Make Quantum Indistinguishability Great Again*

Tommaso Gagliardoni[1], Juliane Krämer[2], and Patrick Struck[2]

[1] Kudelski Security, Switzerland
paper.qpke2020@gagliardoni.net
[2] Technische Universität Darmstadt, Germany
{jkraemer,pstruck}@cdc.tu-darmstadt.de

Abstract. In this work we study the (superposition-based, or QS2) quantum security of public key encryption schemes. Boneh and Zhandry (CRYPTO 2013) initiated this research area for symmetric and public key encryption, albeit restricted to a classical indistinguishability phase. Gagliardoni et al. (CRYPTO 2016) advanced the study of quantum security, for symmetric key encryption schemes, by giving the first definition where the indistinguishability phase is quantum. For public key encryption schemes, on the other hand, no notion of quantum security with a quantum indistinguishability phase exists.

In this work we close this gap by defining strong QS2 security notions for the public key case. Our core idea follows the approach of Gagliardoni et al. by using so-called type-2 operators for encrypting the challenge message. Extending this idea to the public key case brings non-trivial obstacles: On the one hand, public key encryption schemes typically cannot recover the randomness when decrypting ciphertexts. On the other hand, many real-world schemes (including most quantum-resistant NIST candidates) suffer from a small probability of decryption failures. These two problems make it difficult to enforce the reversibility of the encryption operation needed by type-2 operators. Nevertheless, we identify a class of encryption schemes, which we call *recoverable*, that allow to avoid decryption failures given knowledge of the original encryption randomness, and we show that many real-world quantum-resistant schemes, including many NIST candidates, are of this type. Then we show how to define and construct type-2 encryption operators for schemes that are fully correct or recoverable. Moreover, we show that for recoverable schemes, the type-2 operator can be efficiently implemented *even without knowledge of the secret key*. This means that, for the public key case, type-2 operators are actually very natural, and already included in the traditional "post-quantum" (QS1) definition of security.

Equipped with these results, we (1) give the first quantum security notion (qIND-qCPA) for public key encryption with a quantum indistinguishability phase, (2) prove that the canonical LWE-based encryption scheme achieves our security notion, (3) show that our notion is strictly stronger than existing security notions, and (4) study the general classification of quantum-resistant public key encryption schemes.

* The title of this work is not meant to convey any political or ideological endorsement by the authors.

1 Introduction

The discovery of Shor's [39] and Grover's [24] quantum algorithms had a significant impact on cryptographic research. Shor's algorithm in particular has the potential to completely break most of the public key cryptosystems used nowadays. This has lead to the development of quantum-resistant cryptography[3], that is, cryptography that can run on non-quantum computers but should withstand attackers equipped with quantum computing power. In recent years the research efforts on quantum-resistant cryptography accelerated significantly due to the standardization process initiated by NIST [35].

Modern cryptography is based on the paradigm of *provable security*, which is itself given in terms of a security notion, an adversarial model, and a security proof. A widely used framework for defining security notions is the so-called *game-based security*, which is presented as a game between two or more parties.[4] In the case of public key encryption schemes these parties are: a challenger, representing the user of the scheme, and an adversary, representing an attacker against the scheme. Any meaningful model for quantum-resistant schemes should entail that the adversary has quantum computing power. Based on this, we can differentiate between different models depending on the computing power of the challenger. In the literature there are mainly two of these models that are taken into account. In the first, the challenger remains fully classical, implying that any communication between adversary and challenger is also classical (including oracles provided by the challenger to the adversary), while the adversary retains local quantum computing power. This is the model most often considered in quantum-resistant cryptography, and it is also called QS1 [21] or Q1 [27]. Second, the challenger also has quantum computing power, which enables quantum communication between the challenger and the adversary. This stronger model is sometimes called "superposition-attack security" [19], QS2 [21], or Q2 [27].

Boneh and Zhandry [13] initiated the study of QS2 security for cryptographic primitives. For signature schemes, they give a security definition that allows the adversary to query the signing oracle on a superposition of messages. For public and symmetric key encryption schemes, on the other hand, they prove that simply allowing the adversary to query a superposition of messages as challenge in a "natural" way gives an unachievable security notion (fqIND-CPA). This is due to entanglement between the plaintext register and the ciphertext register. They show how to exploit this entanglement to break this security notion irrespectively of the used encryption scheme. To resolve this, they propose another security notion (IND-qCPA) which allows the adversary superposition queries in the CPA phase while the challenge messages in the IND phase are restricted

[3] This type of cryptography is often called "post-quantum cryptography "[10]. We consider this term misleading and of not immediate interpretation, so we will not use it in general.

[4] Other frameworks exist, such as simulation-based or concrete security, but we argue that, as a first approximation, game-based security notions are very convenient for their intuitivity and simplicity, even if they can certainly be improved when considering more complex scenarios, such as composability.

to be classical. This notion coincides with the traditional QS1 security notion for public key encryption schemes (as the adversary can simulate the encryption in superposition using his local computing power and the public key), while for secret key encryption schemes, this yields a notion of QS2 security - although the restriction to a classical challenge in this case is clearly a shortcoming.

Gagliardoni et al. [22] overcame this shortcoming in the symmetric key case by showing how to model a quantum challenge query, while keeping the resulting security notion (qIND-qCPA) still achievable, yet stronger than IND-qCPA. At the heart of their idea lies the use of so-called *type-2 operators*[5] rather than so-called type-1 operators when encrypting the challenge messages of the adversary. Type-1 operators are the "canonical" way of implementing an arbitrary, classical function $\mathcal{F}$ on a quantum superposition of input, by mapping the state $|x, y\rangle$ to $|x, y \oplus \mathcal{F}(x)\rangle$, thereby ensuring reversibility for any function $\mathcal{F}$ (reversibility being necessary when defining non-measurement quantum operations). An important property of type-1 operators is that they create entanglement between the input and output registers. This is exactly the entanglement which Boneh and Zhandry exploit to show that fqIND-CPA is unachievable. In contrast to these, type-2 operators work directly on the input register, i.e., they map the state $|x\rangle$ to $|\mathcal{F}(x)\rangle$. Only reversible functions, for instance permutations, can be implemented as type-2 operators, while it is impossible to compute, say, an arbitrary one-way function through a type-2 operator. Gagliardoni et al. observe that secret key encryption schemes act as permutations between the plaintext space and the ciphertext space, which allows to implement the encryption algorithm as a type-2 operator. This, in turn, allows to build a solid framework for QS2 security in the case of symmetric key encryption (SKE).

In [22] the authors speculate that their techniques could be extended to the public key case (PKE) as well. However, defining type-2 operators for PKE schemes is much more involved than for SKE schemes. First, to achieve IND-CPA security, PKE schemes are inherently randomised and the randomness is usually erased in the process of decryption. Second, many constructions for quantum-resistant PKE schemes, in particular lattice-based and code-based schemes, suffer from a small probability of decryption failures, i.e., ciphertexts which do not decrypt correctly. Given the above, at a first glance it is unclear whether type-2 operators for PKE schemes are possible at all, as these two properties seem to thwart the mandatory reversibility. Hence, QS2 security for the public key case remained an open problem so far.

1.1 Our Contribution

In this work we give the first proper QS2 notion for PKE, we show that natural quantum-resistant PKE schemes can achieve these strong security notion, and we provide a separation example and a general classification.

Our core focus is to extend the results from [22] to the public key scenario. We first formalize the theory of type-2 encryption operators for PKE. For fully

[5] Also called *minimal oracles* in [28].

correct schemes (i.e., schemes which do not suffer from the possibility of decryption failures) we define the type-2 operator to preserve a randomness register in input and output. Even if such approach might look strange at a first glance, we show that this is the most natural way of defining type-2 operators for PKE schemes. As a next step, we identify a class of PKE schemes (which we call *recoverable*) where decryption failures can always be avoided given knowledge of the randomness used during encryption, regardless of the actual failure probability of the decryption algorithm. We observe that most real-world partially correct PKE schemes (including many quantum-resistant NIST candidates) are actually of this type. Then, for schemes that are fully correct or recoverable, we show how to efficiently construct the type-2 encryption operator. Moreover, we show that for recoverable schemes, this can be done by knowledge of the public key only! Which implies, perhaps surprisingly, that the adversary can implement efficiently this type-2 operator already in the QS1 model. Such observation marks a substantial difference from the symmetric key case, where the need for type-2 operators was dictated by necessity in order to cover exotic attacks models.

Using the theory of type-2 operators so developed, we give the first proper QS2 security notion for PKE, that we call *quantum ciphertext indistinguishability under quantum chosen plaintext attack* (qIND-qCPA). For a new security notion to be meaningful, two properties are required. First, it has to be achievable, and, second, it has to differ from existing security notions. Regarding achievability, we show that the canonical LWE-based PKE scheme is secure according to our new notion. As all of the lattice-based encryption schemes in the NIST standardization process follow this blueprint, our proof indicates that they are all qIND-qCPA-secure. The main challenge of the proof lies in the simulation of the type-2 encryption operator by the reduction. The canonical way of implementing this operator requires knowledge of the secret key since, in contrast to the security game where the challenger has knowledge of the secret key, the reduction only knows the public key. Even worse, there are game hops where a secret key does not even exist at all. Note that this is an issue that arises only in the QS2 model.[6] We circumvent this issue by showing that the canonical LWE-based scheme is *recoverable*, and hence an efficient realization of the type-2 encryption operator does not require knowledge of the secret key.

Regarding separation results, we give an example which is secure according to the QS1 security notion from [13] while insecure in the sense of our new QS2 security notion. For this we exploit the distinguishing attack given in [22]; this separation example is not artificial, as it follows the general blueprint of hybrid encryption / key encapsulation commonly found in practical applications.

Finally, we discuss the difficulty of defining type-2 operators (and the related QS2 security notion) for arbitrary schemes that are neither fully correct nor recoverable. We study the problem of their general classification and we identify a class of schemes, that we call *isometric*, that allow to overcome such difficulty, and we provide constructions and separation results.

[6] In the QS1 model, the reduction can simply encrypt with the public key regardless of the secret key.

1.2 The Motivation for QS2 Security

Defining security against quantum adversaries with superposition access to certain oracles requires some motivation. In certain cases, the resulting security notion is already implicitly captured by the corresponding QS1 scenario (for example in the case of *quantum random oracles* [11]). In other cases, such as those considered in [6, 26, 30], it might look like an artificial extension of the theory.

However, QS2 security extends quantum properties to types of attack scenarios not covered in QS1, and at the same time "bridges" certain security notions from the classical realm to schemes which are meant to run natively on a quantum computer. Some of the reasons why QS2 notions are important to consider are explained in detail in [21]. They basically boil down to five points.

1. To ensure that quantum-resistant classical schemes retain their security even if executed on a quantum computer, possibly in complex environments or protocols where composition should be taken into account.
2. To fix security proofs, where the sole QS1 security of certain underlying building blocks is not enough to ensure that the whole proof goes through. An example is the need of QS2-secure pseudorandom functions (QPRF) in order to simulate a quantum random oracle [41], which is a QS1 concept.
3. To ensure the security of quantum protocols (i.e., meant to run natively on a quantum computer and protect quantum data) when using classical algorithms as building blocks. For example, [21] shows how it is possible to build a secure symmetric quantum encryption scheme (falling into the so-called QS3 domain) by using a qIND-qCPA symmetric classical encryption scheme (QS2), but not necessarily a simple quantum-resistant (QS1) one.
4. To consider cases of *code obfuscation*; for example creating a quantum-resistant PKE scheme by hardcoding a symmetric key into an obfuscated encryption program (a technique known as *whiteboxing* [17]), which is then distributed as a public key.
5. To cover cases of *exotic quantum attacks*. These include, for instance, *quantum fault injection attacks*, where a classical device is subject to controlled and artificial physical conditions that induce full or partial quantum behaviour of its hardware ("tricking" a classical device into being quantum, like in the "frozen smart-card attack" presented in [22]); or cases where a quantum computer is used to run a classical algorithm, but an adversary manages to intercept the intermediate result of the computation *before* the final measurement meant to produce a classical outcome.

In our specific case, our results follow from the core use of type-2 operators. This kind of quantum operations is poorly studied in the quantum computing realm, and might therefore look artificial for cryptographic use. In the present work we make an effort to expand in a detailed way the formalization of such operators which, we stress, are only given for functions that are inherently invertible. It is a well-known fact (see for example [22]) that implementing these operators for encryption schemes usually requires knowledge of the secret key. We do not consider this to be a limitation because in the quantum setting, an

honest challenger equipped with the secret key could be allowed to generate particular ciphertext states which would be hard to compute for an external party: it is therefore necessary to cover this distinction in the preparation of ciphertext states, and type-2 operators do just that. Moreover, as we show in the present work, for many natural PKE schemes, type-2 encryption operators can actually be efficiently implemented by knowledge of the public key only.

1.3 Related Work

The study of quantum security under adversarial queries in superposition can be traced back to works such as [11, 19, 40], which explore different settings where this additional adversarial power has an impact on security. However, for the case of signatures and encryption schemes, the first framework going beyond the traditional QS1 paradigm was given in [13]. This paradigm was further extended in [22] for symmetric key encryption schemes, and in [20] for signatures.

Regarding examples of exotic quantum attacks previously mentioned: it is currently not known whether any of these are feasible at all, but as noted in [21]: (1) if they are feasible, in some cases they do not even require a fully fledged quantum computer (for example, in the attack from [22] it would be only necessary to produce and detect a Hadamard superposition of messages); and (2) it is already known in the literature that these attacks can be devastating. For example, *related-key attacks* [38], and superposition attacks against Even-Mansour [30], Feistel networks [25, 29], block ciphers [6, 8], and HMAC constructions [26].

Qualitatively different, but technically very connected to the QS2 setting is the *fully quantum setting*, or QS3 in short. This security domain encompasses security notions and constructions for schemes which are natively run on quantum hardware. In the case of QS3 encryption, these are schemes which are meant to protect quantum, rather than classical data. It turns out that many of the challenges in this area are shared with the QS2 case. In the computational security setting, the first security notions have been provided in [15] for the CPA case, and in [3] for the CCA1 and semantic security case. These results have been further extended to the CCA2 setting in [5] for the symmetric case, and in [4] for the public key case.

Concurrent Work. In concurrent and independent work, Chevalier et al. [16] propose alternative QS2 security notions for public and symmetric key encryption schemes. There are important, conceptual differences between this work and ours which we illustrate in this section.

Chevalier et al. start by resuming a game-based quantum indistinguishability notion previously introduced by Mossayebi and Schack [33] which is not comparable to ours. This notion is based on a real-or-permuted approach: in the security game, the adversary sends a *single* quantum plaintext of the form $\sum_x \alpha_x |x\rangle$ and (depending on the value of the secret challenge bit b) receives back either $\sum_x \alpha_x |x, \mathtt{Enc}(x)\rangle$, or $\sum_x \alpha_x |x, \mathtt{Enc}(\pi(x))\rangle$, where π is a random permutation implemented by the challenger. To avoid confusion with our notion

(qIND-qCPA), we refer to their notion as π-qIND-qCPA. Consider the canonical IND-CPA symmetric key encryption scheme that works by XOR-ing the message with $\mathtt{F}_{\mathsf{k}}(r)$, where $\mathtt{F}$ is a keyed pseudorandom function and r is a freshly sampled randomness which is then attached to the resulting ciphertext. This scheme was previously known to be secure according to Boneh and Zhandry's IND-qCPA notion; however, in [33], Mossayebi and Schack show that such scheme is not π-qIND-qCPA secure,[7] thereby yielding a separation result.

Starting from this consideration, the authors of [16] develop a framework of new QS2 security notions (both for the symmetric and public key case) where the challenge query is quantum but implemented as a single message in the real-or-permuted setting. This approach has advantages and disadvantages compared with the one in [22] (for the symmetric key case) and the one we adopt in this work (for the public key case):

- the notion of π-qIND-qCPA (and the related CCA and non-malleability notions) only require the use of type-1 oracles, therefore simplifying a lot the modelling of the security game. Another advantage is that it can be defined for any encryption scheme while we require *isometric schemes* (cf. Section 6).
- On the other hand, the notion of Chevalier et al. (unlike ours) deviates from the established framework for the classical case. In the traditional setting of symmetric and public key security notions, in fact, it is well-known that many different characterizations of IND-CPA (with two or more messages chosen by the adversary, with one chosen and one random or fixed, etc.) are all equivalent to an intuitive (but more cumbersome) notion of semantic security. For π-qIND-qCPA, however, it is *crucial* that the adversary can only send *one single challenge message* to the challenger.[8] This is not the case for our qIND-qCPA (and related) notions: although we do not write them down explicitly here (we leave them for a future update in the appendix of this manuscript), all these good 'sanity checks' can be easily inferred by:
  - the lifting from a two-message qIND (QS2) challenge query to a two-message QIND (QS3) challenge query in [21];
  - the equivalence between different types of QIND challenge query (two messages, many messages, real-or-random, etc) in [15];
  - the equivalence to a sound notion of *quantum semantic security* in [3].

  This means that our notions (in the public key case) and the ones in [22] (for the symmetric key case) closely mirror the well-established framework in the classical setting.
- Analogously, because of the presence of entanglement between plaintext and ciphertext registers, the notions by Chevalier et al. do not mirror the existing solid framework for *fully quantum notions* (QS3 setting) in the literature. This is not a flaw by itself, but it has the drawback that many

[7] The proof of the attack is only sketched, whereas it is formally given by Chevalier et al. in [16].

[8] Chevalier et al. prove the *composability* of their notions, but this refers to the fact that one can formulate their security game using multiple challenge queries, where each query is still restricted to a *single* message.

useful tools cannot be straightforwardly 'imported' from the QS3 setting. An example mentioned above is the difficulty of formalizing the equivalence of π-qIND-qCPA to a natural notion of quantum semantic security, or the possibility of easily lifting the QS2 security of a classical scheme Σ to the QS3 security of a quantum scheme Π that uses Σ as a building block. Another example is the difficulty of defining quantum CCA2 security, which can be done in a relatively easy way in the QS3 setting with the real-vs-ideal approach by Alagic et al. from [5], while requiring the more involved compressed oracle technique by Zhandry [43] for the results in [16].[9]

- Chevalier et al. expand substantially Mossayebi and Schack's results, answering many questions left previously open (some of which also mentioned in this work in Section 7) such as the security of the encrypt-then-MAC construction, and of the hybrid symmetric/public key construction. Moreover they introduce a technique (based on Zhandry's compressed oracles) to record queries and simulate answers to inverse oracles which is of independent interest.
- It is important to notice that the separation result by Chevalier et al. and Mossayeby and Schack rely on entanglement between message and ciphertext register and not on a particular weakness in the scheme. In contrast, our separation (and the one in [22]) rely solely on a property of the encryption scheme in question. One has to consider how the ability of a quantum adversary of receiving back an entangled pair of message and ciphertext mirrors the classical intuition, where an adversary would only receive a ciphertext.
- Finally, and most importantly, in the present work we show that for many real-world PKE schemes (including most of the NIST candidates) type-2 encryption operators can be implemented *without knowledge of the secret key*. This invalidates Chevalier et al.'s argument that type-2 operators are unreasonable in the public key setting, and actually makes the need for our qIND-qCPA notion in the public key case stronger than ever.

Ultimately, we think that the contribution of Chevalier et al. is of great importance and their results are undoubtedly interesting. It is important to notice that the canonical IND-CPA scheme used by Chevalier et al. and Mossayebi and Schack as a separation from Boneh and Zhandry's IND-qCPA is also shown to be insecure according to the qIND-qCPA security notion in [22]. We can hence see π-qIND-qCPA as a QS2 security notion which is incomparable to the qIND-qCPA notion we present in this work, with advantages and disadvantages as explained above.

2 Preliminaries

In the following, we use "classical" as meaning "non-quantum". By *algorithm* or *procedure* (classical or quantum) we mean a uniform family of circuits (classical

[9] The authors of [16] also explain in their work why Alagic et al.'s approach would not work in their case.

or quantum) of depth and width polynomial in the index of the family. We call such index a *security parameter*, and we denote it by λ (or 1^λ if written in unary notation). We implicitly assume that all algorithms take 1^λ as a first input, so we will often omit this. If an algorithm A is deterministic, we denote its output y on input x as $y \coloneqq A(x)$, while if it is randomized we use $y \leftarrow A(x)$; when derandomizing an algorithm we look at the deterministic algorithm obtained when considering explicitly the internal randomness r as an additional auxiliary input, and we write $y \coloneqq A(x; r)$. We will also use $x \leftarrow \mathcal{D}$ to denote that an element x is sampled from a distribution $\mathcal{D}$; or we will write $x \xleftarrow{\$} X$ if x is sampled uniformly at random from a set X. We will call *negligible* a function that grows more slowly than any inverse polynomial, and *overwhelming* a function which is 1 minus a negligible function.

2.1 Quantum Notation

We assume familiarity with the topic of quantum computing, but recall here the basic required notation. For an in-depth discussion we refer to [34].

A quantum system, identified by a letter A, is represented by a complex Hilbert space, which we denote by $\mathfrak{H}_A$. If A is clear from the context, we write $\mathfrak{H}$ rather than $\mathfrak{H}_A$. Pure states in a Hilbert space $\mathfrak{H}$ are representatives of equivalence classes of elements of $\mathfrak{H}$ of norm 1. Mixed states, on the other hand, are a more general representation of quantum states that takes *entanglement* with external systems into account; they are elements of the density matrix operator space over $\mathfrak{H}$, that is, Hermitian positive semi-definite linear operators of trace 1, denoted as $\mathfrak{D}(\mathfrak{H})$. We use the ket notation for pure states, e.g., $|\varphi\rangle$, while mixed states will be denoted by lowercase Greek letter, e.g., ρ. Operations on pure states from A to B are performed by applying a unitary operator $U\colon \mathfrak{H}_A \to \mathfrak{H}_B$ to the state, while the more general case of operations on mixed states is described by superoperators of the form $U\colon \mathfrak{D}(\mathfrak{H}_A) \to \mathfrak{D}(\mathfrak{H}_B)$

The canonical way to compute a classical function $\mathcal{F}\colon \mathcal{X} \to \mathcal{Y}$ on a superposition of possible inputs $\sum_{x\in\mathcal{X}} \alpha_x |x\rangle$ is through the so-called *type-1 operator for* $\mathcal{F}$ described by:

$$U_{\mathcal{F}}^{(1)}\colon \sum_{x,y} \alpha_{x,y} |x, y\rangle \mapsto \sum_{x,y} \alpha_{x,y} |x, y \oplus \mathcal{F}(x)\rangle \ ,$$

This can always be implemented efficiently whenever $\mathcal{F}$ is efficient [34]. By linearity, it is sufficient to specify just the behaviour on the basis elements, i.e.:

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

If $\mathcal{F}$ is invertible, then there is another non-equivalent possible way to compute $\mathcal{F}$ in superposition. This is done through the so-called *type-2 operators*, which are defined as the unitary:

$$U_{\mathcal{F}}^{(2)}\colon |x\rangle \mapsto |\mathcal{F}(x)\rangle \ .$$

See Fig. 1 for an illustration of these different operators. Kashefi et al. [28] first introduced type-2 operators, albeit they called them *minimal oracles*. They show that these operators are strictly stronger by giving a problem which can be solved exponentially faster with type-2 operators than with type-1 operators. They also observe that the adjoint of the type-2 operator corresponds to the type-2 operator of the inverse function $\mathcal{F}^{-1}$, which is (usually) not the case for type-1 operators. Besides that, type-2 operators have been used by Gagliardoni et al. [22] to define quantum security for secret key encryption schemes.

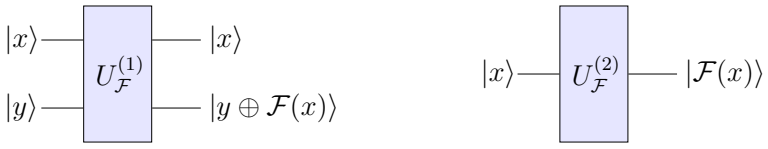

Fig. 1: Type-1 operator (left) and type-2 operator (right) for a function $\mathcal{F}$.

2.2 Public Key Encryption

In this section we give the formal definition for public key encryption schemes and the correctness of such schemes.

Definition 1 *A* public key encryption (PKE) scheme *is a tuple* $(\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *of three efficient algorithms such that:*

- $\mathtt{KGen}\colon \mathbb{N} \times \mathcal{R} \to \mathcal{P} \times \mathcal{S}$ *is the key generation algorithm which takes a security parameter* λ *and a randomness* r *as input, and returns a keypair consisting of a public key* pk *and a secret key* sk*. If clear from the context, we will denote it by* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$*.*
- $\mathtt{Enc}\colon \mathcal{P} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ *is the encryption algorithm which takes a public key* pk*, a message* m*, and a randomness* r *as input, and returns a ciphertext* c*. It will be usually denoted by* $c \leftarrow \mathtt{Enc}_{\mathsf{pk}}(m)$ *or* $c := \mathtt{Enc}_{\mathsf{pk}}(m; r)$*.*
- $\mathtt{Dec}\colon \mathcal{S} \times \mathcal{C} \to \mathcal{M}$ *is the (deterministic) decryption algorithm*[10] *which takes as input a secret key* sk *and a ciphertext* c*, and returns a message* m*. It will be usually denoted by* $m := \mathtt{Dec}_{\mathsf{sk}}(c)$*.*

By $\mathcal{P}$*,* $\mathcal{S}$*,* $\mathcal{M}$*,* $\mathcal{C}$*, and* $\mathcal{R}$*, we denote the public key space, secret key space, message space, ciphertext space, and randomness space, respectively.*

We assume w.l.o.g. that the randomness space for key generation and encryption are identical. Below we define two notions of correctness for PKE schemes.

[10] For simplicity we only consider decryption with *implicit rejection*, that is, such that the output is unspecified (or the algorithm aborts) whenever the input is not a well-formed ciphertext for the particular sk. The extension to *explicit rejection* decryption can be done for example by adding a *flag bit* that marks the output as $\perp$ whenever decryption fails.

Definition 2 (Fully Correct PKE**)** *A* PKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *is* fully correct *if it holds that*

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

$\forall\ (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}, \forall\ m \in \mathcal{M}, \forall\ r \in \mathcal{R}$.

Definition 3 ($(1-\alpha)$**-Correct** PKE**)** *A* $(1-\alpha)$-correct PKE scheme*, or* PKE with decryption error α*, is a* PKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *such that:*

$$\forall\ m \in \mathcal{M} \implies \Pr_{\substack{(\mathsf{pk},\mathsf{sk})\leftarrow\mathtt{KGen} \\ r\leftarrow\mathcal{R}}}[\mathtt{Dec}_{\mathsf{sk}}(\mathtt{Enc}_{\mathsf{pk}}(m; r)) \neq m] \leq \alpha.$$

3 Quantum Indistinguishability for PKE Schemes

In this section we extend the QS2 security notion of qIND-qCPA introduced for SKE schemes in [22] to the public key case. This is much more complex than the symmetric case, for the following reasons:

1. all the schemes are randomized in order to achieve chosen plaintext security;
2. when derandomizing the encryption procedure and considering the randomness as additional input, there might be collisions (different randomnesses leading to the same ciphertext), hence ensuring reversibility of type-2 operators is not straightforward;
3. many existing schemes, such as lattice- or code-based NIST candidates, suffer from a small decryption failure probability;
4. there is the problem of defining which type of encryption unitary (type-1 or type-2) should be modelled by the learning CPA phase.

In particular, as we will see, there are two main consequences: (1) the inverse of type-2 encryption operators is not generally a type-2 decryption operator; and (2), most interestingly, some type-2 encryption operators can be built efficiently by using only knowledge of the public key. The last point is crucial: it shows that in the PKE case, type-2 encryption operators are much more natural than in the SKE case, and for certain schemes they are actually covered in the usual notion of QS1 "post-quantum" security already. We will also show that some of these schemes are very natural, such as LWE-based schemes used as a blueprint for many NIST submissions.

In this section we will do the following:

1. First, we revisit and define formally type-1 operators for PKE, and we show the difference between type-1 encryption and decryption (cf. Section 3.1).
2. We define type-2 operators for fully correct public key encryption schemes, and we show that they can be efficiently implemented with knowledge of secret and public key (cf. Section 3.2).

3. We define what we call *recoverable* PKE schemes, i.e., schemes that admit an efficient procedure to recover the message given randomness, ciphertext and public key, without the secret key. We show that for such schemes the 'canonical' type-2 encryption operator can be built by only using the public key, *even* if the scheme is not fully correct (cf. Section 3.3).
4. We define the qIND-qCPA security notion for any PKE scheme where one can efficiently build the type-2 encryption operator. This includes in particular fully correct and recoverable schemes (cf. Section 3.4).
5. Finally, we discuss how to extend these results to the *chosen ciphertext attack* (CCA) scenario (cf. Section 3.5).

3.1 Type-1 Operators for PKE

Recall that, for an arbitrary function $f : \mathcal{X} \to \mathcal{Y}$, the corresponding type-1 operator is the "canonical" way of computing f on a superposition of input through the unitary operator $U_f : \mathfrak{H}_{\mathcal{X}} \to \mathfrak{H}_{\mathcal{Y}}$ defined by: $U_f : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$. Realizing U_f is always efficient if f is efficiently computable.

Traditionally, when looking at (deterministic) encryption schemes, the type-1 operator for encryption has been defined as:

$$U_{\mathtt{Enc}} : |m, y\rangle \mapsto |m, y \oplus \mathtt{Enc}(m)\rangle \ .$$

This is the approach used in, e.g., [13] and [22]. However, in our case of PKE schemes (which are generally randomized), we have to consider that encryption can be performed locally by the quantum adversary, who therefore has full control not only on the randomness used for encryption (i.e., it is necessary to explicitly derandomize the encryption procedure[11]), but also on the public key used (i.e., it is theoretically possible to compute encryption for a superposition of different public keys). Therefore, the most general definition of a type-1 encryption operator would look like:

$$U_{\mathtt{Enc}} : |\mathsf{pk}, r, m, y\rangle \mapsto |\mathsf{pk}, r, m, y \oplus \mathtt{Enc}_{\mathsf{pk}}(m; r)\rangle \ .$$

We argue that this is the most general and correct way to model the local computational power of a quantum adversary, even in the QS1 case. However, for ease of exposition (and also because it would go beyond the traditional meaning of ciphertext indistinguishability), in the present work we do not consider superpositions of public keys, as we assume that the public key to be attacked is given to the adversary at the beginning of the security game. Moreover notice that, as observed in [13], it is equivalent whether considering the same randomness r for every message m in the superposition, or considering different randomnesses sampled independently for every message, as the latter case can be simulated by using a QS2-secure pseudorandom function [42] that takes as argument a message m and a unique global random seed r. This leads us to the following definition.

[11] This is implicitly considered in [13] and [22], but not explicitly formalized.

Definition 4 (Type-1 Encryption for PKE**)** *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a* PKE *scheme and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$*. The* type-1 encryption operator *for* pk *is the unitary defined by:*

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

Usually the public key is understood, so we will omit that dependency and just write $U^{(1)}_{\mathtt{Enc}}$. As usual, when there is no ambiguity, we identify the corresponding superoperator acting on mixed states rather than pure states with the same symbol $U^{(1)}_{\mathtt{Enc}} : \mathfrak{D}(\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{M}} \otimes \mathfrak{H}_{\mathcal{C}}) \to \mathfrak{D}(\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{M}} \otimes \mathfrak{H}_{\mathcal{C}})$.

The type-1 decryption operator is defined analogously, but with an important difference: the decryption algorithm does not need to take as additional input the randomness used for encryption.

Definition 5 (Type-1 Decryption for PKE**)** *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a* PKE *scheme and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$*. The* type-1 decryption operator *for* sk *is the unitary defined by:*

$$U^{(1)}_{\mathtt{Dec}_{\mathsf{sk}}} : |c, z\rangle \mapsto |c, z \oplus \mathtt{Dec}_{\mathsf{sk}}(c)\rangle \ .$$

As usual we denote it by $U^{(1)}_{\mathtt{Dec}}$ leaving the secret key understood, and when there is no ambiguity with the same symbol we denote the superoperator acting on mixed states also by $U^{(1)}_{\mathtt{Dec}} : \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{M}}) \to \mathfrak{D}(\mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{M}})$.

Notice the difference in type-1 encryption and decryption acting on different spaces: this is not surprising, as it is already known that the adjoint of a type-1 encryption operator is not, generally, a type-1 decryption operator. Notice also how both operators are efficiently computable, because `Enc` and `Dec` are efficient algorithms. The difference is that realizing $U^{(1)}_{\mathtt{Dec}}$ requires knowledge of the secret key sk, while for realizing $U^{(1)}_{\mathtt{Enc}}$ it is sufficient to know pk.

3.2 Type-2 Encryption for PKE

When defining type-2 encryption for PKE schemes, we have to remember that defining these operators only makes sense for functions which are reversible. If a PKE scheme is fully correct, then encryption is always reversible if seen as a function of the *plaintext*, but not necessarily as a function of the *randomness*. That is because it might be the case that for a given message different randomnesses lead to the same ciphertext. In the context of security games, plaintext and randomness have very different roles anyway, because one is generally chosen by the adversary, while the other is generally chosen by the challenger.

Ultimately, what we want is to define a type of unitary which generalizes the case of arbitrary permutations from plaintext to ciphertext spaces (the same approach as considered in [22]). In order to avoid the issue raised by randomness collisions, we will keep the auxiliary randomness register both in input and output of the circuit. This ensures reversibility of the operator, because given a certain ciphertext and a certain randomness, there is only one possible plaintext

which was mapped to that ciphertext (otherwise we would have a decryption error, and for now we are only considering fully correct schemes). So, if the sizes of the plaintext space and the ciphertext space coincide, i.e., there is no ciphertext expansion and thus $\dim(\mathfrak{H}_{\mathcal{M}}) = \dim(\mathfrak{H}_{\mathcal{C}})$, then we can define the corresponding type-2 encryption operator as:

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

where, as usual, the public key pk is implicit in the definition of $U^{(2)}_{\mathtt{Enc}}$, i.e., it is a parameter of the unitary operator in question.

In the more general case of message expansion, i.e., $\dim(\mathfrak{H}_{\mathcal{M}}) < \dim(\mathfrak{H}_{\mathcal{C}})$, we use the same approach as in [22]: we introduce an auxiliary register in a complementary space $\mathfrak{H}_{\mathcal{C}-\mathcal{M}}$[12] that ensures reversibility of the operation, and which is initialized to $|0 \dots 0\rangle$ during an honest execution to yield a correct encryption. So we consider a family of unitary superoperators of the form:

$$\begin{aligned} U : \mathfrak{D}(\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{M}} \otimes \mathfrak{H}_{\mathcal{C}-\mathcal{M}}) \to \mathfrak{D}(\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{C}}), \text{ such that} \\ U : |r, m, y\rangle\langle r, m, y| \mapsto \psi \, , \end{aligned}$$

and we define a type-2 encryption operator any arbitrary, efficiently computable (purified) representative of the above family such that:

$$U^{(2)}_{\mathtt{Enc}} : |r, m, 0 \dots 0\rangle \mapsto |r, \mathtt{Enc}_{\mathsf{pk}}(m; r)\rangle \, . \tag{1}$$

The choice of the particular representative is irrelevant in our exposition as long as it respects (1) above and it is efficiently computable. However, as already discussed in [22], it might be the case that realizing this operator requires knowledge of the secret key, not only of the public key. This finally leads to the following.

Definition 6 (Type-2 Encryption for PKE**)** *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a fully correct* PKE *scheme, and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$*. A* type-2 encryption operator *for* Σ *is an efficiently computable unitary in the family defined by:*

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

It will be usually denoted by just $U^{(2)}_{\mathtt{Enc}}$ *when there is no ambiguity.*

It is always possible to find and efficiently sample and implement at least one valid representative for $U^{(2)}_{\mathtt{Enc}}$ given the secret and public keys, by using a conversion circuit of type-1 encryption and decryption operators in a similar way as presented in [22]. We call this the *canonical* type-2 operator.

[12] We denote by $\mathfrak{H}_{\mathcal{C}-\mathcal{M}}$ a Hilbert space such that $\mathfrak{H}_{\mathcal{C}-\mathcal{M}} \otimes \mathfrak{H}_{\mathcal{M}}$ is isomorphic to $\mathfrak{H}_{\mathcal{C}}$. Notice that the opposite case, i.e., $\dim(\mathfrak{H}_{\mathcal{M}}) > \dim(\mathfrak{H}_{\mathcal{C}})$, cannot happen because it would lead to collisions on the ciphertexts and thus introduce decryption failures. Also notice that, as in [22], the case of adversarially-controlled ancilla qubits is left as an open problem.

Theorem 7 (Efficient Realization of Type-2 Encryption) *Let Σ be a fully correct* PKE *scheme with $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$, and let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. Then there exists an efficient procedure which takes pk and sk as input, and outputs a polynomial-size quantum circuit realizing $U^{(2)}_{\mathtt{Enc}}$.*

Proof. The explicit circuit of the procedure is shown in Fig. 2. It uses type-1 encryption and decryption operators as underlying components, which are both efficient with knowledge of the respective keys. □

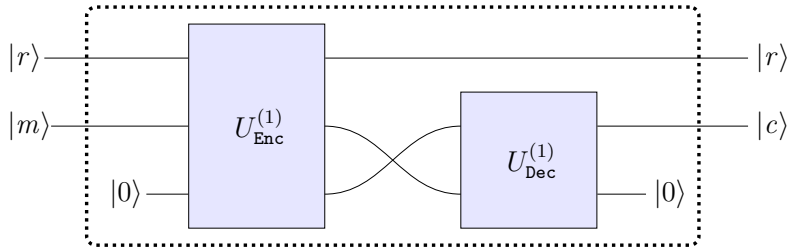

Fig. 2: Canonical type-2 encryption operator for fully correct PKE schemes.

Notice that realizing this canonical type-2 operator requires knowledge of the secret key, even if it is just an encryption operator, but that's fine because as previously mentioned type-2 operators are more powerful than type-1 ones, and usually require this additional knowledge.

3.3 Recoverable PKE Schemes

Now we introduce a special case of PKE schemes where it is possible to decrypt a ciphertext *without* knowledge of the secret key, but having access to the randomness used for the encryption instead. These schemes might not be fully correct, so the decryption procedure might fail on some ciphertext, yet still the recovery procedure will 'decrypt' correctly if the right randomness is provided. We will see in Section 4 that many natural PKE schemes are actually of this type.

Definition 8 (Recoverable PKE **Scheme)** *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a (not necessarily fully correct)* PKE *scheme. We call Σ a* recoverable PKE scheme *if there exists an efficient algorithm $\mathtt{Rec} : \mathcal{P} \times \mathcal{R} \times \mathcal{C} \to \mathcal{M}$ such that, for any $m \in \mathcal{M}, r \in \mathcal{R}, \mathsf{pk} \in \mathcal{P}$, it holds that*

$$\mathtt{Rec}(\mathsf{pk}, r, \mathtt{Enc}_{\mathsf{pk}}(m; r)) := m\,.$$

Notice how the recovery procedure will always allow to avoid decryption failures even for schemes which do not have full correctness. We will sometimes write a recoverable scheme $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ with recovery algorithm $\mathtt{Rec}$ directly as $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Rec})$. Given pk, it is of course possible to define a type-1 operator for $\mathtt{Rec}$ in the canonical way.

Definition 9 (Type-1 Recovery for PKE) *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Rec})$ *be a recoverable* PKE *scheme, and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. *The* type-1 recovery operator *for* pk *is the unitary defined by:*

$$U^{(1)}_{\mathtt{Rec}_{\mathsf{pk}}} : |r, c, z\rangle \mapsto |r, c, z \oplus \mathtt{Rec}_{\mathsf{pk}}(r, c)\rangle \ .$$

As usual we will denote this operator by $U^{(1)}_{\mathtt{Rec}}$ when there is no ambiguity in the choice of pk, and with the same symbol we denote the superoperator acting on mixed states, i.e., $U^{(1)}_{\mathtt{Rec}} : \mathfrak{D}(\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{M}}) \to (\mathfrak{H}_{\mathcal{R}} \otimes \mathfrak{H}_{\mathcal{C}} \otimes \mathfrak{H}_{\mathcal{M}})$.

Now, the crucial observation is the following: for recoverable PKE schemes, the canonical type-2 encryption operator can be efficiently implemented without knowledge of the secret key!

Theorem 10 (Type-2 Encryption Operator for Recoverable Schemes) *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec}, \mathtt{Rec})$ *be a recoverable* PKE*, and let* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$. *Then there exists an efficient procedure which only takes* pk *as input, and outputs a polynomial-size quantum circuit realizing the canonical operator* $U^{(2)}_{\mathtt{Enc}}$.

Proof. The explicit circuit of the procedure is shown in Fig. 3. It uses type-1 encryption and recovery operators as underlying components, which are both efficient with knowledge of the public key only. Realization of both these components is independent of the fact whether the scheme has full correctness or not, as the decryption algorithm itself is never used. □

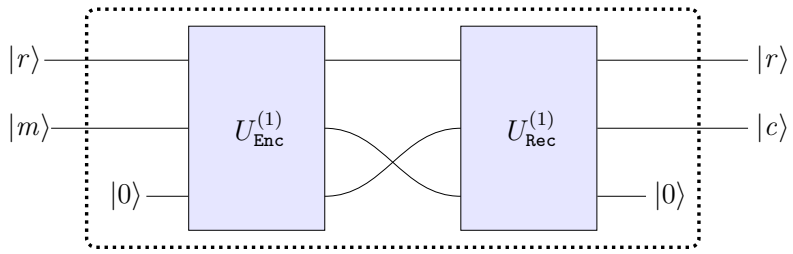

Fig. 3: Canonical type-2 encryption operator for recoverable PKE schemes.

In particular, for recoverable PKE schemes the type-2 encryption operator can be realized locally by a quantum adversary (or a reduction), without need of additional oracle access. This, together with the fact that many real-world PKE schemes are recoverable (as we will see) shows that type-2 encryption operators are very natural, and unlike in the symmetric key case considered in [22] they also appear implicitly in QS1 security notions for such schemes.

3.4 The qIND-qCPA Security Notion

We are now ready to define the notion of *quantum ciphertext indistinguishability under quantum chosen plaintext attack* (qIND-qCPA) for PKE schemes

which admit an efficient construction of the canonical type-2 encryption operator $U^{(2)}_{\mathtt{Enc}}$. This includes in particular fully correct schemes and recoverable schemes.[13] We follow the approach in [22] and we define a game where a polynomially bounded quantum adversary plays against an external challenger. We have to define the challenge phase and the learning (quantum CPA) phases (pre- and post-challenge), using the theory of type-2 operators we have devised so far.

For the challenge query it is pretty straightforward: as in the original qIND security definition for symmetric key encryption, we assume that the challenger $\mathcal{C}$ generates a keypair and sends the public key pk to the adversary $\mathcal{A}$. Then $\mathcal{A}$ sends two plaintext quantum states (possibly mixed) φ_0, φ_1 to $\mathcal{C}$, who will flip a random bit $b \leftarrow \{0,1\}$, discard (trace out) φ_{1-b}, and encrypt the other message with the type-2 encryption operator $\psi \leftarrow U^{(2)}_{\mathtt{Enc}}\varphi_b$. Finally, ψ is sent back to $\mathcal{A}$, who will have to guess b in order to win the game.

Justifying the use of a type-2 encryption during the challenge phase requires arguments different from the symmetric key case. In the classical IND-CPA game for PKE, the challenger might 'forget' the secret key immediately after generation (as it is not needed for encryption), and we saw already that the secret key is sometimes necessary to implement the canonical type-2 encryption operator. However, we want a security notion that is general, in that it does not require an extra assumption such as 'the challenger forgets the secret key'. Moreover we also saw how certain schemes, like the recoverable ones, allow to build the type-2 operator without the secret key. Thus it makes sense for a QS2 security notion to include the use of type-2 operators during the challenge phase.

There is an important caveat here though: looking at Definition 6, we must deal with the randomness register $|r\rangle$ which is both input and output of $U^{(2)}_{\mathtt{Enc}}$. The challenger cannot send this register back to $\mathcal{A}$, because we have already seen that in many schemes (e.g., the recoverable ones) this would allow $\mathcal{A}$ to decrypt the challenge ciphertext, and it would also be definitely unnatural. But at the same time if the challenger withholds the randomness register, from $\mathcal{A}$'s perspective this would be equivalent to tracing it out, and if the type-2 encryption operator introduces entanglement between ciphertext and randomness output registers, then tracing out the randomness would disturb the ciphertext state.

Luckily, there is a simple solution to this dilemma: since the randomness is chosen by the (honest) challenger during the challenge query, we can safely assume it is always going to be classical[14]. Looking at Definition 6, this means that the input global state is always going to be separable as $|r\rangle \otimes \varphi_b \otimes |0\dots0\rangle$, hence the output state is also going to be separable as $|r\rangle \otimes \psi$. Therefore, in our qIND query definition the challenger can discard the randomness register after

[13] As we will see, these cover all the interesting cases in practice, although there might be other classes of schemes which allow efficient construction of $U^{(2)}_{\mathtt{Enc}}$; we address the general case in Section 6

[14] Even if considering challengers that use superpositions of randomnesses, we show in Supplementary Section B that the difference is irrelevant, and that we can always restrict ourselves to the case of a classical randomness register.

applying the type-2 encryption without disturbing the ciphertext state. Thus there is no problem in returning only the ciphertext register to $\mathcal{A}$.

The other question we have to address, which was left unspecified in [22], is about the learning (qCPA) phase. Shall the adversary be able to perform only type-1 encryption operations, or type-2 as well? In the QS1 case the answer is obvious: it depends on the scheme, e.g., for recoverable schemes both type-1 and type-2 operations should be allowed, but in the general case only type-1 operations should. Instead, in the QS2 case that we are considering, the answer is less straightforward. For recoverable schemes again there is no difference, as the adversary can implement both types of operators locally. But for general schemes there might be a difference, and there might exist non-recoverable schemes which become insecure when giving oracle access to a type-2 encryption operator.[15]

In our definition of qIND-qCPA we opt for giving to the adversary as much power as possible, hence explicitly giving access to a type-2 encryption oracle when dealing with non-recoverable schemes. The reason for this choice is threefold. First, this allows us to aim for potentially stronger security notions. Second, remember that, classically, CPA attacks model not only the case where the adversary can compute ciphertexts himself (as in the case of PKE), but also scenarios where the adversary can "trick" an honest encryptor in providing certain ciphertexts (as in the case of IND-CPA security for symmetric key encryption). In the quantum PKE setting, there is a difference whether these ciphertexts are computed locally by the adversary or obtained by the challenger through "trickery" (including scenarios already considered in [22], such as quantum side-channel attacks, quantum obfuscation, etc.), because the challenger has knowledge of the secret key, and is therefore capable of generating type-2 ciphertexts even if the scheme is not recoverable. So, giving the adversary access to the type-2 encryption oracle seems to be the "safe" choice. The last reason is that, when switching from CPA to CCA scenarios, the adversary will get oracle access to at least a type-1 decryption oracle, which in turn allows to implement a type-2 encryption operator as explained in Theorem 7. In that case, having already provided the adversary with this oracle will simplify the notation and help in obtaining more meaningful security results (i.e., if a CPA-secure scheme becomes insecure in the CCA case, intuitively it should be because of the additional power to decrypt ciphertexts, and not because a new kind of ciphertexts can be generated).

All these considerations finally lead to the following.

Experiment 11 *The* qIND-qCPA experiment $\mathsf{qIND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda)$ *for a* PKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *is defined as follows:*

1: $\mathcal{C}$ *runs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$
2: $\mathcal{A}^{U_{\mathtt{Enc}}^{(2)}}(\mathsf{pk}) \rightarrow (\varphi_0, \varphi_1, \sigma_{state})$
3: $\mathcal{C}$ *receives* φ_0, φ_1 *and does the following:*
- *flips* $b \xleftarrow{\$} \{0,1\}$
- *samples* $r \xleftarrow{\$} \mathcal{R}$
- *traces out* φ_{1-b}

[15] We leave finding such a counterexample as an open problem.

- *applies* $U_{\mathtt{Enc}}^{(2)}\left(|r\rangle\langle r| \otimes \varphi_b\right) \to |r\rangle\langle r| \otimes \psi$
- *traces out* $|r\rangle\langle r|$
- *sends* ψ *to* $\mathcal{A}$

4: $\mathcal{A}^{U_{\mathtt{Enc}}^{(2)}}(\sigma_{state}, \psi) \to b' \in \{0,1\}$

5: ***if*** $b = b'$ ***then return*** *win;* ***else return*** *rej.*

Security is defined as negligible advantage over guessing.

Definition 12 (qIND-qCPA **Security)** *A public key encryption scheme Σ has* quantum ciphertext indistinguishability under quantum chosen plaintext attack*, or is* qIND-qCPA*-secure, iff for any QPT adversary $\mathcal{A}$ it holds:*

$$\left|\Pr\left[\mathsf{qIND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda) \to \mathit{win}\right] - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda)\,.$$

It is easy to show that the above notion is at least as strong as the QS1 notion of IND-qCPA for PKE introduced in [13].

Theorem 13 (qIND-qCPA $\Rightarrow$ IND-qCPA**)** *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a* PKE *scheme. If Σ is* qIND-qCPA*-secure, then it is* IND-qCPA*-secure.*

Proof. We will show that any adversary $\mathcal{A}$ wins the qIND-qCPA game with at least the same probability of winning the IND-qCPA game; the latter (Experiment 26) is described in Section A. The differences with Experiment 11 are:

1. in the IND-qCPA game $\mathcal{A}$ does not get oracle access to $U_{\mathtt{Enc}}^{(2)}$. Hence, when switching to qIND-qCPA, the winning probability cannot decrease, because the power of the adversary is augmented by the type-2 oracle.
2. In the IND-qCPA game $\mathcal{A}$ is restricted to classical challenge messages m_0, m_1. When switching to qIND-qCPA, the adversary will simply use quantum states $|m_0\rangle, |m_1\rangle$ as challenge plaintexts instead, and will measure the quantum ciphertext received by the challenger.

Notice in fact that, since the randomness r in the qIND-qCPA challenge query is classical, the type-2 operator $U_{\mathtt{Enc}}^{(2)}$ will produce a ciphertext state which is just a classical ciphertext encoded as a basis state $|c = \mathtt{Enc}_{\mathsf{pk}}(m; r)\rangle$. In other words, quantum plaintexts are *more generic* than classical plaintexts (or, to put it differently, classical plaintexts are a very special case of quantum plaintexts), and hence again the power of the adversary is not diminished when switching to the qIND-qCPA game. This implies that any adversary winning the IND-qCPA game with non-negligible advantage over guessing can also win the qIND-qCPA game with at least the same advantage. □

3.5 The CCA Case

We leave the case of extending our exposition to the quantum chosen ciphertext attack case (with the relevant notions of qIND-qCCA1 and qIND-qCCA2) as future work, but we want anyway to sketch here the general strategy.

The first task is to formalize a type-2 operator for decryption. Unlike in the symmetric key setting considered in [22], this is not necessarily going to be the adjoint of the type-2 encryption operator, and in particular it might not require a randomness register as input; this has to be expected given that there is already an asymmetry in the definition of type-1 encryption and decryption operators in the public key setting. Then, in the qIND-qCCA1 case, we just extend the qIND-qCPA experiment by also providing the adversary with oracle access to the type-1 and type-2 decryption operators.

Extending the framework to the qIND-qCCA2 case is not straightforward, mainly due to no-cloning and the destructive nature of quantum measurement. In fact, this case was left as an open problem already in [22] for the symmetric key setting. Fortunately, the technique presented in [5] shows how to overcome this difficulty, by using a real-VS-ideal approach which makes it possible to differentiate the behaviour of the adversary when replaying the challenge ciphertext to the decryption oracle, hence effectively detecting a challenge replay attack. The approach in [5] (and its extension to the public key case presented in [4]) is given in the context of *quantum encryption schemes* (a scenario which falls under the QS3 domain in [21]), but it is easy to generalize to the QS2 notions we are considering here.

4 Achievability Result

In this section we show that our QS2 security notion qIND-qCPA is achievable. We show this for the canonical LWE-based encryption scheme, which follows the blueprint by Regev [37]. This canonical scheme is the core idea underlying lattice-based PKE schemes such as Kyber [14], LIMA [7], the Lindner-Peikert scheme [31], and the schemes underlying NewHope [36] and LAC [32]. These concrete schemes typically deploy some tweaks to this generic blueprint.[16] The pseudocode (that we give for simplicity in a generic form, i.e., not specifying concrete domains and distributions for the parameters) is given in Fig. 4.

Recall, from the foregoing discussion, that qIND-qCPA security can only be defined for schemes which admit an efficient realisation of a type-2 encryption operator, which is why we show this first. The required property directly follows from Theorem 10, provided that the canonical LWE-based encryption scheme is recoverable. Hence, in the lemma below we show that the canonical LWE-based encryption scheme is recoverable.

Lemma 14 *The canonical* LWE-*based* PKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$*, shown in Fig. 4, is recoverable as from Definition 8.*

Proof. To prove the statement, we have to specify the algorithm `Rec` that is introduced in Definition 8. Its input is a public key $\mathsf{pk} = (\mathsf{a}, \mathsf{b})$, a randomness r, and a ciphertext $c = (c_1, c_2)$ such that c corresponds to the encryption of

[16] Hence, to formally prove these schemes qIND-qCPA-secure, it should be verified that these tweaks do not thwart our proof for the canonical scheme.

$\mathtt{KGen}(\lambda; r)$	$\mathtt{Enc}(\mathsf{pk}, m; r)$	$\mathtt{Dec}(\mathsf{sk}, c)$
$\mathsf{a}, \mathsf{s}, \mathsf{e} := r$	**parse** pk **as** (a, b)	**parse** sk **as** s
$\mathsf{b} := \mathsf{as} + \mathsf{e}$	$\mathsf{e}_1, \mathsf{e}_2, \mathsf{d} := r$	**parse** c **as** (c_1, c_2)
$\mathsf{pk} := (\mathsf{a}, \mathsf{b})$	$c_1 := \mathsf{bd} + \mathsf{e}_1 + \mathsf{Encode}(m)$	$m := \mathsf{Decode}(c_1 - c_2\mathsf{s})$
$\mathsf{sk} := \mathsf{s}$	$c_2 := \mathsf{ad} + \mathsf{e}_2$	**return** m
return $(\mathsf{sk}, \mathsf{pk})$	**return** $c := (c_1, c_2)$	

Fig. 4: Pseudocode of the canonical LWE-based public key encryption scheme $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$. For the randomness r used by $\mathtt{KGen}$ and $\mathtt{Enc}$, let $x := r$ denote that x is deterministically derived from r.

a message m, using the public key pk and randomness r. The algorithm $\mathtt{Rec}$ proceeds as follows. Given the randomness r, it obtains the same values e_1, e_2, and d that have been derived from r during encryption. Finally, the algorithm $\mathtt{Rec}$ outputs

$$\begin{aligned}\mathsf{Decode}(c_1 - \mathsf{bd} - \mathsf{e}_1) &= \mathsf{Decode}(\mathsf{bd} + \mathsf{e}_1 + \mathsf{Encode}(m) - \mathsf{bd} - \mathsf{e}_1) \\ &= \mathsf{Decode}(\mathsf{Encode}(m)) = m\,.\end{aligned}$$

This concludes the proof. □

From Theorem 10 and Lemma 14 it follows that type-2 encryption operators can be efficiently implemented for the canonical LWE-based PKE scheme.

Corollary 15 *For the public key encryption scheme $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$, shown in Fig. 4, the type-2 encryption operator can be efficiently realised without requiring the secret key.*

Having established that qIND-qCPA security *can* be defined for the canonical LWE-based encryption scheme, we turn towards *proving* it qIND-qCPA-secure. To this end, we first recall the QS1 security proof of the scheme, see for instance [31], since the same proof idea carries over to the QS2 setting.

The QS1 security proof consists of three games G_0, G_1, and G_2. Game G_0 is the security game for classical IND-CPA security instantiated with the scheme. Game G_1 is the same, except that b, the LWE sample in the public key, is replaced by a random value. Game G_2 is like G_1, except that the two LWE samples generated during encryption, i.e., $\mathsf{bd} + \mathsf{e}_1$ and $\mathsf{ad} + \mathsf{e}_2$, are replaced by random samples. Recall that the LWE problem asks to distinguish whether a tuple (a, b) is sampled randomly or whether $\mathsf{b} = \mathsf{as} + \mathsf{e}$, for unknown s and e. It is fairly easy to see that the adversarial advantage in distinguishing between G_0 and G_2 is bound by the advantage in solving LWE. Combined with the fact that the adversary has no advantage in G_2, this concludes the QS1 security proof.

To lift the security proof to our QS2 notion, we enhance each of the games by using a type-2 encryption operator to encrypt (in superposition) one of the messages by the adversary. Based on this, G_0 corresponds to the qIND-qCPA

security game while G_2 is a game in which the adversary still has no advantage. The reason is that the message is encrypted by adding a random value to it. This allows us to bound the qIND-qCPA advantage of an adversary by its advantage in distinguishing between G_0 and G_2. The crucial observation is that the reduction injects its challenge (received from the LWE oracle) into the public key (first game hop, i.e., from G_0 to G_1) and into the randomness (second game hop, i.e., from G_1 to G_2). Since both the public key and the randomness remain classical in our QS2 notion, the same approach from the QS1 setting holds. However, the reduction has to be able to simulate the correct games for the adversary, which essentially means it has to efficiently implement the type-2 operator in *all* games. This is important in particular for the second game hop between G_1 and G_2. For this game hop, the canonical type-2 implementation does not work as there does not even exist a secret key any more. Corollary 15 states that the reduction can efficiently implement the type-2 operator not only when the reduction does not know the secret key, but also when such a secret key does not exist at all.

We are now ready to state the main theorem of this section. It proves the qIND-qCPA security of the canonical LWE-based public key encryption scheme.

Theorem 16 *Assuming the quantum hardness of the* LWE *problem, the public key encryption scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$*, displayed in Fig. 4, is* qIND-qCPA*-secure according to Definition 12.*

Proof. The proof consists of three games G_0, G_1, and G_2, where G_0 is the qIND-qCPA game as described in Experiment 11 instantiated with the encryption scheme depicted in Fig. 4. In G_1, the value b from the public key is replaced by a random value. In G_2, also the values $\mathsf{bd} + \mathsf{e}_1$ and $\mathsf{ad} + \mathsf{e}_2$ are replaced by random values.

Any adversary $\mathcal{A}$ that distinguishes between G_0 and G_1 can be transformed into an adversary $\mathcal{B}$ against the LWE problem. The adversary $\mathcal{B}$ behaves as an honest challenger apart from using its LWE challenge as the public key, which it also sends to $\mathcal{A}$ at the start of the game. Likewise, any adversary $\mathcal{A}$ that distinguishes between G_1 and G_2 can be transformed into an adversary $\mathcal{C}$ against the LWE problem. The difference is that $\mathcal{C}$ receives two LWE samples which it uses as the public key and the randomness during encryption. For both reductions, efficient implementations follow from Corollary 15. Since no adversary has an advantage in G_2, this proves the claim. □

At this point, we also note that the code-based PKE schemes which underlie the NIST proposals BigQuake [18], ROLLO-II [9], HQC [1], and RQC [2] are recoverable as well. The proof is straightforward just as for the canonical LWE-base encryption scheme (cf. Lemma 14). Regarding the QS2 security of these schemes, we stress that they follow a very similar proof idea. That is, they do a game hop in which a part of the public key is changed and another one in which a part of the randomness used during the qIND phase is replaced. Based on this, it is reasonable to assume that they are also qIND-qCPA-secure.

5 Separation Result

In this section we provide a separation example to show that our security notion qIND-qCPA is strictly stronger than the security notion IND-qCPA from [13]. We also show that allowing superpositions of randomness in the challenge query does not affect this separation, thus reinforcing our motivation for only considering challengers who generate a classical randomness.

Our separation example is the standard hybrid PKE-SKE encryption scheme, combining a public key encryption scheme and a symmetric key encryption scheme. That is, a message is encrypted using a fresh one-time key of the symmetric key encryption scheme and then this one-time key is first encrypted using the public key encryption scheme and then attached to the ciphertext. To decrypt, one first recovers the one-time symmetric key, and then uses it to decrypt the ciphertext containing the message. For additional background on symmetric key encryption schemes and the mentioned security notions, see Appendix A. The standard hybrid encryption scheme is shown in Fig. 5. We start with the lemma

$\mathtt{KGen}(\lambda)$	$\mathtt{Enc}_{\mathsf{pk}}(m;r)$	$\mathtt{Dec}_{\mathsf{sk}}(c)$
$(\mathsf{pk},\mathsf{sk}) \leftarrow \mathtt{KGen}^P(\lambda)$	**parse** r **as** (r_1,r_2,r_3)	**parse** c **as** (c_1,c_2)
return $(\mathsf{pk},\mathsf{sk})$	$\mathsf{k} \coloneqq \mathtt{KGen}^S(\lambda;r_1)$	$\mathsf{k} \coloneqq \mathtt{Dec}^P_{\mathsf{sk}}(c_2)$
	$c_1 \coloneqq \mathtt{Enc}^S_{\mathsf{k}}(m;r_2)$	$m \coloneqq \mathtt{Dec}^S_{\mathsf{k}}(c_1)$
	$c_2 \coloneqq \mathtt{Enc}^P_{\mathsf{pk}}(\mathsf{k};r_3)$	**return** m
	return (c_1,c_2)	

Fig. 5: Hybrid PKE scheme $\Sigma = (\mathtt{KGen},\mathtt{Enc},\mathtt{Dec})$ built from a PKE scheme $\Sigma^P = (\mathtt{KGen}^P,\mathtt{Enc}^P,\mathtt{Dec}^P)$ and a SKE scheme $\Sigma^S = (\mathtt{KGen}^S,\mathtt{Enc}^S,\mathtt{Dec}^S)$.

below which shows that the hybrid public key encryption scheme is IND-qCPA-secure if the underlying symmetric key and public key encryption schemes are IND-qCPA-secure (cf. Definition 31 and 27).[17]

Lemma 17 *Let $\Sigma = (\mathtt{KGen},\mathtt{Enc},\mathtt{Dec})$ be the hybrid PKE scheme built as shown in Fig. 5 from a SKE scheme $\Sigma^S = (\mathtt{KGen}^S,\mathtt{Enc}^S,\mathtt{Dec}^S)$ and a PKE scheme $\Sigma^P = (\mathtt{KGen}^P,\mathtt{Enc}^P,\mathtt{Dec}^P)$. If both Σ^S and Σ^P are* IND-qCPA*-secure, then Σ is* IND-qCPA*-secure.*

Proof. The proof is standard for this hybrid PKE scheme, so we just recall here a brief sketch. The proof uses one game hop, which is bound by the IND-qCPA security of the PKE scheme Σ^P. In this hop, the ciphertext part c_2 is replaced by encrypting a random key k' rather than k. The adversarial advantage in the resulting game can be straightforwardly bound by the IND-qCPA security of the SKE scheme Σ^S. □

[17] Actually it is sufficient that the SKE scheme is one-time quantum-secure, but for simplicity we will restrict to the IND-qCPA case.

To show that the hybrid scheme is qIND-qCPA insecure, we use the results of [22] and exploit the insecurity of the underlying SKE scheme when the indistinguishability phase is quantum. We first recall the necessary results from [22].

Definition 18 (Core Function [22]) *Let* $(\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a* SKE *scheme. We call the function* $f\colon \mathcal{K} \times \{0,1\}^\tau \times \mathcal{M} \to \mathcal{Y}$ *the* core function *of the encryption scheme, if for some* $\tau \in \mathbb{N}$*:*

- *for all* $\mathsf{k} \in \mathcal{K}$ *and* $m \in \mathcal{M}$, $\mathtt{Enc}(\mathsf{k}, m)$ *can be written as* $(r, f(\mathsf{k}, r, m))$*, where* $r \in \{0,1\}^\tau$ *is independent of the message; and*
- *there exists a function* f' *such that for all* $\mathsf{k} \in \mathcal{K}$*,* $r \in \{0,1\}^\tau$*,* $m \in \mathcal{M}$*, we have:* $f'(\mathsf{k}, r, f(\mathsf{k}, r, m)) = m$*.*

Definition 19 (Quasi-Length-Preserving Encryption [22]) *We call a symmetric key encryption scheme with core function* f quasi-length-preserving *if*

$$\forall m \in \mathcal{M}, r \in \{0,1\}^\tau, \mathsf{k} \in \mathcal{K} : |f(\mathsf{k}, r, m)| = |m|,$$

i.e., if the output of the core function has the same bit length as the message.

Theorem 20 (Impossibility Result [22]) *No quasi-length-preserving symmetric key encryption scheme can be* qIND-qCPA*-secure.*[18]

On a high-level, Theorem 20 states that any encryption scheme for which the lengths of the message and the ciphertext coincide, is insecure when allowing a quantum indistinguishability phase. The attack is based on a quantum Fourier transform distinguisher, following the observation that for these schemes the plaintext $H\,|0,\ldots,0\rangle$ is left unaltered by the encryption procedure, where H is the Hadamard transform. Under this observation, there exists an instantiation of the hybrid encryption scheme which is not qIND-qCPA-secure.

Theorem 21 *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be the hybrid* PKE *scheme built as shown in Fig. 5 from a* SKE *scheme* $\Sigma^S = (\mathtt{KGen}^S, \mathtt{Enc}^S, \mathtt{Dec}^S)$ *and a* PKE *scheme* $\Sigma^P = (\mathtt{KGen}^P, \mathtt{Enc}^P, \mathtt{Dec}^P)$*. Suppose that* Σ^P *is* qIND-qCPA*-secure and* Σ^S *is an* IND-qCPA *secure quasi-length-preserving* SKE *scheme (cf. Definition 19). Then* Σ *is not* qIND-qCPA*-secure.*

Proof. We show the insecurity by describing an adversary against the scheme. Notice the following: during the challenge phase, it is the (honest) challenger who performs the encryption through the type-2 operator $U^{(2)}_{\mathtt{Enc}}$, so he will use a classical randomness. As a consequence, the one-time symmetric key k will also be classical, hence the first part c_1 of the challenge ciphertext will be a superposition of ciphertexts encrypted under the same symmetric key. This is exactly the case for which Gagliardoni et al. [22] show that quasi-length-preserving symmetric key encryption schemes are insecure. Our attack, which exploits exactly this property of the hybrid encryption scheme, works as follows.

[18] In [22], the authors use the term qIND as their attack does not require the qCPA phase. Hence it also holds for qIND-qCPA.

First, the adversary receives the public key pk of the encryption scheme. Then it prepares two challenge messages $|m_0\rangle \coloneqq H\,|0,\ldots,0\rangle$ and $|m_1\rangle \coloneqq H\,|1,\ldots,1\rangle$ and sends them to the challenger. When receiving the challenge ciphertext $|c\rangle = |c_1, c_2\rangle$, the adversary only needs the first part c_1. It applies H to the core function part of $|c_1\rangle$, measures it in the computational basis, outputs 0 if the outcome is $|0,\ldots,0\rangle$, and 1 otherwise. Using the result from [22], this adversary can distinguish which message the challenger has encrypted. □

6 Classifying Other PKE Schemes

So far we have built a solid framework for QS2 security of PKE schemes which are fully correct or recoverable (or both). But what about schemes which do not fall in either of these two categories? Are there such examples at all? And what can we learn from this? In this section we try to classify PKE schemes in general, and try to extend our results to other classes where possible, and to point out at the difficulties in other cases.

6.1 Dealing With Decryption Failures: The General Case

First, we discuss why arbitrary non-correct PKE schemes do not allow, in general, to define a type-2 encryption operator. Consequently, we cannot define the qIND-qCPA game as from Experiment 11. However, we also discuss a possible workaround.

First of all, recall that defining a type-2 operator is only possible for functions that are inherently invertible. Then observe that a $(1-\alpha)$-correct PKE scheme (cf. Definition 3) could have arbitrary, even overwhelming error α. In the most extreme case, the scheme can be almost identical to a constant function (for example, consider a scheme where every public key pk always encrypts to 0, except for one particular randomness $\bar{r}$ where it produces a correctly decryptable ciphertext instead). In the presence of decryption failures, it is therefore impossible to find a general way to define type-2 operators for encryption, and hence, to define a suitable qIND-qCPA security notion.[19]

We call *non-isometric* such schemes, where it is simply not possible to define a unitary operator that behaves *exactly* as from Definition 6 for any keypair, even if we drop the requirement of efficiency.

Definition 22 (Non-Isometric Schemes) *Let $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ be a* PKE *scheme. We say that Σ is* non-isometric *if, for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, and for any unitary U acting on the appropriate Hilbert space of randomness, plaintext, and ancilla, there exists at least a pair $(m_{\mathsf{pk}}, r_{\mathsf{pk}})$ such that:*

$$\Pr\left[M(U\,|r_{\mathsf{pk}}, m_{\mathsf{pk}}, 0,\ldots,0\rangle) \to (r_{\mathsf{pk}}, \mathtt{Enc}_{\mathsf{pk}}(m_{\mathsf{pk}}; r_{\mathsf{pk}}))\right] < 1\,,$$

where M denotes measurement in the canonical computational base.

[19] Recoverable schemes are a special case: they might not be always reversible in the message space only, but they are always reversible in the union of message space and randomness space.

A possible workaround for these non-isometric schemes is to 'enforce' the reversibility of the encryption, obtaining a new type of encryption unitary. Consider what happens if we want to use the type-1 encryption operator (cf. Definition 4) during the challenge query:

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

As already observed, the randomness r can be understood as classical and discarded by the challenger. However, the other two registers are generally going to be entangled, and both would have to be sent to the adversary for a meaningful quantum notion; but this would clearly break security because the message would remain in clear.[20] We could try to 'fix' this issue by (reversibly) masking somehow the message register sent to the adversary, for example by using a permutation π on the message space drawn uniformly at random. The following unitary:

$$U^{(\pi)}_{\mathtt{Enc}_{\mathsf{pk}}} : |r, m, y\rangle \mapsto |r, \pi(m), y \oplus \mathtt{Enc}_{\mathsf{pk}}(m; r)\rangle$$

allows hence to define a new type of quantum challenge query, where the challenger still discards the randomness register after encryption, but sends back the other two registers to the adversary. Notice how, from the adversary's point of view, $\pi(m)$ is a completely random element, and therefore the presence of this additional register does not offer any distinguishing advantage. Moreover, in actual security reductions, the uniformly drawn π can be replaced by a quantum-secure pseudorandom permutation [22], or QPRP in short.

We can hence use these *type-π operators* to define (for *any* PKE scheme, including the non-isometric ones) a new indistinguishability game and a related security notion with quantum challenge query. Motivating the use of such operators when modelling security is arguably non-trivial. In certain cases, one could see $\pi(m)$ as some sort of side-channel information given to the adversary, but in general it looks like just an artificial way to enforce reversibility on schemes which are not. We will therefore not study the resulting security notion in this work, but we want nevertheless to make a few observations on it.

First of all, notice that such a new security notion cannot be stronger than qIND-qCPA, at least when considering correct or recoverable schemes. As a separating example, consider the distinguishing attack from Theorem 21: this will not work anymore because of the presence of the entangled $\pi(m)$ register, so that the hybrid scheme might be secure according to the new notion but still qIND-qCPA insecure.

Second, notice how the challenge query resulting from the use of type-π operators reminds of the one given in an alternative quantum indistinguishability notion proposed by Mossayebi and Shack [33] - the difference is basically producing $|m, \mathtt{Enc}_{\mathsf{pk}}(\pi(m))\rangle$ instead of $|\pi(m), \mathtt{Enc}_{\mathsf{pk}}(m)\rangle$ - which is itself not comparable qIND-qCPA. This security notion has been recently investigated and expanded by Chevalier et al. in [16].

[20] This explanation appears in detail in [22].

6.2 Refining The Classification

Now we know how to define qIND-qCPA security of PKE schemes which are fully correct or recoverable (or both), and at the same time we know that it is not possible for schemes that are non-isometric. But it turns out we can say more. First of all we make a distinction for those schemes which *are* isometric: it means that 'there exists' a unitary operator that behaves exactly as a type-2 encryption operator, but we distinguish whether finding and building such operator is efficient or not.

Definition 23 ((Computationally) Isometric Schemes) *Let Σ be a* PKE *scheme with $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$. We say that Σ is* isometric *if, for any $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$, and for any pair (m, r), there exists a unitary U acting on the appropriate Hilbert space of randomness, plaintext, and ancilla, such that:*

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

where M denotes measurement in the canonical computational base. Furthermore, we say that Σ is computationally isometric *if U can be efficiently realized.*

Then, notice how a type-2 encryption operator (as from Definition 6) satisfies the above definition of U, both by construction and by efficiency. In other words computationally isometric schemes are exactly all and only those schemes which, by definition, admit an efficient construction of the type-2 encryption operator. Clearly, in particular this includes fully correct schemes (by Theorem 7) and recoverable schemes (by Theorem 10).

Corollary 24 *Let Σ be a* PKE *scheme. If Σ is fully correct or recoverable, then it is computationally isometric.*

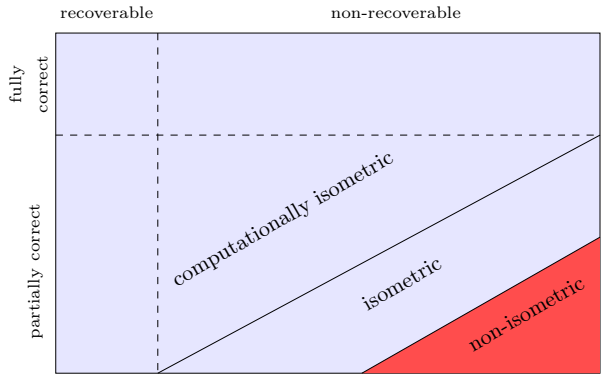

Fig. 6: Classification of PKE schemes. The qIND-qCPA security notion can be defined for all of them except those in the lower right corner, i.e., the non-isometric ones.

The situation is depicted in Fig. 6. This means that, as from Definition 12, we can extend the qIND-qCPA security notion not only to recoverable or fully correct schemes, but to all the computationally isometric ones. For the non-computational case the qIND-qCPA notion can still be defined, but its usefulness would be less clear, as it might require unbounded challengers in the security game (and therefore, difficulty in simulating them by efficient reductions when proving the security of a particular scheme). Still, it would be useful for *impossibility results*, i.e., proving that a particular isometric scheme is not qIND-qCPA-secure.

Finally, can we find representative examples of schemes which fall in the categories that we have just defined? We have already mentioned an example of non-isometric scheme at the beginning of Section 6.1 (the almost-constant one). Here we show a construction of computationally isometric scheme that is neither correct nor recoverable. The construction is given in Fig. 7: basically it transforms a recoverable, not fully correct encryption scheme by pre-processing the plaintext with a quantum-secure trapdoor function [21] before encrypting it, and inverts again the trapdoor after decryption (the public and secret keys of the trapdoor are embedded in the public and secret key, respectively, of the resulting scheme). It works because the trapdoor 'scrambles' the resulting ciphertexts but not the randomness, thereby hindering an adversary (or a challenger) who tries to build an efficient recovery algorithm $\mathtt{Rec}$ for the transformed scheme. At the same time, we show how such construction is computationally isometric, by showing an efficient circuit for the canonical type-2 encryption operator $U^{(2)}_{\mathtt{Enc}}$.

$\mathtt{KGen}(\lambda)$	$\mathtt{Enc}_{\mathsf{pk}}(m;r)$	$\mathtt{Dec}_{\mathsf{sk}}(c)$
$(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathtt{KGen}^{\Sigma}(\lambda)$	**parse** pk **as** $(\mathsf{pk}_e, \mathsf{pk}_f)$	**parse** sk **as** $(\mathsf{sk}_e, \mathsf{sk}_f)$
$(\mathsf{pk}_f, \mathsf{sk}_f) \leftarrow \mathtt{KGen}^{F}(\lambda)$	$y \coloneqq \mathtt{F}(\mathsf{pk}_f, m)$	$y \coloneqq \mathtt{Dec}^{\Sigma}(\mathsf{sk}_e, c)$
$\mathsf{pk} \coloneqq (\mathsf{pk}_e, \mathsf{pk}_f)$	$c \coloneqq \mathtt{Enc}^{\Sigma}(\mathsf{pk}_e, y; r)$	$m \coloneqq \mathtt{F}^{-1}(\mathsf{sk}_f, y)$
$\mathsf{sk} \coloneqq (\mathsf{sk}_e, \mathsf{sk}_f)$	**return** c	**return** m
return $(\mathsf{pk}, \mathsf{sk})$		

Fig. 7: Transformed scheme Γ, where $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ is a PKE scheme and $\Pi = (\mathtt{KGen}^{F}, \mathtt{F}, \mathtt{F}^{-1})$ is a deterministic trapdoor function.

Theorem 25 *Let $\Pi = (\mathtt{KGen}^{F}, \mathtt{F}, \mathtt{F}^{-1})$ be a deterministic trapdoor function and $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ be a* PKE *scheme. If Π is quantum-secure and Σ is recoverable and $(1-\alpha)$-correct, then the scheme $\Gamma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ depicted in Fig. 7 is a computationally isometric, $(1-\alpha)$-correct* PKE *scheme that is not recoverable.*

Proof. Partial correctness of the encryption scheme Γ follows immediately from the partial correctness of Σ, as permuting the plaintexts does not change the overall decryption failure probability.

Assume, for sake of contradiction, that Γ is recoverable. Then there exists an efficient algorithm $\mathtt{Rec}$ that, on input pk, r, and $\mathtt{Enc}_{\mathsf{pk}}(m;r)$, outputs m. We construct the following adversary $\mathcal{B}$ against the trapdoor function Π. He receives a public key pk_f for the function $\mathtt{F}$ along with $y \coloneqq \mathsf{pk}_f(x)$ for a random x, and is asked to find x. $\mathcal{B}$ computes $(\mathsf{pk}_e, \mathsf{sk}_e) \leftarrow \mathtt{KGen}^{\Sigma}(\lambda)$, chooses $r \leftarrow \mathcal{R}$, computes $c \coloneqq \mathtt{Enc}^{\Sigma}_{\mathsf{pk}_e}(y;r)$, and uses $\mathsf{pk} = (\mathsf{pk}_e, \mathsf{pk}_f), r, c$ as an input to $\mathtt{Rec}$. Since by construction $c = \mathtt{Enc}^{\Sigma}_{\mathsf{pk}_e}(\mathtt{F}_{\mathsf{pk}_f}(x);r) = \mathtt{Enc}_{\mathsf{pk}}(x;r)$, then $\mathtt{Rec}$ outputs x. So $\mathcal{B}$ can find the correct preimage with probability 1, hence breaking the security of the trapdoor. This contradicts the recoverability of Γ.

Finally, in Fig. 8 we show an efficient circuit for the realization of $U^{(2)}_{\mathtt{Enc}}$. This uses subcircuits for computing the type-1 operator for the trapdoor permutation and its inverse (given the permutation's public key and trapdoor), and type-1 encryption and recovery for the underlying PKE scheme. □

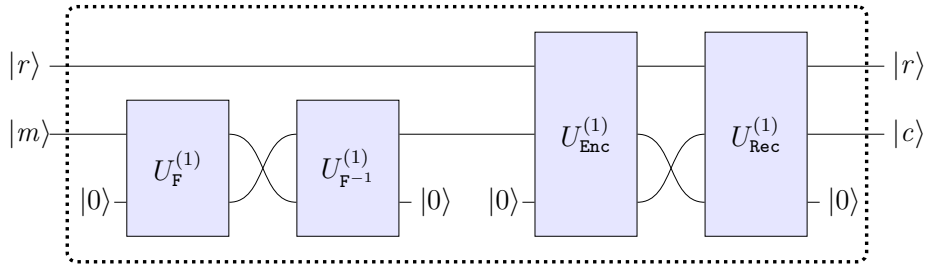

Fig. 8: Efficient realization of the canonical type-2 encryption operator for the construction shown in Fig. 7.

7 Future Directions

In this work we have filled the existing gap between the symmetric key and the public key case when defining security in the QS2 setting. We showed how the existence of this gap was not due to a mere lack of interest, but because of non-trivial definitional issues that we solved. We believe that our results provide useful guidelines in the security analysis of quantum-resistant PKE, but many research directions remain open to exploration.

In Section 3.5 we sketch a general strategy for extending our results to the chosen ciphertext case. Although we believe that such a strategy works, we leave it as an open problem to formalize it correctly. We also leave it as an open problem to improve our game-based definitions to stronger provable security paradigms such as simulation-based or concrete security.

We notice how our notions of qIND-qCPA for PKE can be also used to study the security of cryptographic primitives that 'extend' PKE with extra functional-

ities. Such primitives include, for example, fully homomorphic encryption [15,23], identity-based encryption [42], and functional encryption [12].

We also leave as an open conjecture the security of the canonical hybrid encryption scheme in the case that both the underlying PKE and SKE schemes are qIND-qCPA. One idea for this would be to follow a similar approach as in [5].

We did not found any natural example of a scheme that is isometric, yet not computationally so. Notice the following: it would be trivial to modify the construction from Fig. 7 in such a way that the circuit provided in Fig. 8 becomes non-efficient (for example by using a hard to invert permutation instead of a trapdoor permutation). However this would only show that *that* particular construction of the type-2 operator is inefficient, while we would need to show that *any* construction is.

Also, notice the following: given that qIND-qCPA is a stronger notion than IND-qCPA, having a PKE scheme where it is not even possible to define a type-2 encryption operator can actually be *desirable*. For such a scheme in fact, one should not worry about proving the (stricter) qIND-qCPA security notion, because the related attack scenario is simply not enforceable, and hence the scheme cannot be broken in a qIND-qCPA sense. So it would be interesting to find schemes which are IND-qCPA secure but non-isometric. We conjecture that a generic transformation to obtain such schemes is possible using *indistinguishability obfuscation*, but leave the problem open to further study.

We have also left unstudied the possibility of extending QS2 security notions to the use of type-π operators, and to models where the adversary can query oracles on superpositions of public keys. Finally, we leave as an open problem finding a generic transformation to 'patch' generic IND-qCPA secure schemes into qIND-qCPA secure ones.

Acknowledgements

The authors thank Cecilia Boschini and Marc Fischlin for helpful discussions regarding the correctness of public key encryption schemes and Andreas Hüsling for general discussions on the content of this work. TG acknowledges support by the EU H2020 Project FENTEC (Grant Agreement #780108). JK and PS acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

References

1. C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor. HQC. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
2. C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, G. Zémor, A. Couvreur, and A. Hauteville. RQC. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.

3. G. Alagic, A. Broadbent, B. Fefferman, T. Gagliardoni, C. Schaffner, and M. S. Jules. Computational security of quantum encryption. In A. C. A. Nascimento and P. Barreto, editors, *ICITS 16*, volume 10015 of *LNCS*, pages 47–71. Springer, Heidelberg, Aug. 2016.
4. G. Alagic, T. Gagliardoni, and C. Majenz. Can you sign a quantum state. *IACR Cryptology ePrint Archive*, 2018:1164, 2018.
5. G. Alagic, T. Gagliardoni, and C. Majenz. Unforgeable quantum encryption. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 489–519. Springer, Heidelberg, Apr. / May 2018.
6. G. Alagic and A. Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 65–93. Springer, Heidelberg, Apr. / May 2017.
7. M. R. Albrecht, E. Orsini, K. G. Paterson, G. Peer, and N. P. Smart. Tightly secure ring-LWE based key encapsulation with short ciphertexts. In S. N. Foley, D. Gollmann, and E. Snekkenes, editors, *ESORICS 2017, Part I*, volume 10492 of *LNCS*, pages 29–46. Springer, Heidelberg, Sept. 2017.
8. M. V. Anand, E. E. Targhi, G. N. Tabia, and D. Unruh. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In T. Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 44–63. Springer, Heidelberg, 2016.
9. N. Aragon, O. Blazy, J.-C. Deneuville, P. Gaborit, A. Hauteville, O. Ruatta, J.-P. Tillich, G. Zémor, C. Aguilar Melchor, S. Bettaieb, L. Bidoux, M. Bardet, and A. Otmani. ROLLO. Technical report, National Institute of Standards and Technology, 2019. available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions`.
10. D. J. Bernstein, J. Buchmann, and E. Dahmen. Post-quantum cryptography, 2009.
11. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In D. H. Lee and X. Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, Dec. 2011.
12. D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, Mar. 2011.
13. D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, Aug. 2013.
14. J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367, 2018.
15. A. Broadbent and S. Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 609–629. Springer, Heidelberg, Aug. 2015.
16. C. Chevalier, E. Ebrahimi, and Q.-H. Vu. On the security notions for encryption in a quantum world. *IACR Cryptology ePrint Archive*, 2020:237, 2020.
17. S. Chow, P. A. Eisen, H. Johnson, and P. C. van Oorschot. White-box cryptography and an AES implementation. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 250–270. Springer, Heidelberg, Aug. 2003.
18. A. Couvreur, M. Bardet, E. Barelli, O. Blazy, R. Canto-Torres, P. Gaborit, A. Otmani, N. Sendrier, and J.-P. Tillich. BIG QUAKE. Technical report, National

Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
19. I. Damgård, J. Funder, J. B. Nielsen, and L. Salvail. Superposition attacks on cryptographic protocols. In C. Padró, editor, *ICITS 13*, volume 8317 of *LNCS*, pages 142–161. Springer, Heidelberg, 2014.
20. E. Eaton and F. Song. Making existential-unforgeable signatures strongly unforgeable in the quantum random-oracle model. In *10th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2015, May 20-22, 2015, Brussels, Belgium*, pages 147–162, 2015.
21. T. Gagliardoni. *Quantum Security of Cryptographic Primitives*. PhD thesis, Darmstadt University of Technology, Germany, 2017.
22. T. Gagliardoni, A. Hülsing, and C. Schaffner. Semantic security and indistinguishability in the quantum world. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 60–89. Springer, Heidelberg, Aug. 2016.
23. C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.
24. L. K. Grover. A fast quantum mechanical algorithm for database search. In *28th ACM STOC*, pages 212–219. ACM Press, May 1996.
25. G. Ito, A. Hosoyamada, R. Matsumoto, Y. Sasaki, and T. Iwata. Quantum chosen-ciphertext attacks against Feistel ciphers. In M. Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 391–411. Springer, Heidelberg, Mar. 2019.
26. M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, Heidelberg, Aug. 2016.
27. M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia. Quantum differential and linear cryptanalysis. *IACR Trans. Symm. Cryptol.*, 2016(1):71–94, 2016. http://tosc.iacr.org/index.php/ToSC/article/view/536.
28. E. Kashefi, A. Kent, V. Vedral, and K. Banaszek. Comparison of quantum oracles. *Physical Review A*, 65(5):050304, 2002.
29. H. Kuwakado and M. Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *IEEE International Symposium on Information Theory, ISIT 2010, June 13-18, 2010, Austin, Texas, USA, Proceedings*, pages 2682–2685, 2010.
30. H. Kuwakado and M. Morii. Security on the quantum-type even-mansour cipher. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012*, pages 312–316, 2012.
31. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In A. Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, Feb. 2011.
32. X. Lu, Y. Liu, D. Jia, H. Xue, J. He, Z. Zhang, Z. Liu, H. Yang, B. Li, and K. Wang. LAC. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
33. S. Mossayebi and R. Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *CoRR*, abs/1609.03780, 2016.
34. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

35. N. I. of Standards and Technology. Post-quantum cryptography standardization process, 2017.
36. T. Poppelmann, E. Alkim, R. Avanzi, J. Bos, L. Ducas, A. de la Piedra, P. Schwabe, D. Stebila, M. R. Albrecht, E. Orsini, V. Osheter, K. G. Paterson, G. Peer, and N. P. Smart. NewHope. Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions.
37. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
38. M. Rötteler and R. Steinwandt. A note on quantum related-key attacks. *Inf. Process. Lett.*, 115(1):40–44, 2015.
39. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, Nov. 1994.
40. J. Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.
41. M. Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, Oct. 2012.
42. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, Aug. 2012.
43. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, Aug. 2019.

A Additional Preliminaries

We give some additional background on symmetric key encryption schemes for the results in Section 5.

A.1 IND-qCPA Security of Public Key Encryption Schemes

The security game for IND-qCPA security [13] of public key encryption schemes is defined as follows.

Experiment 26 *The* IND-qCPA experiment $\mathsf{IND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda)$ *for a* PKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *is defined as follows:*

1: $\mathcal{C}$ *runs* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KGen}$
2: $\mathcal{A}(\mathsf{pk}) \rightarrow (m_0, m_1, \sigma_{state})$
3: $\mathcal{C}$ *receives* m_0, m_1 *and does the following:*
- *flips* $b \xleftarrow{\$} \{0,1\}$
- *samples* $r \xleftarrow{\$} \mathcal{R}$
- *computes* $\mathtt{Enc}_{\mathsf{pk}}(m_b; r) \rightarrow c$
- *sends* c *to* $\mathcal{A}$

4: $\mathcal{A}(\sigma_{state}, c) \rightarrow b' \in \{0,1\}$
5: ***if*** $b = b'$ ***then return*** *win;* ***else return*** *rej.*

Like for secret key encryption schemes, security is defined as negligible advantage over guessing in winning the security game.

Definition 27 (IND-qCPA**,** PKE**)** *A* SKE *scheme* Σ *has* ciphertext indistinguishability under quantum chosen plaintext attack*, or it is* IND-qCPA*-secure, iff for any QPT adversary* $\mathcal{A}$ *it holds:*

$$\left| \Pr\left[\mathsf{IND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda) \rightarrow \mathit{win}\right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda)\,.$$

A.2 Symmetric Key Encryption

Below we define symmetric key encryption (SKE) schemes.

Definition 28 *A symmetric key encryption* (SKE) *scheme* Σ *is a tuple of three efficient algorithms* $(\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *such that:*

- $\mathtt{KGen}\colon \mathbb{N} \rightarrow \mathcal{K}$ *is the (randomized) encryption algorithm which takes a security parameter* λ *as input, and returns a key* k*.*
- $\mathtt{Enc}\colon \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ *is the (randomized) encryption algorithm which takes a key* k *and a message* m *as input, and returns a ciphertext* c*.*
- $\mathtt{Dec}\colon \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$ *is the decryption algorithm which takes as input a key* k *and a ciphertext* c*, and returns a message* m*.*

By $\mathcal{K}$*,* $\mathcal{M}$*, and* $\mathcal{C}$*, we denote the key space, message space, and ciphertext space, respectively.*

Next, we define the two security games for IND-qCPA and qIND-qCPA security, following [13] and [22]. Note that the type-2 oracle access in for the qIND-qCPA security game is not explicitly mentioned in [22].

Experiment 29 *The* IND-qCPA experiment $\mathsf{IND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda)$ *for a* SKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *is defined as follows:*

1: $\mathcal{C}$ *runs* $\mathsf{k} \leftarrow \mathtt{KGen}(\lambda)$
2: $\mathcal{A}^{U_{\mathtt{Enc}}^{(1)}}() \rightarrow (m_0, m_1, \sigma_{state})$
3: $\mathcal{C}$ *receives* m_0, m_1 *and does the following:*
- *flips* $b \xleftarrow{\$} \{0,1\}$
- *computes* $\mathtt{Enc}_{\mathsf{k}}(m_b) \rightarrow c$
- *sends* c *to* $\mathcal{A}$

4: $\mathcal{A}^{U_{\mathtt{Enc}}^{(1)}}(\sigma_{state}, c) \rightarrow b' \in \{0,1\}$
5: ***if*** $b = b'$ ***then return*** *win;* ***else return*** *rej.*

Experiment 30 *The* qIND-qCPA experiment $\mathsf{qIND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda)$ *for a* SKE *scheme* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *is defined as follows:*

1: $\mathcal{C}$ *runs* $\mathsf{k} \leftarrow \mathtt{KGen}(\lambda)$
2: $\mathcal{A}^{U_{\mathtt{Enc}}^{(1)}, U_{\mathtt{Enc}}^{(2)}}() \rightarrow (\varphi_0, \varphi_1, \sigma_{state})$
3: $\mathcal{C}$ *receives* φ_0, φ_1 *and does the following:*
- *flips* $b \xleftarrow{\$} \{0,1\}$
- *traces out* φ_{1-b}
- *applies* $U_{\mathtt{Enc}}^{(2)}\varphi_b \rightarrow \psi$
- *sends* ψ *to* $\mathcal{A}$

4: $\mathcal{A}^{U_{\mathtt{Enc}}^{(1)}, U_{\mathtt{Enc}}^{(2)}}(\sigma_{state}, \psi) \rightarrow b' \in \{0,1\}$
5: ***if*** $b = b'$ ***then return*** *win;* ***else return*** *rej.*

Just as for our new security notion, security is defined as negligible advantage over guessing in winning the games.

Definition 31 (IND-qCPA**,** SKE**)** *A* SKE *scheme* Σ *has* ciphertext indistinguishability under quantum chosen plaintext attack*, or it is* IND-qCPA*-secure, iff for any QPT adversary* $\mathcal{A}$ *it holds:*

$$\left| \Pr\left[\mathsf{IND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda) \rightarrow \mathit{win}\right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda)\,.$$

Definition 32 (qIND-qCPA**,** SKE**)** *A* SKE *scheme* Σ *has* quantum ciphertext indistinguishability under quantum chosen plaintext attack*, or it is* qIND-qCPA*-secure, iff for any QPT adversary* $\mathcal{A}$ *it holds:*

$$\left| \Pr\left[\mathsf{qIND\text{-}qCPA}(\Sigma, \mathcal{A}, \lambda) \rightarrow \mathit{win}\right] - \frac{1}{2} \right| \leq \mathrm{negl}(\lambda)\,.$$

B The Role of Randomness Superposition

In this section we discuss the possibility of having superposition of randomness in the challenge query. So far, we have only considered the case of classical randomness, as this is chosen by the (honest) challenger. But one could consider scenarios where the adversary can somehow trick the challenger into using a superposition of randomness in the challenge query. Here we discuss two possible ways to deal with this issue, one of which turns out to be unachievable while the other yields a notion equivalent to the one we propose in Section 3.

Assume that the challenger chooses a superposition of randomness to encrypt one of the messages chosen by the adversary. Following our security experiment, the challenger would keep the randomness register and merely send the ciphertext register to the adversary. The crucial observation is that the registers containing the randomness and the ciphertext are now entangled. As observed in [22], withholding the randomness register is equivalent to measuring it from the point of view of the adversary. This means that this approach would in fact be equivalent to our security notion using a classical randomness.

Alternatively, to prevent the aforementioned issue of entanglement between the challenger and the adversary, we might let the challenger send the randomness register to the adversary. However, the resulting security notion is unachievable as it would allow the adversary to always distinguish encryptions. We illustrate this with the following attack. First, the adversary chooses two distinct classical messages m_0, m_1, and executes the qIND challenge query with these two. Upon receiving the ciphertext register and the randomness register, the adversary evaluates (locally) the type-1 encryption operator initialising the input register with $|m_0\rangle$, the randomness register with the randomness state received from the challenger, and the ancilla register with the received ciphertext. Finally, the adversary measures the ciphertext register output of the type-1 encryption operator: if he measures 0, then he outputs $b = 0$, otherwise outputs $b = 1$. The circuit is depicted in Fig. 9. The attack works because, if $b = 0$, then the adversary will compute the same ciphertext as the challenger, hence the output register of the type-1 encryption will be $|0\rangle$; on the other hand, if $b = 1$, a random value will be observed instead. Clearly, this results in output states that the adversary can distinguish with overwhelming probability.

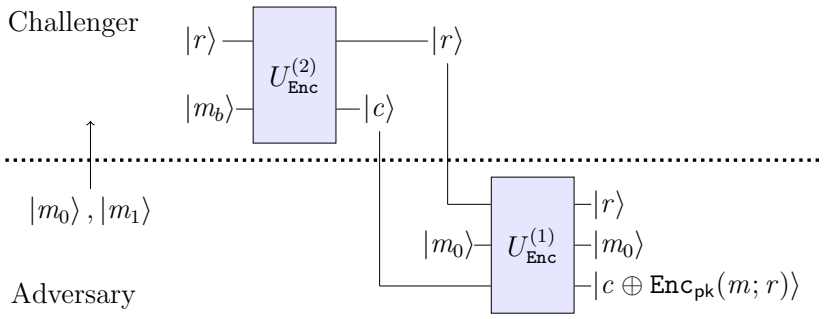

Fig. 9: Generic attack against superposition of randomness.