On Quantum Ciphertext Indistinguishability, Recoverability, and OAEP

Juliane Krämer and Patrick Struck

Universität Regensburg, Germany
{juliane.kraemer,patrick.struck}@ur.de

Abstract. The qINDqCPA security notion for public-key encryption schemes by Gagliardoni et al. (PQCrypto'21) models security against adversaries which are able to obtain ciphertexts in superposition. Defining this security notion requires a special type of quantum operator. Known constructions differ in which keys are necessary to construct this operator, depending on properties of the encryption scheme.
We argue—for the typical setting of securing communication between Alice and Bob—that in order to apply the notion, the quantum operator should be realizable for challengers knowing only the public key. This is already known to be the case for a wide range of public-key encryption schemes, in particular, those exhibiting the so-called recoverability property which allows to recover the message from a ciphertext using the randomness instead of the secret key.
The open question is whether there are real-world public-key encryption schemes for which the notion is not applicable, considering the aforementioned observation on the keys known by the challenger. We answer this question in the affirmative by showing that applying the qINDqCPA security notion to the OAEP construction requires the challenger to know the secret key. We conclude that the qINDqCPA security notion might need to be refined to eventually yield a universally applicable PKE notion of quantum security with a quantum indistinguishability phase.

1 Introduction

In light of the threat posed by quantum algorithms such as Shor's [33], cryptographic primitives that are assumed to withstand attacks using quantum computing are mandatory to ensure security also in the future. Over the last decade, a lot of research focused on exactly this type of cryptographic primitives—commonly known as post-quantum cryptography.

Security against attackers with quantum computing power can be divided into two categories. The first, and arguably the one that will be realistic in the upcoming years, is post-quantum security. Here, the adversary has local quantum computing power, which allows to evaluate public primitives like hash functions in superposition, while keyed cryptographic primitives can be accessed only classically. The second, and more conservative setting, is quantum security, which grants the adversary superposition access also to keyed cryptographic primitives. The latter defines the scope of this work.

The research area of quantum security was initiated by Boneh and Zhandry, who gave the first quantum security notions for encryption schemes and signature schemes [8]. At the moment, quantum security is still at a point where definitional challenges have to be solved, e.g., it has to be understood how classically well-established concepts like "distinguishing two ciphertexts" and "forge a signature for a new message" can be translated to the quantum setting. For signatures, initial problems of the Boneh–Zhandry notion where identified in [20] and a potential solution was given in [1]. For public-key encryption schemes, there are two different approaches to avoid limitations when switching to a quantum challenge phase: a left-or-right approach given in [19] and a real-or-random approach given in [12]. While the latter can be defined for any public-key encryption scheme, this is not the case for the former. This is discussed in detail in the full version of [19] and stems from the fact that the security notion developed in [19] requires a special type of quantum operator. Simply speaking, the notion requires an in-place quantum operator that transforms a state $|x\rangle$ into $|\mathcal{F}(x)\rangle$ instead of the canonical xor operator that transforms $|x, y\rangle$ into $|x, y \oplus \mathcal{F}(x)\rangle$. The authors of [19] give two constructions for the required in-place operator, based on the properties of the encryption scheme. These two constructions, which we describe later, differ in the keys that are necessary: one construction requires merely the public key, the other construction requires both the secret key and the public key. While the latter seems artificial, we stress that [19] focuses on whether the operator required by their security notion can be constructed *at all*. In fact, the authors show, surprisingly, that most real-world public-key encryption schemes allow for the construction that requires just the public key.

1.1 Our Contribution

In this work, we study the quantum security notion from [19] regarding its applicability.

We first revisit the typical notion of ciphertext indistinguishability in the context of securing communication between two parties. We argue that, for this setting, challengers should only have access to the public key. Regarding the construction of the in-place quantum operator from the qINDqCPA security notion, this is known to be the case for most real-world public-key encryption schemes, namely, those exhibiting the recoverability property [19].

We then focus on the question whether there are public-key encryption schemes which do not have the recoverability property. The only known schemes are obtained by a generic transformation [19] which transforms a recoverable public-key encryption scheme into a non-recoverable one. We refine the classification by showing that there are not just recoverable and non-recoverable public key encryption schemes, but also what we call *equivalent recoverable*.

Finally, we investigate the OAEP construction. We show that this construction is non-recoverable, thereby giving the first real-world PKE scheme with this property. We then show that—for the OAEP construction—the in-place operator needed for the quantum security notion from [19] cannot be constructed using just the public key. Thereby we show that the quantum security notion

qINDqCPA cannot be applied to all PKE schemes when imposing the restriction that the challenger only knows the public key.

1.2 Related Work

Quantum security notions were first considered by Boneh and Zhandry [8]. Since then, many works developed new quantum security notions or analyzed primitives with respect to existing notions, ranging from signature schemes and message authentication codes [1,7,20], to symmetric encryption [11,12,14,18,28,29], and to public-key encryption schemes [12,19].

A series of works [2,3,9,10,21–23,25–27,31] show that superposition attacks (which are modeled by quantum security notions) can have devastating effects on cryptographic primitives by providing attacks against primitives like Even-Mansour, the FX construction, Feistel networks, block ciphers, and HMAC.

The optimal asymmetric encryption padding (OAEP) was developed by Bellare and Rogaway [6]. The initial proof had a gap, as detected by Shoup [34], who provided a variant of OAEP with an alternative proof. Fujisaki et al. [17] provide a proof for OAEP avoiding the initial gap by strengthening the requirements of the underlying function. Security of OAEP against quantum attackers was first considered in [35] which required a slight modification to prove security. Recently, post-quantum security of the plain OAEP construction was shown [15].

1.3 Outline

In Section 2, we provide background on quantum security notions, quantum operators, and cryptography as necessary for this work. In Section 3, we refine the qINDqCPA security notion. In Section 4 we review the notion of recoverable PKE schemes. In Section 5, we analyze the OAEP construction with respect to the qINDqCPA security notion.

2 Preliminaries

2.1 Notation

For a set $\mathcal{X}$, we write $x \leftarrow\!\!\$\ \mathcal{X}$ to denote the process of picking an element from $\mathcal{X}$ at random and assigning it to x. 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 of a cryptographic scheme, respectively. For a deterministic algorithm $\mathcal{F}$, we write $y \leftarrow \mathcal{F}(x)$ to denote that y is the output of $\mathcal{F}$ on input x. For a probabilistic algorithm, $y \leftarrow \mathcal{F}(x;r)$ denotes that the output of $\mathcal{F}$ on input x with randomness r equals y. We write $y \leftarrow\!\!\$\ \mathcal{F}(x)$ to denote the process that the randomness r is chosen uniformly at random and y is the output of $\mathcal{F}$ on input x with randomness r.

2.2 Public-Key Cryptography

Public-key encryption schemes are defined below.

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* $(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}()$*. We will generally drop the security parameter.*
- $\mathtt{Enc}\colon \mathcal{P} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ *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* $c \leftarrow\!\!\$\ \mathtt{Enc}(pk, m)$ *or* $c \leftarrow \mathtt{Enc}(pk, m; r)$*.*
- $\mathtt{Dec}\colon \mathcal{S} \times \mathcal{C} \to \mathcal{M}$ *is the deterministic decryption algorithm which takes as input a secret key* sk *and a ciphertext* c*, and returns a message* m*. It will be usually denoted by* $m \leftarrow \mathtt{Dec}(sk, c)$*.*

Next we define trapdoor permutations. The definition is tailored to permutations over the Cartesian product over two sets $\mathcal{X}_1$ and $\mathcal{X}_2$. This is without loss of generality but allows for a simple definition of the security notions required later.

Definition 2. *A* trapdoor permutation *is a tuple* $(\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ *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*.*
- $\mathcal{F}\colon \mathcal{P} \times \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{X}_1 \times \mathcal{X}_2$ *is the permutation algorithm which takes a public key* pk*, permuting over the set* $\mathcal{X}_1 \times \mathcal{X}_2$*.*
- $\mathcal{F}^{-1}\colon \mathcal{S} \times \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{X}_1 \times \mathcal{X}_2$ *is the inverse permutation which takes as input a secret key* sk *and permutes over the set* $\mathcal{X}_1 \times \mathcal{X}_2$*.*

Below we define two security notions for a trapdoor permutation $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$. One asks to find the preimage of a given output whereas the other only asks to find a partial preimage, i.e., a preimage with respect to $\mathcal{X}_1$.

Definition 3. *Let* $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ *be a trapdoor permutation. Let further the games* OW *and* pdOW *be defined as in Fig. 1. For any adversary* $\mathcal{A}$*, we define its advantages as*

$$\mathbf{Adv}^{\mathsf{OW}}(\mathcal{A}) \coloneqq \Pr[\mathsf{OW}^{\mathcal{A}} \to \text{true}]$$
$$\mathbf{Adv}^{\mathsf{pdOW}}(\mathcal{A}) \coloneqq \Pr[\mathsf{pdOW}^{\mathcal{A}} \to \text{true}]\,.$$

Game OW	Game pdOW
$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}()$	$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}()$
$(x_1, x_2) \leftarrow\!\!\$\ \mathcal{X}_1 \times \mathcal{X}_2$	$(x_1, x_2) \leftarrow\!\!\$\ \mathcal{X}_1 \times \mathcal{X}_2$
$(z_1, z_2) \leftarrow \mathcal{F}(pk, (x_1, x_2))$	$(z_1, z_2) \leftarrow \mathcal{F}(pk, (x_1, x_2))$
$(y_1, y_2) \leftarrow \mathcal{A}(pk, (z_1, z_2))$	$y_1 \leftarrow \mathcal{A}(pk, (z_1, z_2))$
return $(y_1, y_2) = (x_1, x_2)$	**return** $y_1 = x_1$

Fig. 1: Game OW (One-Wayness) and game pdOW (Partial-Domain One-Wayness) to define security of a trapdoor permutation $\mathcal{F}$.

2.3 Quantum Computing

We assume familiarity with quantum computing and refer to [30] for details. Implementing a function $\mathcal{F}\colon \mathcal{X} \to \mathcal{Y}$ on a quantum computer is typically done via the canonical construction. This is what we call an xor operator, which is defined as

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

This xor operator can be implemented efficiently whenever $\mathcal{F}$ is efficient [30]. If a function is invertible, there is another operator—besides the xor operator—with which the function can be realized. This operator is what we call an in-place operator, which is defined as

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

Fig. 2 illustrates the two operators for a function $\mathcal{F}$. Kashefi et al. [24] first introduced in-place operators giving them the name minimal oracles. They show that the two variants are not equivalent by showing that in-place operators are stronger than xor operators. The core observation is that inverting an in-place operator gate-by-gate gives an in-place operator for the inverse function. The same does not apply to the xor operator, as xor operators are self-inverse.

Fig. 2: **Left:** xor operator for $\mathcal{F}$. **Right:** in-place operator for $\mathcal{F}$.

In the following we recall two variants how xor and in-place operators (for invertible functions) can be constructed from one another. Fig. 3 shows how

an xor operator for $\mathcal{F}$ can be constructed from an in-place operator for $\mathcal{F}$. Likewise, Fig. 4 shows how an in-place operator for $\mathcal{F}$ can be constructed from xor operators for *both* $\mathcal{F}$ and $\mathcal{F}^{-1}$. Note here, that an xor operator for $\mathcal{F}$ does—in general—not allow to construct an xor operator for $\mathcal{F}^{-1}$. As an example, consider $\mathcal{F}$ to be some one-way function. The latter construction (cf. Fig. 4) is important for the qINDqCPA security notion.

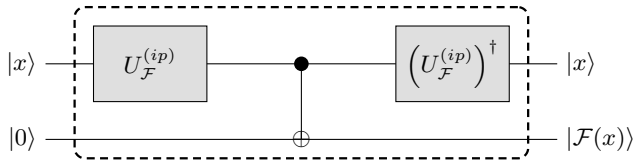

Fig. 3: Construction of an xor operator for a function $\mathcal{F}$ from an in-place operator for $\mathcal{F}$ and its inverse.

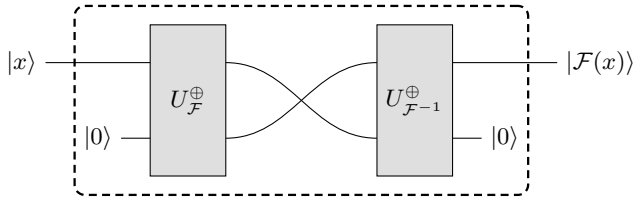

Fig. 4: Construction of an in-place operator for a function $\mathcal{F}$ from xor operators for $\mathcal{F}$ and $\mathcal{F}^{-1}$.

3 (Quantum) Ciphertext Indistinguishability

3.1 The qINDqCPA Security Notion

Ciphertext indistinguishability is a security notion for encryption schemes—both symmetric encryption and public-key encryption. It asks an adversary to distinguish between the encryption of two adversarial chosen messages. An encryption scheme that achieves ciphertext indistinguishability comes with the guarantee that an adversary cannot learn any information about the message underlying a ciphertext.

When considering quantum adversaries, one can distinguish between adversaries restricted to local quantum computing power—the setting that is widely known as post-quantum security—and adversaries that have full quantum access to all oracles—known as quantum security. The latter is the setting considered by Gagliardoni et al. [19] who develop the qINDqCPA security notion for

public-key encryption schemes. This notion models security where the challenge ciphertexts by the adversary can be in superposition, hence it provides stronger security guarantees than post-quantum security.

The security game qINDqCPA is displayed in Fig. 5. The adversary receives a public key pk and gets access to an in-place operator for encryption.[1] The adversary then outputs two messages $|\varphi_0\rangle$ and $|\varphi_1\rangle$, possibly in superposition, one of which will be encrypted and then sent back to the adversary. The adversary continues to have access to the in-place operator for encryption and finally has to output its guess b', indicating which of the two messages was encrypted.

Game qINDqCPA	**oracle** $\mathsf{Enc}(\lvert\varphi\rangle)$, where $\lvert\varphi\rangle = \sum_m \alpha_m \lvert m\rangle$
$b \leftarrow\!\!\$\ \{0,1\}$	$r \leftarrow\!\!\$\ \mathcal{R}$
$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}()$	$\lvert r\rangle \lvert c\rangle \leftarrow U^{ip}_{\mathtt{Enc}} (\lvert r\rangle \lvert\varphi\rangle \lvert 0\cdots 0\rangle)$
$\lvert\varphi_0\rangle, \lvert\varphi_1\rangle \leftarrow \mathcal{A}_1^{\mathsf{Enc}}(pk)$	trace out $\lvert r\rangle$
trace out $\lvert\varphi_{1-b}\rangle$	**return** $\lvert c\rangle$
$r \leftarrow\!\!\$\ \mathcal{R}$	
$\lvert r\rangle \lvert c\rangle \leftarrow U^{ip}_{\mathtt{Enc}} (\lvert r\rangle \lvert\varphi_b\rangle \lvert 0\cdots 0\rangle)$	
trace out $\lvert r\rangle$	
$b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}}(\lvert c\rangle)$	
return $(b' = b)$	

Fig. 5: Security game qINDqCPA.

At the core of the qINDqCPA security notion[2] is the in-place operator $U^{ip}_{\mathtt{Enc}}$ of the form

$$U^{ip}_{\mathtt{Enc}} \colon\ |r, m, 0\cdots 0\rangle \mapsto |r\rangle\, |\mathtt{Enc}(pk, m; r)\rangle\ , \tag{1}$$

where the extra qubits initialized with 0 are for the ciphertext expansion as, in general, we have $|\mathcal{C}| > |\mathcal{M}|$. In [19], the authors observe that this is the most general in-place operator for encryption. Keeping the randomness in an extra register deals with randomness collisions, i.e., different randomnesses that encrypt a message to the same ciphertext, which would thwart the mandatory reversibility. Note that the extra randomness register also prevents to construct a simple decryption operator by inverting the encryption operator gate-by-gate. It would only allow to decrypt if the randomness used to produce a ciphertext is known.

[1] Note that this oracle is important in case the adversary cannot locally implement this oracle. Without it, the standard simplification to a single challenge via a hybrid argument does not work.

[2] Note that there are other security notions [8, 12, 28] which only require an xor operator. However, the relation between these different approaches is not completely understood and requires more research.

The core part of [19] lies in the construction of the in-place operator for encryption. The authors show that for schemes which do not suffer from decryption failures, such in-place operators can be efficiently constructed. This construction consists of an xor operator for encryption and an xor operator for decryption. This part exploits that decryption is the inverse of encryption and essentially follows the idea of the construction in Fig. 4. They further show that in-place operators can also be constructed for schemes which are recoverable; a property they define and which was concurrently[3] and independently defined by Bellare et al. in the context of domain separation for random oracles [4]. This construction also consists of an xor operator for encryption, but instead of an xor operator for decryption, it uses an xor operator for the so-called recover algorithm (see Section 4 for details). Interestingly, realizations of the in-place operator for the two types of encryption schemes—those without decryption failures and those exhibiting the recoverable property—is quite different: while it can be realized solely from the public key for the latter, the former necessitates the secret key. For schemes which fall into neither category, they show that there are schemes for which the in-place operator can be realized, as well as schemes for which it simply cannot be realized. We illustrate the two constructions from [19] in Fig. 6.

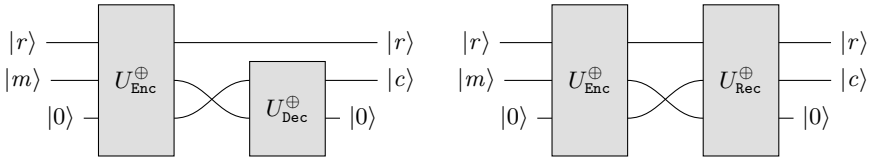

Fig. 6: Two constructions for an in-place encryption operator as given in [19].

3.2 Interpretation of Ciphertext Indistinguishability

Having discussed the qINDqCPA security notion and the relevance of the keys for constructing the required in-place operator, we now take a step back and focus on ciphertext indistinguishability in general.

Fig. 7 displays the security game for ciphertext indistinguishability under chosen ciphertext attacks. Here, the adversary receives a public key and can query two oracles: First, an encryption oracle that takes two messages as input and encrypts one of the two, based on a secret bit chosen uniformly at random at the beginning of the game. Second, a decryption oracle which takes a ciphertext as input and returns the decryption of it.[4]

[3] The full version of both works appeared within a week.

[4] For simplicity we ignore how cheating adversaries, which forward a response from the encryption to the decryption oracle, are prevented. For a detailed discussion on this, we refer to [5].

In the security notion, there is only one challenger providing the adversary its input (the public key) and its oracles (encryption and decryption oracle). In particular, the challenger generates, and thus knows, both keys. Considering what the two oracles correspond to in the real-world, they are quite different. The encryption oracle represents the sender, sending a ciphertext, while the decryption oracle represents the recipient, receiving a ciphertext. In the typical setting of Alice sending an encrypted message to Bob, Alice is represented by the encryption oracle while Bob is represented by the decryption oracle. The main difference is that Alice only knows Bob's public key, whereas Bob knows both his public and secret key. In this sense, the encryption oracle should be realizable using only the public key. The decryption oracle can be realized from both the secret and the public key.[5]

We note that there are scenarios where the above observation does not apply, for instance when considering public-key encryption schemes used for commitment schemes. Here, Alice, holding both keys, would encrypt the message she wants to commit to and send the ciphertext to Bob. Later, when Alice wants to open the commitment, she reveals her secret key, more precisely the random coins used to generate it, to Bob. In this case, the encrypting party, Alice, knows both keys while the adversary still only knows the public key.

We conclude this section with the observation that in the qINDqCPA security notion—when modeling standard encrypted communication with it—, the challenger should be able to construct the in-place encryption operator using the public key only. One can, for instance, let the challenger discard the secret key after generating the key pair or let the challenger receive only the public key from another trusted party. This raises the following question:

Are there public-key encryption schemes for which qINDqCPA security cannot be defined for challengers knowing only the public key?

Using the results by Gagliardoni et al. [19], we know that the notion, when imposing the restriction that the challenger only knows the public key, can be defined for any recoverable public-key encryption scheme. For non-recoverable schemes, however, the answer is unclear. Gagliardoni et al. [19] show that it *can* be defined for challengers knowing both keys, but they do not discuss if it *can only* be defined if the challenger knows both keys. Hence, in Section 4 we will focus on non-recoverable public-key encryption schemes and recoverability in general. In Section 5 we return to the main question above by studying the OAEP construction.

4 Observations on Recoverability

The notion of recoverable PKE schemes has been introduced in [19]. In Section 3 we concluded that especially non-recoverable schemes have to be fur-

[5] When using the FO transformation [16], the public key is mandatory for the re-encrypting part. This dependence is often implicit, e.g., schemes such as Kyber [32] and Saber [13] specify the secret key to already include the public key.

Game INDCCA	**oracle** LR-Enc(m_0, m_1)	**oracle** Dec(c)
$b \leftarrow\!\!\$\ \{0,1\}$	$c \leftarrow \mathtt{Enc}(pk, m_b)$	$m \leftarrow \mathtt{Dec}(sk, c)$
$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}()$	**return** c	**return** m
$b' \leftarrow \mathcal{A}^{\mathsf{LR\text{-}Enc},\mathsf{Dec}}(pk)$		
return $(b' = b)$		

Fig. 7: Security game INDCCA. For simplicity we drop the check that the decryption oracle checks whether a queried ciphertext was forwarded from the oracle LR-Enc.

ther studied to understand whether qINDqCPA security can be defined using solely the public key. Unfortunately, so far we are not aware of any real-world non-recoverable scheme. Instead of a concrete non-recoverable scheme, in [19] a generic transformation was introduced that transforms a recoverable scheme into a non-recoverable scheme. In this section, however, by introducing what we call equivalent recoverable schemes, we show that this transformation can also be defined inversely, which questions the meaningfulness of the transformation with respect to the existence of a non-recoverable scheme. By introducing equivalent recoverable schemes, we hence refine the classification introduced in [19], such that two kinds of schemes exist which are not recoverable: equivalent recoverable schemes and non-recoverable schemes. We conclude this section with the open question whether (real-world) non-recoverable PKE schemes exist at all.

We first repeat the notion of recoverability in Section 4.1 and then define equivalent recoverable schemes in Section 4.2.

4.1 Recoverability

Recoverability is a property of public-key encryption schemes that was defined by Gagliardoni et al. [19]. Simply speaking, a public-key encryption scheme is recoverable if one can recover the message from a ciphertext when knowing the randomness that was used to create said ciphertext—even without the secret key. Below we formally define recoverable public-key encryption schemes.

Definition 4 (Recoverable PKE Scheme [19, Definition 6]). *Let* $\Sigma = (\mathtt{KGen}, \mathtt{Enc}, \mathtt{Dec})$ *be a PKE scheme. We call* Σ *a* recoverable *PKE scheme if there exists an efficient algorithm* $\mathtt{Rec}\colon \mathcal{P} \times \mathcal{R} \times \mathcal{C} \to \mathcal{M}$ *such that, for any* $pk \in \mathcal{P}, r \in \mathcal{R}, m \in \mathcal{M}$*, it holds that*

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

An important property of recoverable PKE schemes is that the recover algorithm Rec allows to perfectly recover the message from a ciphertext; even if the scheme itself suffers from decryption failures as is the case for many candidate quantum-resistant cryptographic algorithms.

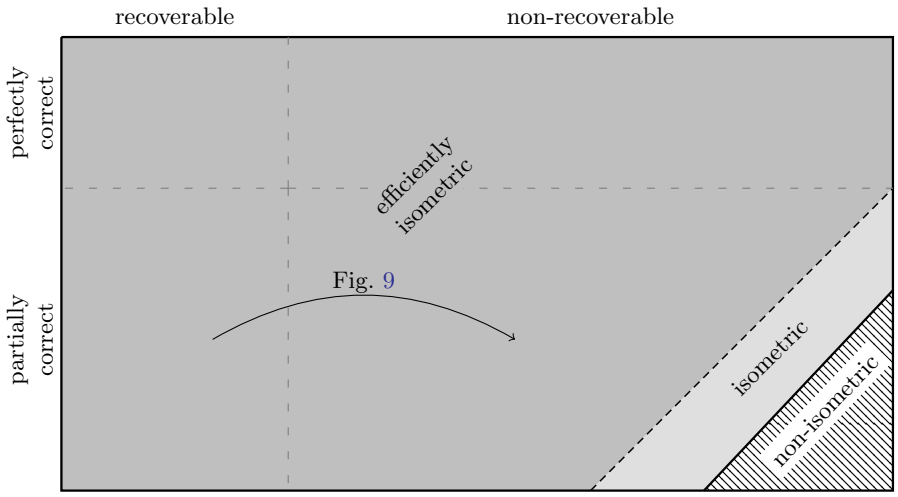

Fig. 8: Classification of PKE schemes as presented in the full version of [19].[6]

Based on the recoverability property, the following Fig. 8 classifies PKE schemes regarding the applicability of the qINDqCPA security notion.

Remark 5. Concurrently and independent of [19], Bellare et al. [4] defined the same property, which they call *randomness-based decryption*. They show that some submissions to the NIST PQC standardization process instantiate the random oracles in a way, that an adversary can recover the randomness (that is used for encryption) from the ciphertext. Based on this, they then exploit the randomness-based decryption property (i.e., the recoverability) to extract the message, thereby breaking these submission. We note that this weakness is not due to the scheme but rather the specific choice of how the random oracles are instantiated from a single hash function.

Gagliardoni et al. [19] show that most real-world PKE schemes are indeed recoverable. In fact, they do not give a real-world PKE scheme that is non-recoverable. They provide, however, a generic transformation from a recoverable PKE scheme into a non-recoverable PKE scheme via a trapdoor permutation. The transformed scheme is displayed in Fig. 9. It first permutes the message using the trapdoor permutation and afterwards encrypts the permuted message using the encryption scheme. Decryption works in the obvious reversed way, i.e., first decrypting using the encryption scheme followed by inverting the trapdoor permutation. Gagliardoni et al. [19] observe that the trapdoor permutation prevents the transformed scheme from being recoverable.

[6] The terms (efficiently) isometric and non-isometric have been introduced in the full version of [19]. They are not relevant for the work at hand, but for better comprehensibility and comparability with the original work, we decided not to remove them from this figure.

$\mathtt{KGen}^{\Sigma'}()$	$\mathtt{Enc}^{\Sigma'}(pk, m; r)$	$\mathtt{Dec}^{\Sigma'}(sk, c)$
$(pk_\Sigma, sk_\Sigma) \leftarrow\!\!\$\ \mathtt{KGen}^{\Sigma}()$	**parse** pk **as** (pk_Σ, pk_Π)	**parse** sk **as** (sk_Σ, sk_Π)
$(pk_\Pi, sk_\Pi) \leftarrow\!\!\$\ \mathtt{KGen}^{\mathcal{F}}()$	$y \leftarrow \mathcal{F}(pk_\Pi, m)$	$y \leftarrow \mathtt{Dec}^{\Sigma}(sk_\Sigma, c)$
$pk \leftarrow (pk_\Sigma, pk_\Pi)$	$c \leftarrow \mathtt{Enc}^{\Sigma}(pk_\Sigma, y; r)$	$m \leftarrow \mathcal{F}^{-1}(sk_\Pi, y)$
$sk \leftarrow (sk_\Sigma, sk_\Pi)$	**return** c	**return** m
return (pk, sk)		

Fig. 9: Transformed scheme $\Sigma' = (\mathtt{KGen}^{\Sigma'}, \mathtt{Enc}^{\Sigma'}, \mathtt{Dec}^{\Sigma'})$ as presented in the full version of [19], where $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ is a public-key encryption scheme and $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ is a deterministic trapdoor permutation.

In Theorem 6 we show that the transformed scheme as displayed in Fig. 9 indeed is non-recoverable.

Theorem 6 ([19, adapted from the full version's Theorem 26]). *Let $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ be a deterministic trapdoor permutation and $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ be a PKE scheme. Let further $\Sigma' = (\mathtt{KGen}^{\Sigma'}, \mathtt{Enc}^{\Sigma'}, \mathtt{Dec}^{\Sigma'})$ be the PKE scheme constructed from Π and Σ according to the transformation depicted in Fig. 9. If Σ is recoverable, then Σ' is non-recoverable.*

Remark 7. In the full version of [19], the theorem comprises further requirements on Π and Σ, which achieve further properties for the transformed scheme. Since these are not relevant for the remainder of this section, we omit them.

4.2 Equivalent Recoverable PKE Schemes

We refine the classification shown in Fig. 8 by identifying a set of schemes which are not recoverable but for which it is possible to transform them into a recoverable PKE scheme. We call such schemes *equivalent recoverable*. Hence, equivalent recoverable schemes are not recoverable, but for them it still holds that after transformation it is possible to decrypt a ciphertext without knowledge of the secret key, but having access to the randomness used for the encryption instead. The transformation exploits that the secret key of an equivalent recoverable scheme consists of two parts, one of which is part of the public key after transformation. Note that equivalent recoverable schemes are neither recoverable nor non-recoverable (and, hence, not all schemes that are not recoverable are non-recoverable), but a third class of PKE schemes that are worth to be studied in the context of qINDqCPA security.

Definition 8 (Equivalent recoverable PKE Scheme). *Let $\Sigma' = (\mathtt{KGen}^{\Sigma'}, \mathtt{Enc}^{\Sigma'}, \mathtt{Dec}^{\Sigma'})$ be a PKE scheme with key pair $(pk_{\Sigma'}, sk_{\Sigma'})$, using internally a deterministic trapdoor permutation $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ and a PKE scheme $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ such that $pk_{\Sigma'} = (pk_\Pi, pk_\Sigma)$ and $sk_{\Sigma'} = (sk_\Pi, sk_\Sigma)$.*

Let $\Sigma^* = (\mathtt{KGen}^{\Sigma^*}, \mathtt{Enc}^{\Sigma^*}, \mathtt{Dec}^{\Sigma^*})$ *be a transformed scheme which works identically as* Σ' *(thus,* Σ^* *and* Σ' *have the same plaintext, ciphertext, and randomness spaces* $\mathcal{M}, \mathcal{C}$*, and* $\mathcal{R}$*, respectively) except that the secret key* sk_Π *of* Π *is part of the scheme's public key, i.e.,* $pk_{\Sigma^*} = (pk_{\Sigma'}, sk_\Pi) = (pk_\Pi, pk_\Sigma, sk_\Pi)$ *and* $sk_{\Sigma^*} = sk_\Sigma$*.*

We call Σ' *an* equivalent recoverable PKE scheme *if* Σ' *is not recoverable but* Σ^* *is recoverable, i.e., if there exists an efficient algorithm* $\mathtt{Rec} : \mathcal{P}_{\Sigma^*} \times \mathcal{R} \times \mathcal{C} \to \mathcal{M}$ *such that, for any* $pk \in \mathcal{P}_{\Sigma^*}, r \in \mathcal{R}, m \in \mathcal{M}$*, it holds that*

$$\mathtt{Rec}(pk, r, \mathtt{Enc}^{\Sigma^*}(pk, m; r)) = m.$$

In particular, all schemes that are constructed from the transformation shown in Fig. 9 are equivalent recoverable, thus they are not really non-recoverable since the transformation can be inverted. In the following Fig. 10 we display an equivalent recoverable scheme after transformation.

$\mathtt{KGen}^{\Sigma^*}()$	$\mathtt{Enc}^{\Sigma^*}(pk, m; r)$	$\mathtt{Dec}^{\Sigma^*}(sk, c)$
$(pk_\Sigma, sk_\Sigma) \leftarrow\!\!\$\ \mathtt{KGen}^{\Sigma}()$	**parse** pk **as** $(pk_\Pi, pk_\Sigma, sk_\Pi)$	**parse** sk **as** (sk_Σ)
$(pk_\Pi, sk_\Pi) \leftarrow\!\!\$\ \mathtt{KGen}^{\mathcal{F}}()$	$y \leftarrow \mathcal{F}(pk_\Pi, m)$	$y \leftarrow \mathtt{Dec}^{\Sigma}(sk_\Sigma, c)$
$pk \leftarrow (pk_\Pi, pk_\Sigma, sk_\Pi)$	$c \leftarrow \mathtt{Enc}^{\Sigma}(pk_\Sigma, y; r)$	$m \leftarrow \mathcal{F}^{-1}(sk_\Pi, y)$
$sk \leftarrow (sk_\Sigma)$	**return** c	**return** m
return (pk, sk)		

Fig. 10: Transformed recoverable scheme $\Sigma^* = (\mathtt{KGen}^{\Sigma^*}, \mathtt{Enc}^{\Sigma^*}, \mathtt{Dec}^{\Sigma^*})$, where $\Sigma = (\mathtt{KGen}^{\Sigma}, \mathtt{Enc}^{\Sigma}, \mathtt{Dec}^{\Sigma})$ and $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ are a PKE scheme and a deterministic trapdoor permutation, respectively, that are used internally within an equivalent recoverable PKE scheme Σ'.

Recoverability of the transformed scheme from Fig. 10 can be easily seen. First, the recover algorithm from the underlying PKE scheme is applied to recover y from the ciphertext c, subsequently, the trapdoor permutation is inverted which can be done as sk_Π is part of the public key.

Note that the above definition of equivalent recoverable schemes does not include a statement on the security of the involved schemes. In particular, we do not claim that all equivalent recoverable schemes are as secure after transformation as they are before. This of course depends on whether the security of the equivalent recoverable scheme depends on the trapdoor permutation or only on the underlying encryption scheme. What we claim instead is that all schemes that are constructed from the transformation shown in Fig. 9 are not helpful examples for non-recoverable schemes since in fact they are equivalent recoverable. For these schemes, we observe that their security depends entirely on the underlying encryption scheme, but not on the trapdoor permutation. Hence, after applying the transformation from Fig. 10, these schemes remain as secure as they are before.

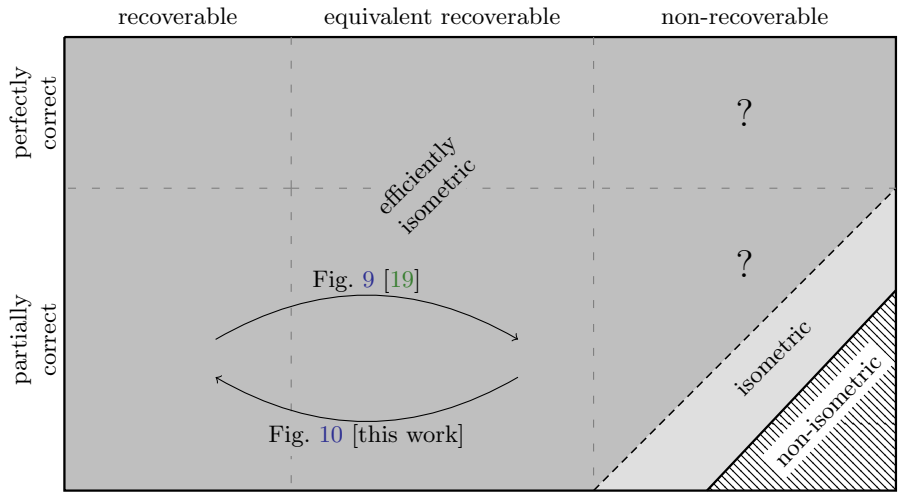

Fig. 11: Refined classification of PKE schemes. Compared to Fig. 8, equivalent recoverable schemes have been included. The transformation shown in Fig. 10 unveils that the existence of a non-recoverable PKE scheme is still unexplained.

We conclude this section with two findings: First, the transformation presented in [19] (cf. Fig. 9), that transforms a recoverable PKE scheme into a non-recoverable PKE scheme, in fact transforms the scheme into an equivalent recoverable PKE scheme, see Fig. 11. We are not aware of any transformation that transforms a recoverable PKE scheme into a non-recoverable PKE scheme. Any such transformation would, if it was reversible, entail the extension of Definition 8, hence also transform the scheme into an equivalent recoverable PKE scheme. Second, the existence of real-world schemes that are non-recoverable (and not equivalent recoverable) remains unclear. We will provide an example of such a scheme in the following section by proving that the OAEP construction is non-recoverable, i.e., both not recoverable and not equivalent recoverable.

5 OAEP

The optimal asymmetric encryption padding (OAEP) is due to Bellare and Rogaway [6]. It constructs an encryption scheme from a trapdoor permutation $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ and two hash functions G and H. The construction is illustrated in Fig. 12, the pseudocode is given in Fig. 13.[7] The OAEP construction takes a message m and randomness r and applies a two-round Feistel construction using G and H to it, yielding s and t. These values are then used as input to the trapdoor permutation $\mathcal{F}$ to compute the ciphertext c. The security of the

[7] Note that we consider the CPA-secure variant of OAEP for simplicity. The CCA-secure variant pads the message with additional 0s.

construction is based on the trapdoor permutation being partial-domain one-way (cf. Definition 3), meaning it is infeasible to compute s from a ciphertext c.

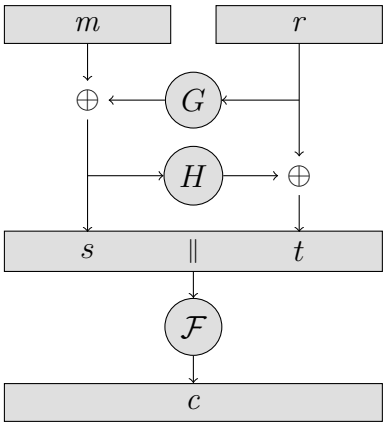

Fig. 12: The OAEP construction using a trapdoor permutation $\mathcal{F}$ and two hash functions G and H.

$\mathtt{KGen}()$	$\mathtt{Enc}(pk, m; r)$	$\mathtt{Dec}(sk, c)$
$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}^{\mathcal{F}}()$	$s \leftarrow m \oplus G(r)$	$s \parallel t \leftarrow \mathcal{F}^{-1}(sk, c)$
return (pk, sk)	$t \leftarrow r \oplus H(s)$	$r \leftarrow H(s) \oplus t$
	$c \leftarrow \mathcal{F}(pk, s \parallel t)$	$m \leftarrow s \oplus G(r)$
	return c	**return** m

Fig. 13: Pseudocode of the OAEP construction using a trapdoor permutation $\Pi = (\mathtt{KGen}^{\mathcal{F}}, \mathcal{F}, \mathcal{F}^{-1})$ and two hash functions G and H.

5.1 Recoverability of OAEP

Section 4 raises the question whether there are real-world public-key encryption scheme which are non-recoverable. In the following lemma, we answer this question in the affirmative by showing that the OAEP construction is non-recoverable, given that the trapdoor permutation is partial-domain one-way.

Lemma 9. *The OAEP construction is non-recoverable under the assumption that the trapdoor permutation is partial-domain one-way with respect to* $\mathcal{X}_1$.

Proof. We can view $G(r)$ as a one-time pad encryption of m with key s. From the randomness r, one can trivially compute $G(r)$. The ciphertext c does not reveal s due to the assumption of $\mathcal{F}$ being partial-domain one-way. Knowledge of the randomness r (corresponding to $H(s) \oplus t$) does not provide any additional information about s due to H being a random oracle, hence one-way. □

It turns out that partial-domain one-wayness is crucial for the non-recoverability of the OAEP construction—just one-wayness is not sufficient. To show this, we consider the function

$$\mathcal{F}^*(s,t) \coloneqq (s, \mathcal{F}(t)) \,.$$

If the underlying function $\mathcal{F}$ is OW-secure, then $\mathcal{F}^*$ is OW-secure but not pdOW-secure. The adversary trivially finds s as it is part of its input. Based on this, it is straightforward to show that the OAEP construction instantiated with this function is recoverable. The adversary obtains the value s from the ciphertext (simply the first part of the ciphertext). From the randomness r, it computes $G(r)$ and xors it with s to get the message m.

5.2 Quantum Operators for OAEP

In the previous section we showed that the OAEP construction is non-recoverable. This precludes the construction of the public-key-based in-place encryption operator that Gagliardoni et al. [19] provide for recoverable public-key encryption schemes. The open question is whether the in-place operator can be constructed solely using the public key, which we answer negatively here. This also answers the main question from Section 3 by showing that there are PKE schemes for which the qINDqCPA security notion cannot be defined for challengers knowing only the public key.

We first introduce a variant of pdOW, which we denote pdOW^*. The corresponding game is illustrated in Fig. 14. In this variant, the adversary receives some extra information, which corresponds to the randomness of the OAEP construction. As a first step, we show that the extra information does not help the adversary in breaking security (cf. Lemma 10). Subsequently, we show how an in-place operator for the OAEP construction can be transformed into one breaking security according to pdOW^*, yielding a contradiction (cf. Theorem 11).

The following lemma shows that, for OAEP, the extra information from game pdOW^* does not help the adversary in breaking security.

Lemma 10. *Let H be a random oracle and the games pdOW and pdOW^* be as displayed in Fig. 1 and Fig. 14, respectively. Then for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that*

$$\mathbf{Adv}^{\mathsf{pdOW}^*}_{OAEP}(\mathcal{A}) \leq 2\,\mathbf{Adv}^{\mathsf{pdOW}}_{OAEP}(\mathcal{B}) \,.$$

Proof. The proof essentially relies on the fact that r does not yield any additional information for $\mathcal{A}$. As a first step, we modify game pdOW^* by picking r uniformly

Game pdOW^*

$(pk, sk) \leftarrow\!\!\$\ \mathtt{KGen}^{\mathcal{F}}()$
$(s, t) \leftarrow\!\!\$\ \mathcal{X}_1 \times \mathcal{X}_2$
$c \leftarrow \mathcal{F}(pk, (s, t))$
$r \leftarrow H(s) \oplus t$
$s' \leftarrow \mathcal{A}(pk, c, r)$
return $s' = s$

Fig. 14: Security game pdOW^*. It is a variant of pdOW where the adversary additionally receives the xor of $H(s)$ and t as an input.

at random. To distinguish pdOW^* from the modified game, $\mathcal{A}$ needs to query H on s. Hence, a distinguishing adversary can be transformed into an adversary winning pdOW. In the modified game, the value r is independent of anything else, hence a straightforward reduction allows to transform any adversary $\mathcal{A}$, playing pdOW^*, into an adversary $\mathcal{B}$, playing pdOW, with the same winning probability. Adversary $\mathcal{B}$ simply runs $\mathcal{A}$ on its own challenge and a value r sampled uniformly at random. In addition, $\mathcal{B}$ simulates two random oracles G and H for $\mathcal{A}$. □

We finally show that an in-place encryption operator can be used to construct an xor operator for the inverse permutation $\mathcal{F}^{-1}$ underlying the OAEP construction. Hence, assuming that the in-place operator can be constructed using just the public key would yield an xor operator for the inverse permutation, contradicting its security as anyone could invert it using only the public key.

Theorem 11. *Assuming $\mathcal{F}$ to be partial-domain one-way, the in-place encryption operator for the OAEP construction instantiated with $\mathcal{F}$ cannot be constructed using solely the public key.*

Proof. For sake of contradiction, assume that the in-place encryption operator for the OAEP construction can be constructed solely using the public key. We show how to use this operator to construct an xor operator for the inverse trapdoor permutation which also only requires the public key, thus contradicting its security. The circuit is displayed in Fig. 15.

Constructing the in-place operator $U_{\mathtt{Enc}}^{(ip)}$ allows to compute its inverse operator $\left(U_{\mathtt{Enc}}^{(ip)}\right)^{\dagger}$. By definition of the operator, $\left(U_{\mathtt{Enc}}^{(ip)}\right)^{\dagger}$ on input $|r\rangle$ and $|c\rangle$ yields $|r\rangle$, $|m\rangle$, and $|0\rangle$, such that encrypting m using randomness r equals c. In the next step, the registers $|r\rangle$ and $|m\rangle$ are xored to the two output registers initialized with $|0\rangle$. The first three registers are input to $U_{\mathtt{Enc}}^{(ip)}$, yielding $|r\rangle$ and $|c\rangle$. The operator $U_G^{\oplus}$ is applied to the third and fourth register (the former being the input, the latter being the output), which yields $|r\rangle$ and $|m \oplus G(r)\rangle$. Then the operator $U_H^{\oplus}$ is applied to $|m \oplus G(r)\rangle$ (input register) and $|r\rangle$ (output register) which results in $|m \oplus G(r)\rangle$ and $|r \oplus H(m \oplus G(r))\rangle$.

By construction, it holds that the concatenation of register $|m \oplus G(r)\rangle$ and register $|r \oplus H(m \oplus G(r))\rangle$ equals the preimage of $|c\rangle$ under the function $\mathcal{F}$ (otherwise, the in-place operator would not correctly encrypt). This contradicts the security of $\mathcal{F}$ as we get an algorithm for inverting the trapdoor permutation $\mathcal{F}$ (using r as the extra information as specified in game pdOW) using only the public key. □

The above theorem shows that there are PKE schemes for which the in-place encryption operator, required for the qINDqCPA security notion, cannot be constructed using only the public key. This answers the open question from Section 3 by showing that the qINDqCPA security notion is not always applicable for the scenario of confidential communication between two parties.

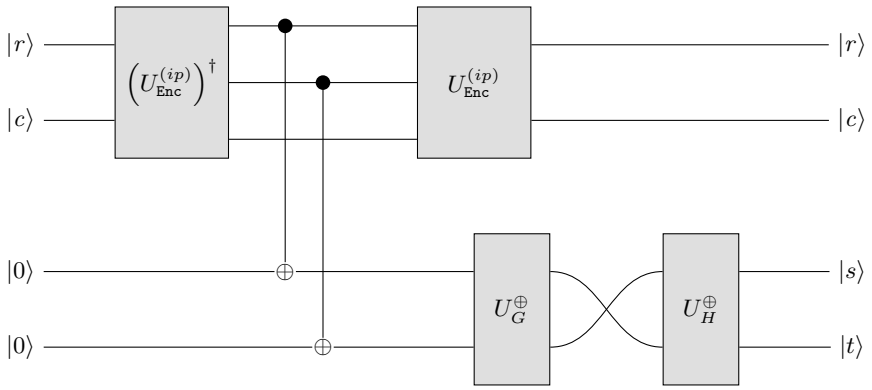

Fig. 15: Circuit for inverting $\mathcal{F}$ based on an in-place operator for `Enc` and xor operators for G and H.

Remark 12. The qINDqCPA security notion essentially provides the adversary a quantum channel transforming the message into a ciphertext, where the randomness is out of reach for the adversary. One might wonder whether ruling out the in-place quantum operator from above (cf. Equation (1)) is sufficient to argue that the qINDqCPA security is not applicable to the OAEP construction. Theoretically, one can consider alternative ways of realizing the quantum channel. Consider, for instance, that the challenger itself has some quantum channel which it uses to realize the quantum channel for the adversary. In this case, the challenger might not know the secret key but its own quantum channel uses it; the quantum channel from the qINDqCPA security notion would then still depend on the secret key. From a cryptographic perspective, we do not believe this setting to be of relevance as the security notion corresponds to communication between the adversary and a challenger, where parts (i.e., the randomness) are beyond the access of the adversary. In this sense, we believe the quantum channel realized by the quantum operator from above to be the only relevant one, while we consider other realizations to be irrelevant from a cryptographic point of view.

Acknowledgements

We thank Nina Bindel for fruitful discussions on the OAEP construction. We also thank Mariami Gachechiladze for helpful discussions about quantum channels. This work was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

References

1. Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 788–817. Springer, Heidelberg, May 2020.
2. Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 65–93. Springer, Heidelberg, April / May 2017.
3. Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In Tsuyoshi Takagi, editor, *PQCrypto 2016*, pages 44–63. Springer, Heidelberg, 2016.
4. Mihir Bellare, Hannah Davis, and Felix Günther. Separate your domains: NIST PQC KEMs, oracle cloning and read-only indifferentiability. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 3–32. Springer, Heidelberg, May 2020.
5. Mihir Bellare, Dennis Hofheinz, and Eike Kiltz. Subtleties in the definition of IND-CCA: When and how should challenge decryption be disallowed? *Journal of Cryptology*, 28(1):29–48, January 2015.
6. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, May 1995.
7. Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 592–608. Springer, Heidelberg, May 2013.
8. Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013.
9. Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 552–583. Springer, Heidelberg, December 2019.
10. Xavier Bonnetain, María Naya-Plasencia, and André Schrottenloher. On quantum slide attacks. In Kenneth G. Paterson and Douglas Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 492–519. Springer, Heidelberg, August 2019.
11. Tore Vincent Carstens, Ehsan Ebrahimi, Gelo Noel Tabia, and Dominique Unruh. Relationships between quantum IND-CPA notions. In Kobbi Nissim and Brent Waters, editors, *TCC 2021*, volume 13042 of *LNCS*, pages 240–272. Springer, 2021.

12. Céline Chevalier, Ehsan Ebrahimi, and Quoc-Huy Vu. On security notions for encryption in a quantum world. Cryptology ePrint Archive, Report 2020/237, 2020. https://eprint.iacr.org/2020/237.
13. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.
14. Mina Doosti, Mahshid Delavar, Elham Kashefi, and Myrto Arapinis. A unified framework for quantum unforgeability. *CoRR*, abs/2103.13994, 2021.
15. Ehsan Ebrahimi. Post-quantum security of plain OAEP transform. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022*, volume 13177 of *LNCS*, pages 34–51. Springer, 2022.
16. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013.
17. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. *Journal of Cryptology*, 17(2):81–104, March 2004.
18. Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 60–89. Springer, Heidelberg, August 2016.
19. Tommaso Gagliardoni, Juliane Krämer, and Patrick Struck. Quantum indistinguishability for public key encryption. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *PQCrypto 2021*, volume 12841 of *LNCS*, pages 463–482. Springer, 2021. Most of the content we refer to in this work is only included in the full version of the paper. For the full version, we refer to Cryptology ePrint Archive, Report 2020/266, https://eprint.iacr.org/2020/266.
20. Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 342–371. Springer, Heidelberg, August 2017.
21. Akinori Hosoyamada and Yu Sasaki. Quantum Demiric-Selçuk meet-in-the-middle attacks: Applications to 6-round generic Feistel constructions. In Dario Catalano and Roberto De Prisco, editors, *SCN 2018*, volume 11035 of *LNCS*, pages 386–403. Springer, Heidelberg, September 2018.
22. Gembu Ito, Akinori Hosoyamada, Ryutaroh Matsumoto, Yu Sasaki, and Tetsu Iwata. Quantum chosen-ciphertext attacks against Feistel ciphers. In Mitsuru Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 391–411. Springer, Heidelberg, March 2019.
23. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, Heidelberg, August 2016.
24. Elham Kashefi, Adrian Kent, Vlatko Vedral, and Konrad Banaszek. Comparison of quantum oracles. *Phys. Rev. A*, 65:050304, May 2002.
25. Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round feistel cipher and the random permutation. In *ISIT 2010*, pages 2682–2685, 2010.
26. Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *ISITA 2012*, pages 312–316, 2012.

27. Gregor Leander and Alexander May. Grover meets simon - quantumly attacking the FX-construction. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 161–178. Springer, Heidelberg, December 2017.
28. Shahram Mossayebi and Rüdiger Schack. Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. *CoRR*, abs/1609.03780, 2016.
29. Tristan Nemoz, Zoé Amblard, and Aurélien Dupin. Characterizing the qind-qcpa (in)security of the cbc, cfb, ofb and ctr modes of operation. *IACR Cryptol. ePrint Arch.*, page 236, 2022.
30. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016.
31. Martin Rötteler and Rainer Steinwandt. A note on quantum related-key attacks. *Inf. Process. Lett.*, 115(1):40–44, 2015.
32. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.
33. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.
34. Victor Shoup. OAEP reconsidered. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 239–259. Springer, Heidelberg, August 2001.
35. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016.