) defined in [AGM18] for quantum SKE. In so doing, we define two experiments (cf. [Shr04]): One (the *Real* world) allows the adversary to query $O^{(2)}_{\mathtt{Enc}^S}$ and $O^{(2)}_{\mathtt{Dec}^S}$ freely; in the other (the *Ideal* world) instead, we replace the oracles with: a fake encryption oracle which always ignores the input and instead encrypts a random state, and a fake decryption oracle which always rejects with $\perp$ except on replayed ciphertexts (cf. Fig. 30). Distinction of replayed vs. non-replayed ciphertexts is done through the usual entanglement-based replay detection mechanism. Security requires that no QPT adversary can reliably distinguish the two worlds.

Experiment 33 (qAE Real) *The experiment* qAE-Real$(S, \mathcal{A}, \lambda)$ *for an SKE scheme $S = (\mathtt{KGen}^S, \mathtt{Enc}^S, \mathtt{Dec}^S)$ is defined in Fig. 28.*

qAE-Real$(S, \mathcal{A}, \lambda)$

1: $\mathfrak{C}$ runs $\mathsf{k} \leftarrow \mathtt{KGen}^S$ and implements oracles $O^{(2)}_{\mathtt{Enc}^S}$ and $O^{(2)}_{\mathtt{Dec}^S}$
2: $\mathcal{A}^{O^{(2)}_{\mathtt{Enc}^S}, O^{(2)}_{\mathtt{Dec}^S}}$ receives λ, and outputs a bit b
3: **return** b

Fig. 28: The qAE real experiment for a classical SKE scheme S.

Experiment 34 (qAE Ideal) *The experiment* qAE-Ideal$(S, \mathcal{A}, \lambda)$ *for an SKE scheme $S = (\mathtt{KGen}^S, \mathtt{Enc}^S, \mathtt{Dec}^S)$ is defined in Fig. 29.*

Having the Real and Ideal games, we can finally define qAE security for SKE:

qAE-Ideal$(S, \mathcal{A}, \lambda)$

1: $\mathfrak{C}$ runs $\mathtt{k} \leftarrow \mathtt{KGen}^S$ and implements a private database Λ (initially empty) and oracles O_E and O_D as in Fig. 30
2: $\mathcal{A}^{O_E, O_D}$ receives λ, and outputs a bit b
3: **return** b

Fig. 29: The qAE ideal experiment for a classical SKE scheme S.

O_E on input $\mu \in \mathfrak{D}(\mathfrak{H}_{\mathcal{M}})$

1: Prepare EPR state $|\varphi^+\rangle_{\mu'\mu''}$ and sample fresh randomness r
2: Store (r, μ, μ'') in the database Λ
3: Send $\chi \leftarrow O^{(2)}_{\mathtt{Enc}^S}(\mu' \otimes |r\rangle\langle r|)$ to $\mathcal{A}$

O_D on input $\chi \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}})$

1: Apply $O^{(2)}_{\mathtt{Dec}^S}$ to χ and perform a partial measurement of the result, obtaining a quantum register μ' and a classical randomness r'
2: **for** each entry $(r, \mu, \mu'') \in \Lambda$, **do**
3: **if** $r = r'$, **then**
4: Apply $\{\Pi^+, \mathbb{1} - \Pi^+\}$ to $\mu'\mu''$
5: **if** outcome is 0, **then** Send μ to $\mathcal{A}$ // replay detected
6: Send $|\bot\rangle\langle\bot|$ to $\mathcal{A}$

Fig. 30: Modified 'ideal' oracles O_E and O_D for the qAE games.

Definition 35 (qAE Security for SKE) *Let S be a SKE scheme. Then S is* qAE *secure, iff for any QPT adversary $\mathcal{A}$ it holds:*

$$|\Pr[\text{qAE-Real}(S, \mathcal{A}, \lambda) \to 0] - \Pr[\text{qAE-Ideal}(S, \mathcal{A}, \lambda) \to 0]| \leq \text{negl}(\lambda).$$

In [AGM18], constructions of SKQE schemes are given, which achieve the QAE security definition (cf. [AGM18, Thm. 14]). Following the same strategy, it is easy to prove that the following scheme achieves qAE.

Construction 36 (qAE SKE Scheme) *Given a (family of) post-quantum pseudorandom functions $\{F_k\}_k$ and a (family of) quantum-resistant strong pseudorandom permutations (cf. [GHS16]) $\{\pi_{k'}\}_{k'}$ on $\mathcal{M} \times \mathcal{T}$, consider the following construction $S = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$, as shown in Fig. 31.*

For encryption with a secret key k, this construction first samples a fresh randomness r and uses F_k on it to derive a one-time key k'. Then, it pads the message m with a $(0 \dots 0)$ tag, and applies $\pi_{k'}$ to it. The result, together with r, is the ciphertext.

Decryption works in the natural way, i.e., first read r and use F_k to derive the one-time key, then use this one-time key to undo the permutation, and only accept the decryption if the resulting tag is $(0 \dots 0)$.

$\mathtt{KGen}(\lambda; r)$

1: $k \xleftarrow{\$} \mathcal{K}$
2: **return** k

$\mathtt{Enc}_k(m; r)$

1: $k' \leftarrow F_k(r)$
2: $c_1 \leftarrow \pi_{k'}(m \| 0 \dots 0)$
3: **return** (c_1, r)

$\mathtt{Dec}_k(c)$

1: **Parse** c **as** (c_1, r)
2: $k' \leftarrow F_k(r)$
3: $(m, t) \leftarrow \pi_{k'}^{-1}(c_1)$
4: **if** $t \neq 0 \dots 0$, **then return** $\perp$
5: **else return** m

Fig. 31: Example of qAE secure SKE construction.

C Separations

C.1 Separating qIND-qCPA and qIND-qCCA1

In this section, we show that qIND-qCCA1 is indeed a stronger security notion than qIND-qCPA (cf. Def. 23). Note that both notions are defined in the same way, using the real-or-random approach, except that the adversary only has access to the encryption oracle in the qIND-qCPA game, whereas she has access to both oracles (encryption and decryption) only in the qIND-qCCA1 game. Thus, it is obvious that any qIND-qCCA1 secure PKE scheme is qIND-qCPA secure as well:

Proposition 1. *Let Π be a* PKE *scheme. If Π is* qIND-qCCA1 *secure, then Π is* qIND-qCPA *secure.*

Proof. We prove by contraposition: Let Π be a qIND-qCPA insecure PKE scheme. This implies the existence of an adversary $\mathcal{A}$ who wins the qIND-qCPA game against Π with non-negligible advantage, i.e., $\mathcal{A}$ can distinguish between the real and random case with *only* access to $O^{\Diamond}_{\mathtt{Enc},\pi}$. Accordingly, $\mathcal{A}$ wins qIND-qCCA1 game against Π with non-negligible advantage as well. □

For the separation of both notions, we consider the following construction:

Construction 37 *Let $\overline{\Pi} = (\overline{\mathtt{KGen}}, \overline{\mathtt{Enc}}, \overline{\mathtt{Dec}})$ be the modification of a* PKE *scheme $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$, as defined in Fig. 32. The modified scheme encrypts as Π, if the chosen randomness is not* weak *(i.e., $\neq \overline{r}$ where $\overline{r} \in \mathcal{R}$ is fixed and specified in the public key), whereas any message with $\overline{r}$ encrypts to a special ciphertext ($\overline{c}$) which is also specified in the public key, i.e., $\overline{\mathtt{Enc}}_{\overline{\mathsf{pk}}}(m; \overline{r}) = \overline{c}$ for any $m \in \mathcal{M}$. Similar to that, $\overline{\Pi}$ decrypts as Π besides for the ciphertext $\overline{c}$, which decrypts to the secret key.*

This modification preserves the qIND-qCPA security of the underlying scheme:

Lemma 38 *If Π is* qIND-qCPA *secure, then $\overline{\Pi}$ obtained from Construction 37 (using Π) is* qIND-qCPA *secure.*

$\overline{\mathtt{KGen}}(\lambda; r)$
1: $\overline{r} \leftarrow^{\$} \mathcal{R}$
2: $\overline{c} \leftarrow^{\$} \mathcal{C}$
3: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}(\lambda; r)$
4: **return** $(\overline{\mathsf{pk}} \coloneqq (\mathsf{pk}, \overline{r}, \overline{c}), \mathsf{sk})$

$\overline{\mathtt{Enc}}_{\overline{\mathsf{pk}}}(m; r)$
1: **if** $r = \overline{r}$, **then**
2: **return** $\overline{c}$
3: **else**
4: **return** $\mathtt{Enc}_{\mathsf{pk}}(m; r)$

$\overline{\mathtt{Dec}}_{\mathsf{sk}}(c)$
1: **if** $c = \overline{c}$, **then**
2: **return** sk
3: **else**
4: **return** $\mathtt{Dec}_{\mathsf{sk}}(c)$

Fig. 32: A modification $\overline{\Pi}$ of the scheme Π. It is easy to see, that $\overline{\Pi}$ decrypts a message-ciphertext pair (m, c) correctly iff the underlying scheme does, $c \neq \overline{c}$ and the chosen randomness for the encryption is *weak* ($\neq \overline{r}$).

Proof. We prove by contraposition: Assume there exists an adversary $\mathcal{A}$ against the qIND-qCPA security of $\overline{\Pi}$, i.e.,

$$\mathbf{Adv}^{\text{qIND-qCPA}}_{\overline{\Pi}}(\mathcal{A}) > \text{negl}(\lambda),$$

then we can construct an adversary $\mathcal{B}$ against the qIND-qCPA security of Π.

$\mathcal{B}$ plays the qIND-qCPA$_{\Diamond}$ game against $\mathfrak{C}$ and simulates $\overline{\mathfrak{C}}$ for $\mathcal{A}$. In particular, $\mathcal{B}$ samples $\overline{r} \leftarrow^{\$} \mathcal{R}$ and $\overline{c} \leftarrow^{\$} \mathcal{C}$, and sends $(\mathsf{pk}, \overline{r}, \overline{c})$ to $\mathcal{A}$, where pk is the public key she received from $\mathfrak{C}$. For any encryption query of $\mathcal{A}$, $\mathcal{B}$ sends this query to her challenger and receives a response of the form $\chi\varepsilon(U^{\pi}\rho)$. Since the challenger picked a classical randomness, the register $U^{\pi}\rho$ is not entangled with $\chi\varepsilon$ and can be measured in the computational basis by $\mathcal{B}$ without affecting the other registers. She checks, whether the weak randomness $\overline{r}$ was used for encryption, by comparing the measurement with $\pi(\overline{r})$ – which is a value she received from sending $\overline{r}$ to the (classical) oracle O_{π}.

In case the weak randomness was used from the challenger, $\mathcal{B}$ discards $\chi\varepsilon$ and the measurement of $(U^{\pi}\rho)$ and prepares the quantum pure state

$$|\overline{c}||0 \dots 0||\pi(\overline{r})\rangle\langle\overline{c}||0 \dots 0||\pi(\overline{r})| \in \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{E}} \otimes \mathfrak{H}_{\mathcal{R}}),$$

which she sends to $\mathcal{A}$. Note that this tag is valid regarding $\overline{\Pi}$, since there is no *error* when encrypting the message $m = \mathsf{sk}$ with randomness $\overline{r}$:

$$\mathsf{sk} \oplus \mathtt{Dec}_{\mathsf{sk}}(\mathtt{Enc}_{\mathsf{pk}}(\mathsf{sk}; \overline{r})) = \mathsf{sk} \oplus \mathtt{Dec}_{\mathsf{sk}}(\overline{c}) = 0 = e.$$

In case any other randomness was used, $\mathcal{B}$ takes the measurement, prepares it as a quantum state to restore $\mathfrak{C}$'s response $\chi\varepsilon(U^{\pi}\rho)$ and sends it to $\mathcal{A}$.

Once $\mathcal{A}$ sends her challenge, $\mathcal{B}$ forwards it to $\mathfrak{C}$ as her challenge. For the challenge ciphertext, she again checks if the weak randomness was used and treats the challenge as the previous encryption queries. In the end, $\mathcal{B}$ outputs the same bit as $\mathcal{A}$. Since $\mathcal{B}$ perfectly simulates $\overline{\Pi}$, it follows

$$\mathbf{Adv}^{\text{qIND-qCPA}}_{\Pi}(\mathcal{B}) = \mathbf{Adv}^{\text{qIND-qCPA}}_{\overline{\Pi}}(\mathcal{A}) > \text{negl}(\lambda).$$

Thus, Π is qIND-qCPA insecure and the claim follows by contraposition. □

However, if the adversary has $O^{\Diamond}_{\mathtt{Dec},\pi}$ available she can ask for a decryption of $\overline{c}$ to get the secret key allowing her to win the qIND-qCCA1 game trivially:

Theorem 39 *Construction 37 is* qIND-qCCA1 insecure.

Proof. Let $\mathcal{A}$ be an adversary playing the qIND-qCCA1 game against $\overline{\Pi}$. $\mathcal{A}$ sends $\overline{c}$ as classical decryption query to $\mathfrak{C}$, who decrypts the ciphertext to the secret key sk and sends it back to $\mathcal{A}$. Since $\mathcal{A}$ knows sk, she can pick any classical message $m \in \mathcal{M}$ as her challenge and trivially distinguish the real from the random case with overwhelming probability: Once she receives the challenge ciphertext, she measures and decrypts the ciphertext register (using sk). If the fully mixed state $\tau_{\mathcal{M}}$ was encrypted, the decrypted ciphertext collapses to m only with probability $1/|\mathcal{M}|$. It follows:

$$\mathbf{Adv}^{\text{qIND-qCCA1}}_{\overline{\Pi}}(\mathcal{A}) = 1 - \frac{1}{|\mathcal{M}|} - \frac{1}{2} > \text{negl}(\lambda).$$

Thus, the PKE scheme $\overline{\Pi}$ is qIND-qCCA1$_{\Diamond}$ insecure. □

C.2 Separating qIND-qCCA1 and qIND-qCCA2

Proposition 40 *Any* qIND-qCCA2 *secure* PKE *scheme is* qIND-qCCA1 *secure.*

Proof. We prove the claim by contraposition: Consider a qIND-qCCA1 insecure PKE scheme Π. This implies, that there exists an adversary $\mathcal{A}$ who wins the qIND-qCCA1 game against Π with non-negligible advantage, i.e., $\mathcal{A}$ can distinguish the real and ideal world with *only* oracle-access in the *pre-challenge phase.* Accordingly, $\mathcal{A}$ is also successful in the qCCA2-Test game without sending any query after the challenge phase, hence

$$\Pr[\text{qCCA2-Test}(\Pi^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{win}] > \frac{1}{2} + \text{negl}(\lambda),$$
$$\Pr[\text{qCCA2-Fake}(\Pi^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{cheat}] = \frac{1}{2},$$

and Π is qIND-qCCA2 insecure. □

Next, we show the separation between qIND-qCCA1 and qIND-qCCA2 by considering the following construction:

Construction 41 *Consider the following modification* $\overline{\Pi} = (\overline{\texttt{KGen}}, \overline{\texttt{Enc}}, \overline{\texttt{Dec}})$ *obtained by a* PKE *scheme* $\Pi = (\texttt{KGen}, \texttt{Enc}, \texttt{Dec})$*, as shown in Figure 33.*

Our modification combines the underlying PKE *scheme with a one-bit one-time-pad. Therefore the scheme has the same ciphertext-space as* Π *but extended by one bit and requires an additional randomness bit for the encryption, i.e.,* $\overline{\mathcal{C}} := \mathcal{C} \times \{0,1\}$ *and* $\overline{\mathcal{R}} := \mathcal{R} \times \{0,1\}$*. We denote* $\overline{r} := r||r^* \in \overline{\mathcal{R}}$*, where* $r \in \mathcal{R}$ *is used as randomness for the underlying scheme and* r^* *is the extra bit. The same notation we use for any* $\overline{c} \in \overline{\mathcal{C}}$*. Furthermore, we denote* $m_1 \in \{0,1\}$ *as the last bit of any* $m \in \mathcal{M}$ *and* m_0 *as the rest of the message.*

$\overline{\texttt{KGen}}(\lambda; r)$

1: **return** $\texttt{KGen}(\lambda; r)$

$\overline{\texttt{Enc}}_{\mathsf{pk}}(m \coloneqq m_0||m_1; \overline{r} \coloneqq r||r^*)$

1: **return** $\texttt{Enc}_{\mathsf{pk}}(m_0||r^*; r)||(m_1 \oplus r^*)$

$\overline{\texttt{Dec}}_{\mathsf{sk}}(\overline{c} \coloneqq c||c^*)$

1: $m_0||r^* \leftarrow \texttt{Dec}_{\mathsf{sk}}(c)$
2: $m_1 \leftarrow r^* \oplus c^*$
3: **return** $m_0||m_1$

Fig. 33: The construction of $\overline{\Pi}$ is – under the assumption that Π is qIND-qCCA1 secure – also qIND-qCCA1 secure *but* qIND-qCCA2 insecure PKE scheme. It is easy to see, that Construction 41 decrypts a message-ciphertext pair $(m, \overline{c} \coloneqq c||c^*)$ correctly iff the underlying scheme decrypts (m, c) correctly.
For the separation result, we first prove:

Lemma 42 *If Π is* qIND-qCCA1 *secure, then $\overline{\Pi}$ obtained from Construction 41 (using Π) is* qIND-qCCA1 *secure.*

Proof. Let Π be a PKE scheme, and $\overline{\Pi}$ the PKE scheme obtained from Construction 41 using Π. We prove the claim by contraposition. So assume that $\overline{\Pi}$ is qIND-qCCA1 *insecure*, i.e., there exists an adversary $\mathcal{B}$ such that

$$\begin{aligned}&\Pr[\text{qCCA2-Test}(\overline{\Pi}^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{win}]\\ &\quad - \Pr[\text{qCCA2-Fake}(\overline{\Pi}^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{cheat}] > \text{negl}(\lambda),\end{aligned}$$

where $\overline{\Pi}^{\Diamond}$ is the purification of $\overline{\Pi}$.

Next we transform $\mathcal{B}$ into a qIND-qCCA1 adversary $\mathcal{A}$ against Π. The adversary $\mathcal{A}$ runs $\mathcal{B}$ under the same public key, as she got from her challenger. For any classical and quantum decryption query, she traces out the last bit (c^*) resp. qubit ($|c^*\rangle$) before forwarding to her decryption oracle. After receiving the response plaintext, she xor's the last qubit of the plaintext with c^* resp. $|c^*\rangle$ and sends the result to $\mathcal{B}$.

For any encryption query she replaces the last qubit ($|m_1\rangle$) with a random bit $r^* \leftarrow^{\$} \{0,1\}$ and forwards this to her encryption oracle. After receiving the response ciphertext, she appends $|m_1 \oplus r^*\rangle$ to the ciphertext before sending the response to $\mathcal{B}$. $\mathcal{A}$ repeats this procedure once she receives the challenge query. When $\mathcal{B}$ outputs her guess, $\mathcal{A}$ forwards this bit as her own output.

It holds that $\mathcal{A}$ perfectly simulates the qIND-qCCA1 game against $\overline{\Pi}$ for $\mathcal{B}$. Thus, it holds that

$$\begin{aligned}\Pr[\text{qCCA2-Test}(\Pi^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{win}] &= \Pr[\text{qCCA2-Test}(\overline{\Pi}^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{win}],\\ \Pr[\text{qCCA2-Fake}(\Pi^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{cheat}] &= \Pr[\text{qCCA2-Fake}(\overline{\Pi}^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{cheat}],\end{aligned}$$

where $\Pi^{\Diamond}$ is the purification of Π, and Π qIND-qCCA1 insecure. □

Theorem 43 *Construction 41 is* qIND-qCCA2 insecure.

Proof. Let $\overline{\Pi}$ the PKE scheme obtained from Construction 41 and $\overline{\Pi}^{\Diamond}$ its purification. Consider the following adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ playing the qIND-qCCA2

game against $\overline{\Pi}^{\Diamond}$. $\mathcal{A}_1$ chooses a classical message m, prepares a memory state $\sigma = |m\rangle$ and a challenge $\mu = |m\rangle$ for the challenger $\mathfrak{C}$.

In the qCCA2-Test game, the challenger $\mathfrak{C}$ generates a classical randomness r and either encrypts through $O^{\Diamond}_{\mathtt{Enc},\pi}$ the challenge μ or $\tau_{\mathcal{M}}$. Let be $\chi\varepsilon\upsilon$ the output of the oracle. Afterward $\mathcal{A}_2$ receives $\chi\varepsilon\upsilon$ and the memory state σ. She measures χ in the computational basis obtaining a classical ciphertext $c||c^*$, flips the last bit and sends $c' \coloneqq c||(c^* \oplus 1$ as a classical decryption query to the challenger.

In case the real message was encrypted, it holds $c||c^* = \mathtt{Enc}_{\mathsf{pk}}(m_0||m_1)$ and the ciphertext c' decrypts to $m_0||(m_1 \oplus 1)$. After adversary $\mathcal{A}_2$ receives the response m' of her decryption query, she measures the memory state σ, and checks if both are equal up to the last qubit. If yes, she knows (with overwhelming probability) that the real message was encrypted and outputs $b' = 0$. Otherwise she outputs $b' = 1$, thus

$$\Pr[\text{qCCA2-Test}(\overline{\Pi}^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{win}] = 1 - \mathrm{negl}(\lambda),$$

where $\mathrm{negl}(\lambda)$ is the probability that the encrypted fully mixed state $\tau_{\mathcal{M}}$ has collapsed to $\overline{\mathtt{Enc}}_{\mathsf{pk}}(\mu; \overline{r})$.

If $\mathcal{A}$ applies the same strategy in the qCCA2-Fake game, the challenger outputs reject with probability 1 since the stored half of the maximally entangled state and the decryption of the sent ciphertext, are not the same anymore. This is because the adversary obtained the ciphertext by first measuring the encrypted half maximally entangled state and then flipping the last bit. Thus, the outcome of the projective measurement is 1 and the challenger returns cheat with probability 0. Thus, $\mathcal{A}$ is a successful qIND-qCCA2 adversary, i.e.,

$$\begin{aligned}&\Pr[\text{qCCA2-Test}(\overline{\Pi}^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{win}] - \Pr[\text{qCCA2-Fake}(\overline{\Pi}^{\Diamond}, \mathcal{A}, \lambda) \to \mathsf{cheat}] \\ &\quad = 1 - \mathrm{negl}(\lambda) - 0,\end{aligned}$$

and $\overline{\Pi}$ is qIND-qCCA2 insecure. □

C.3 qIND-qATK implies IND-ATK

Proposition 44 (ATK = CPA **or** CCA1) *Let* ATK *be either* CPA *or* CCA1*, and let* Π *be* qIND-qATK *secure* PKE *scheme. Then* Π *is* IND-ATK *secure.*

Proof. We prove by contraposition: Let Π be a IND-ATK insecure PKE scheme. Assume there exists an adversary $\mathcal{A}$ against the real-or-random IND-ATK security of Π, i.e.

$$\mathbf{Adv}^{\text{IND-ATK}}_{\Pi}(\mathcal{A}) > \mathrm{negl}(\lambda),$$

then we can construct an adversary $\mathcal{B}$ against the qIND-qATK security of Π.

Let $\Pi^{\Diamond}$ be the purification of Π. $\mathcal{B}$ plays the qIND-qATK game for $\Pi^{\Diamond}$ against $\mathfrak{C}$ and simulates $\overline{\mathfrak{C}}$ for $\mathcal{A}$:

- $\mathcal{B}$ runs $\mathcal{A}$ under the same pk and λ as she received from $\mathfrak{C}$.
- Note that in the classical IND-ATK game, we have no encryption queries as the adversary can encrypt be herself.

– If ATK = CCA1, then $\mathcal{B}$ forwards any decryption query c of $\mathcal{A}$ to her own classical decryption oracle $\mathtt{Dec}_{\mathsf{sk}}$. $\mathcal{B}$ forwards the response to $\mathcal{A}$.
– In the end, $\mathcal{B}$ outputs the same bit as $\mathcal{A}$.

As $\mathcal{B}$ perfectly simulates the IND-ATK game for $\mathcal{A}$, we have

$$\mathbf{Adv}_{\Pi}^{\text{qIND-qATK}}(\mathcal{A}) > \text{negl}(\lambda).$$

This proves the claim. □

Proposition 45 ($ATK = \text{CCA2}$) *Let $\Pi = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a qIND-qCCA2 secure PKE scheme, then Π is* IND-CCA2*.*

Proof. We prove by contraposition: Let Π be a IND-CCA2 insecure PKE scheme. Assume there exists an adversary $\mathcal{A}$ against the real-or-random IND-CCA2 security of Π, i.e.,

$$\mathbf{Adv}_{\Pi}^{\text{IND-CCA2}}(\mathcal{A}) > \text{negl}(\lambda),$$

then we can construct an adversary $\mathcal{B}$ against the qIND-qCCA2 security of Π.

$\mathcal{B}$ plays either the qCCA2-Test or qCCA2-Fake game against $\mathfrak{C}$ and simulates $\bar{\mathfrak{C}}$ for $\mathcal{A}$:

– $\mathcal{B}$ runs $\mathcal{A}$ under the public key pk as she received from $\mathfrak{C}$.
– Note that in the classical IND-CCA2 game, we have no encryption queries, as the adversary can encrypt by herself.
– Decryption queries in the learning phase: On any query c, $\mathcal{B}$ sends c to her classical decryption oracle and forwards the response to $\mathcal{A}$.
– Challenge phase: Once $\mathcal{A}$ sends the challenge message m, $\mathcal{B}$ prepares the state $|m\rangle$ and sends it to $\mathfrak{C}$ as her own challenge. $\mathcal{B}$ receives the challenge $\chi\varepsilon\upsilon$, traces out ε, computes $c^* \leftarrow \mathfrak{M}(\chi)$ and $r^* \leftarrow \mathfrak{M}(\upsilon)$, and sends c^* to $\mathcal{A}$.
– Decryption queries in the post-challenge phase: For any decryption query $c \neq c^*$, she sends the ciphertext to her classical decryption oracle. If $\mathcal{A}$ queries the challenge ciphertext, $\mathcal{B}$ responses with $\perp$.
– In the end, $\mathcal{B}$ outputs the same bit as $\mathcal{A}$.

In case, $\mathfrak{C}$ runs the qCCA2-Test game, then $\mathcal{B}$ simulated perfectly the IND-CCA2 game. Note, that in case $\mathfrak{C}$ encrypted the maximally mixed state $\tau_{\mathcal{M}}$, then the measurement of χ causes it to collapse to the encryption of a random message. Thus,

$$\Pr[\text{qCCA2-Test}(\Pi^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{win}] = \Pr[\text{IND-CCA2}(\Pi, \mathcal{A}, \lambda) \to \mathsf{win}] \geq \frac{1}{2} + \text{negl}(\lambda).$$

In case, $\mathfrak{C}$ runs the qCCA2-Fake game, $\mathcal{B}$ simulates perfectly the IND-CCA2 game where always the random message gets decrypted. As $\mathcal{B}$ "blocks" the cheating queries of $\mathcal{A}$ (by comparing the incoming ciphertext with c^*), $\mathfrak{C}$ does not abort the game. Thus,

$$\Pr[\text{qCCA2-Fake}(\Pi^{\Diamond}, \mathcal{B}, \lambda) \to \mathsf{cheat}] = \frac{1}{2},$$

and Π is not qIND-qCCA2 secure. □

D Missing Proofs

D.1 Proof of Theorem 8

Theorem 46. *Let $U_1, \dots, U_n$ be unitaries which operate on a shared set of variables $\{x_1, \dots, x_m\}$ (with domains $X_1 \times \cdots \times X_m$), where each application of a unitary U_i corresponds to a (type-1)* conversion *of the input variables $I_i \coloneqq \{i_1^{(i)}, \dots\}$ to the output variables $O_i \coloneqq \{o_1^{(i)}, \dots\}$. For instance, if U_1 converts x_1 into x_2 and U_2 converts x_2 into x_1, then $U_2 U_1(x_1) = x_1$ for any $x_1 \in X_1$.*

Let U^* be an operator which is constructed by a conversion-respecting *composition (i.e., the composition will not feed a variable $x_j \neq x_i$ to a unitary as x_i) from the unitaries $U_1, \dots, U_n$, and let $S^* \subset \{x_1, \dots, x_m\}$ be the set of output variables of U^*. Then there exists a unitary U_i – with input variables I_i – , such that $I_i \subset S^*$.*

Proof. Let U^* be an operator constructed by composition from the unitaries $U_1, \dots, U_n$ and let $S^* \subset \{x_1, \dots, x_m\}$ be the set of output variables of U^*.

First, we make the observation that the composition of U^* can be described as a finite linear sequence of type-1 operators $U_1, \dots, U_n$ and the wires on which they are to be applied. This sequence corresponds to the composition order of these operators in the circuit of U^* from left to right. It might be that the composition applies operators in parallel, but then they operate on different wires, so that assigning an order to these operators is equivalent to executing them in parallel.

Second, we claim: if U_i is the last type-1 operator in the sequence, which describes the operator U^*, then $I_i \subset S^*$. Note that this claim implies the theorem. We prove this claim by induction over k, where k is the length of the sequence.

Induction Start: $k = 1$. Let U^* be a operator built from *one* unitary, w.l.o.g. assume from U_1 with $I_1 = \{i_1, \dots, i_k\}$ and $O_1 = \{o_1, \dots, o_l\}$. Then $S^* = I_1 \cup O_1$, in particular we have $I_1 \subset S^*$.

Induction Claim: For any $k \in \mathbb{N}$ and any U^* which can be described by a linear sequence of k operators, where U_i is the k-th operator in the sequence, it holds that $I_i \subset S^*$.

Induction Step: $k \to k+1$. Let U^* be an operator which can be described by a linear sequence of $k+1$ operators. Let U_j and U_i be the k-th and $(k+1)$-st operator in the sequence, respectively, and let U' be the composed operator which is defined by U^*'s sequence without the last operator. Then, by induction claim, it holds that I_j is a subset of the output variables of U'. In case $I_j \cap O_i \neq \emptyset$, some of U_j's input variables become the zero register $|0\rangle$ via "back computation" by operator U_i. This implies that $I_j \not\subset S^*$. However, since U_i is the "last piece", there must be wires containing (all) the input variables I_i after the application of U' and, so, also after U^* (cf. Fig. 34). □

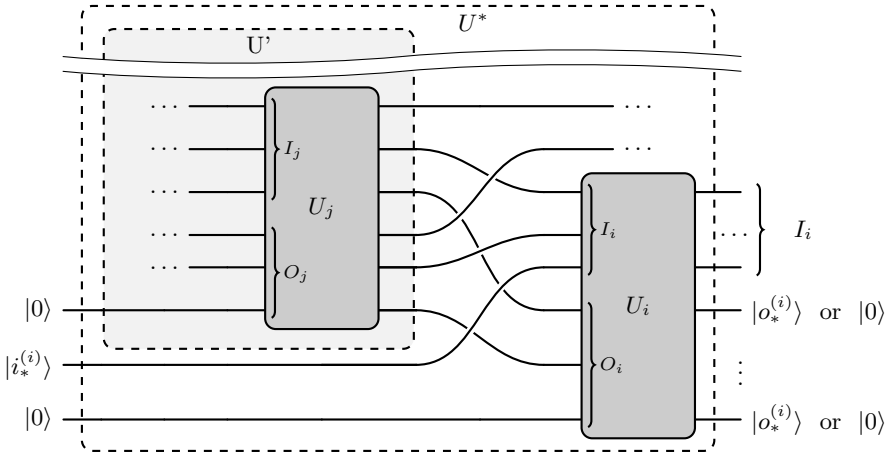

Fig. 34: Any operator U^* which can be described by a linear sequence of $k+1$ operators, where U_j is the k-th and U_i is the $k+1$-th operator in the sequence. It holds that I_j is contained in U''s output variables, where U' is the composed operator which is defined by U^*'s sequence without the last operator, and I_i is contained in U^*'s output variables.

D.2 Proof of Proposition 9

Proposition 47 *It is impossible to construct a type-2 decryption operator from type-1 operators only, which outputs neither r nor c registers.*

Proof. For an encryption scheme, we have the following operators (which may have pk and sk as additional auxiliary input, but this is not relevant here):

$$\mathtt{Enc} : (m, r) \mapsto c, \quad \mathtt{Enc}' : (m, r) \mapsto (c, t), \quad \mathtt{Tag} : (m, r) \mapsto t, \quad \mathtt{Rec} : (r, c) \mapsto m,$$
$$\mathtt{Dec} : c \mapsto m', \quad \mathtt{Dec}' : (c, t) \mapsto m, \quad \mathtt{sDec}^{19} : c \mapsto (m, r),$$

where t denotes a tag which contains auxiliary information that could help to recover the randomness (this becomes relevant in Sec. 5), and m' the decryption of c which is possibly $\neq m$ if there occurs a decryption failure.

Note, that it is enough to consider *only* the type-1 operators in the proof, since the other available type-2 operators have the same type-1 operators as building blocks. All of these operators can be seen as "conversions": for instance, for fixed m, r, we can compute c, t, m', but also convert c, t to the previous m. According to Thm. 8, a type-2 decryption operator, which neither outputs c nor r can only exist if there is at least one unitary U with input variables $I \subset \{m, m', t\}$ (cf. Fig. 35).
This is not the case for the listed operators above, and also cannot build this operator directly, since it is not well-defined. On one hand, it is not clear how

[19] This algorithm exists for so-called *strongly decryptable* schemes, on which we will focus in Sec. 4.

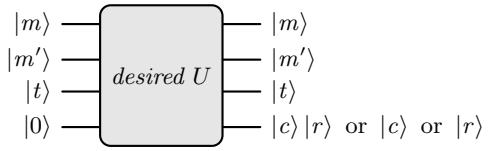

Fig. 35: The desired type-1 operator which would allow us to build a type-2 decryption operator.

to reconstruct $c = \mathtt{Enc}(m; r)$ and r given m, m' (but not knowing r). The only helpful additional information, which could be encoded in a tag t is c or r itself.

On the other hand, it is not possible to define this unitary as "conversion" as for a reasonable PKE scheme – a majority of all possible messages $m \in \mathcal{M}$ is decryptable without error, i.e., we have $m = \mathtt{Dec}(\mathtt{Enc}(m; r))$. Accordingly, a majority of the input pairs (m, m'), where $m \neq m'$, does not make sense. This concludes the proof that we cannot build the desired type-2 operator. □

D.3 Proof of Proposition 11

Proposition 2. *Strong decryptability is* independent *of both recoverability and perfect correctness, i.e., strong decryptability does not imply either of the other two properties, nor does either of them imply strong decryptability.*

Proof. We prove the proposition in four steps:

(1) Recoverability $\not\Rightarrow$ strong decryptability. Consider a recoverable PKE scheme Π and modify it (to Π') to take an extra bit of randomness as input, but ignore that extra bit during encryption. The resulting scheme is also recoverable: given (pk, r, c), define the recovery algorithm of Π' as $\mathtt{Rec}'_{\mathsf{pk}}(r, c) \coloneqq \mathtt{Rec}_{\mathsf{pk}}(r', c)$, where $\mathtt{Rec}$ is the recovery algorithm of Π and r' is r with the extra bit truncated. But now, given only sk and c, it is unlikely to recover the last bit of the original randomness, as that extra bit of information was lost during encryption. Thus, Π' is not strongly decryptable.

(2) Strong decryptability $\not\Rightarrow$ recoverability. Consider a strongly decryptable PKE scheme Π and modify it (to Π') by post-processing generated ciphertexts with a quantum-secure trapdoor permutation family P. Given a key pair $(\mathsf{pk}_P, \mathsf{sk}_P)$ for P, we write P_{pk_P} for the trapdoor permutation and $P^{-1}_{\mathsf{sk}_P}$ for its inverse. The modified key generation $\mathtt{KGen}'$ generates a trapdoor key pair together with a standard key pair from $\mathtt{KGen}$, i.e., the new key pair is $((\mathsf{pk}, \mathsf{pk}_P), (\mathsf{sk}, \mathsf{sk}_P))$. The modified encryption $\mathtt{Enc}'$ first encrypts as $\mathtt{Enc}$, and then permutes the ciphertext using P, i.e., $\mathtt{Enc}'_{(\mathsf{pk},\mathsf{pk}_P)}(m; r) \coloneqq P_{\mathsf{pk}_P}(\mathtt{Enc}_{\mathsf{pk}}(m; r))$. The resulting scheme is also strongly decryptable: given $(\mathsf{sk}, \mathsf{sk}_P, c)$, define $\mathtt{sDec}'_{(\mathsf{sk},\mathsf{sk}_P)}(c) \coloneqq \mathtt{sDec}_{\mathsf{sk}}(P^{-1}_{\mathsf{sk}_P}(c))$, where $\mathtt{sDec}$ is the strong decryption of Π. But now, given only $(\mathsf{pk}, \mathsf{pk}_P), r, c$, it is unlikely to decrypt, since this would yield a reduction against the one-wayness of the trapdoor permutation. Thus, Π' is not recoverable.

(3) Perfect correctness $\not\Rightarrow$ strong decryptability. As seen in the previous proposition, consider a perfectly correct PKE scheme and modify it to take

an extra bit of randomness as input, but ignore that extra bit during encryption. The resulting scheme is still perfectly correct, but there is no hope of recovering all the original randomness from a given ciphertext.

(4) Strong decryptability $\not\Rightarrow$ perfect correctness. Consider a strongly decryptable PKE scheme, and modify `Dec` as follows:

1. first, fix a pair (m, r) and compute a special $c' \coloneqq \mathtt{Enc}_{\mathsf{pk}}(m; r)$;
2. `Dec` decrypts as usual except if the ciphertext is c', then `Dec` returns $\perp$.

It is easy to see that the resulting PKE scheme is not perfectly correct, but it is still strongly decryptable, as `sDec` is the same. □

D.4 Proof of Proposition 12

Proposition 3. *Among* PKE *schemes which are both perfectly correct and recoverable, some are strongly decryptable, and others are not. Among efficiently isometric* PKE *schemes there are schemes which are neither recoverable, nor perfectly correct, nor strongly decryptable.*

Proof. We prove the proposition in two steps:

First claim. We have already seen an example (RSA-OAEP) which fulfills all properties. So we only need to find a separating counterexample to prove the claim. Consider again the RSA-OAEP scheme, and modify it into a new PKE scheme, where the encryption first preprocesses the input randomness by a quantum-secure one-way permutation. The new resulting scheme is still perfectly correct: the one-way permutation just scrambles the randomness space, so the resulting encryption is still an injection. It is also recoverable, because the new `Rec` algorithm still works in recovering the plaintext, by first applying the one-way permutation to the input randomness. However, if an efficient `sDec` algorithm existed, it would be able to recover the original randomness from the ciphertext only. This would yield a reduction against the one-way permutation.

Second claim. Let Π be an efficiently isometric, recoverable, and partially correct PKE. Let T be a family of quantum-secure trapdoor permutations. Modify Π (to Π'), where during encryption the message is preprocessed through an element of T, and the inverse T^{-1} is applied to recover the message during decryption (the trapdoor key-pair is part of the key-pair of Π'). The resulting scheme Π' is efficiently isometric, has the same correctness of Π, but is not recoverable (this example was provided in [GKS21]). Now, further modify the encryption of Π' by preprocessing the input randomness with a quantum-secure one-way permutation P as in the proof of the first claim. The resulting scheme Π'' has the same properties of Π', but is also not strongly decryptable. □