IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 1, 25 pages.

https://doi.org/10.62056/akp2fhbmo

Impossibility of Post-Quantum Shielding Black-Box Constructions of CCA from CPA

Loïs Huguenin-Dumittan and Serge Vaudenay

EPFL, Switzerland

Abstract. Proving whether it is possible to build IND-CCA public-key encryption (PKE) from IND-CPA PKE in a black-box manner is a major open problem in theoretical cryptography. In a significant breakthrough, Gertner, Malkin and Myers showed in 2007 that shielding black-box reductions from IND-CCA to IND-CPA do not exist in the standard model. Shielding means that the decryption algorithm of the IND-CCA scheme does *not* call the encryption algorithm of the underlying IND-CPA scheme. In other words, it implies that every tentative construction of IND-CCA from IND-CPA must have a re-encryption step when decrypting.
This result was only proven with respect to classical algorithms. In this work we show that it stands in a post-quantum setting. That is, we prove that there is no post-quantum shielding black-box construction of IND-CCA PKE from IND-CPA PKE. In the type of reductions we consider, i.e. post-quantum ones, the constructions are still classical in the sense that the schemes must be computable on classical computers, but the adversaries and the reduction algorithm can be quantum. This suggests that considering quantum notions, which are stronger than their classical counterparts, and allowing for quantum reductions does not make building IND-CCA public-key encryption easier.

1 Introduction

In these last few years, as promising developments have been made in the field of quantum computation, post-quantum (PQ) cryptography has been the subject of intense research. In particular, the National Institute of Standards and Technology (NIST) in the US launched a standardization process, and great efforts have been made to standardize one or several PQ signature/encryption schemes and key-encapsulation mechanisms (KEMs). In addition, many works focused on transposing classical security notions and models to the quantum setting. Among these, we can find the quantum equivalent of the random oracle model (ROM) called quantum random oracle model (QROM) [BDF+11], or different security definitions against quantum adversaries for public-key encryption (PKE), e.g. Boneh et al. [BZ13]. For instance, post-quantum indistinguishability notions like IND-CPA or IND-CCA can be defined exactly as their classical counterpart except the adversaries are assumed to be quantum. As quantum computers are strictly more powerful, it also implies that post-quantum indistinguishability is stronger than classical indistinguishability (in the sense that the former implies the latter).

In turn, this means that known relations between classical cryptographic notions do not necessarily hold in the (post-)quantum setting. In addition, such relations are usually defined in term of *reductions*, which must be redefined to capture the quantum capabilities of algorithms. Therefore, known implications and separations in the classical world need

E-mail: lois.huguenin-dumittan@epfl.ch (Loïs Huguenin-Dumittan), serge.vaudenay@epfl.ch (Serge Vaudenay)

Received: 2023-12-22 Accepted: 2024-03-05

to be re-proven in the quantum one. In this work, we focus on the following major open problem:

Is it possible to construct an IND-CCA PKE from an IND-CPA PKE, in a black-box manner?

In the random oracle model, the answer to this question is known to be positive thanks to constructions like the Fujisaki-Okamoto transform [FO99, FO13]. In the so-called *standard model* (as opposed to the ROM), the question is still open. In the classical setting, Gertner et al. [GMM07] proved a partial negative answer to the problem. They showed that no *shielding* black-box reduction from IND-CCA (even IND-CCA1) to IND-CPA exists. A shielding reduction means that the decryption algorithm of the IND-CCA PKE cannot call the encryption function of the underlying IND-CPA PKE.

Our contributions

The motivation behind this work is that Gertner et al.'s result does not deal with quantum algorithms. Our main contribution is to prove that no *post-quantum shielding* reduction from IND-CCA PKE to IND-CPA PKE exists[1]. Here, unlike in Gertner et al.'s, IND-CCA and IND-CPA are defined relative to quantum adversaries. Moreover, the reduction algorithm is assumed to be quantum as well. However, we still consider classical schemes, i.e. both the IND-CCA and IND-CPA PKEs are assumed to be computable classically. This is why we call this type of reduction *post-quantum*.

From a high-level perspective, the proof uses similar techniques as the classical one. That is, we use the well-known *two oracles* technique by Hsiao et al. [HR04], which is itself a variant of the relativizing method introduced by Impagliazzo and Rudich [IR89]. In short, we propose an oracle $\mathcal{O} = (O, R)$ relative to which IND-CPA PKEs exist but IND-CCA schemes Π^O (i.e. Π can query O but not R) do not. One of the main technical difficulties in the proof arises from the fact that the IND-CPA adversaries are quantum, and therefore have quantum access to the oracle. Therefore, we need to show that an adversary that can make quantum queries to $\mathcal{O}$ cannot break the IND-CPA scheme. Our proof relies on reductions from several hard (quantum) problems and thus minimal quantum knowledge is sufficient to verify it.

An obvious limitation, as in the original proof, is that we rule out only *shielding* reductions. However, if non-shielding constructions existed, they would imply a re-encryption step during decryption, as in the Fujisaki-Okamoto (FO) transform, which makes an IND-CCA PKE out of a IND-CPA one in the (quantum) random oracle model. We note that while our impossibility result is proven in the standard model, it should still hold in the Random Oracle Model due to the structure of the oracle O we use in the proof (i.e. O is with high probability made of random oracles). Thus, our result rules out more efficient transforms than the FO one.

Related work

Since the seminal paper by Impagliazzo et al. [IR89], the topic of black-box separation has been extensively studied (e.g. [AS16, HR04, Sim98]). In particular, as mentioned several times, the present work is a generalization of a result by Gertner et al. [GMM07]. More recently, Hosoyamada et al. [HY20] defined the notion of quantum black-box reduction. In addition, they showed that there is no quantum black-box reduction from collision-resistant hash functions to one-way permutations [HY20]. Following this work, Cao et al. [CX21]

[1] In fact, as in Gertner et al.'s, our result is slightly stronger and implies that there is no shielding reduction from IND-CCA*1* to IND-CPA.

proved that one-way permutations cannot be obtained from different flavours of one-way functions in a quantum black-box way.

Different notions of black-box reductions were first formalised by Reingold et al. [RTV04]. These were then extended by Baecher et al. [BBF13].

Technical overview

We use the two-oracle technique by Hsiao et al. [HR04] to rule out post-quantum reductions from (post-quantum) IND-CCA PKE to IND-CPA PKE. That is, we provide an oracle O that helps implement a IND-CPA PKE, and an oracle R that helps break any construction of IND-CCA PKE. More precisely, the oracle O will contain 3 sub-oracles $(\mathbf{g}, \mathbf{e}, \mathbf{d})$, where $\mathbf{g}$ is an ideal key-generation functions, $\mathbf{e}$ an ideal public-key encryption function, and $\mathbf{d}$ is the corresponding decryption function. Then, $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ will correspond to the IND-CPA PKE scheme. Note that without an additional breaking oracle R, the PKE would be IND-CCA secure against classical or quantum adversaries. Now, R is composed of additional sub-oracles, which are approximately defined as follows.

- $\mathbf{w}$, which takes as input a public key pk and encrypts each bit of the corresponding sk using $\mathbf{e}$. That is, $\mathbf{w}(\mathsf{pk}) \to (\mathbf{e}(\mathsf{pk}, \mathsf{sk}_i))_{i\in[n]}$, where sk is s.t. $\mathbf{g}(\mathsf{sk}) = \mathsf{pk}$.
- $\mathbf{u}$, which takes as input a public key pk and a ciphertext c, and outputs 1 iff both the public key and the ciphertext are valid (i.e. the public key has a corresponding secret key and the ciphertext has a corresponding pre-image under the given public key).

We note that the set of oracles $(\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$ is the same as the one used in Gertner et al.'s proof [GMM07].

Then, in order to prove the separation, we need to show two results:

1. $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ is an IND-CPA PKE even if the adversary has access to $\mathbf{w}$ and $\mathbf{u}$. In the classical setting, this is quite straightforward to prove, as was done by Gertner et al. [GMM07]. In the quantum setting, this is much more tricky as the adversary can now query $\mathbf{w}$ in superposition and the demonstration of this result turns out to be the technical contribution of the paper. Our proof involves two reductions to quantum problems. We first introduce the IMG problem, where (informally) a quantum adversary must distinguish between two sets of oracles (e_1, e_2, w_1) and (e_1, e_2, w_2), where e_1, e_2 are random injective functions and w_1 (resp. w_2) is a random function that has the same image as e_1 (resp. e_2). We then show that the IND-CPA security of $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ reduces to the IMG problem. Intuitively, in the reduction, the encryption of a 1 (resp. 0) will be simulated by a call to e_1 (resp. e_2) and w_b will simulate the encryption of a bit of sk.

   Finally, we prove that the IMG problem is hard for any quantum adversary by reducing another provably hard problem (namely the set equality problem SETEQ [Zha15]) to it. In SETEQ, the adversary is given two random injective functions f and g s.t. either f, g have the same image or have completely distinct images, and must distinguish between both cases.

   We believe this proof might be of independent interest as it shows security of (ideal) encryption even in the presence of ciphertexts that are highly correlated with the secret-key. In addition, it requires only minimal quantum knowledge to verify.

2. Any shielding construction of a PKE from O is insecure against an IND-CCA adversary having access to R. For this, we can simply reuse the proof from Gertner et al. [GMM07] as the classical adversary they build can obviously be implemented quantumly.

We conclude the proof by combining these results and applying usual separation arguments.

Outline

In Section 2, we recall security definitions and notions of quantum computing and we define post-quantum reductions. We define the oracles O and R in Section 3. Then, in Section 4, we introduce the IMG problem and prove its quantum hardness by reducing the SETEQ problem (proven to be hard by Zhandry [Zha15]) to it. We also recall a lemma by Cao et al. [CX21] that states that inverting random functions is hard for quantum algorithms, even given access to the partial inverse of the function. In Section 5, we show that an IND-CPA PKE exists relative to (O, R). The proof uses both the hardness of IMG and the lemma by Cao et al. [CX21]. Finally, in Section 6 we recall Gertner et al.'s main lemma, that says that non-shielding constructions of PKEs relative to (O, R) are not IND-CCA secure, which concludes the proof.

2 Preliminaries

2.1 Notation

If Ψ (resp. S) is a distribution (resp. a set), then $x \leftarrow\!\!\$\ \Psi$ (resp. $x \leftarrow\!\!\$\ S$) means that x is sampled uniformly at random from Ψ (resp. S). We write 1_P the indicator function that returns 1 if the predicate P holds and 0 otherwise. We denote by $[n]$ the set $\{1, \ldots, n\}$ and $\mathsf{Inj}_{i,n}$ the set of injective functions from $\{0,1\}^i$ to $\{0,1\}^n$, with $i, n \in \mathbb{N},\ i \leq n$. For any function $f : \{0,1\}^i \mapsto \{0,1\}^n$, we write $\mathsf{Im}(f)$ for the image of f, i.e. $\mathsf{Im}(f) := \{y : \exists x \in \{0,1\}^i \text{ s.t. } f(x) = y\}$. When it is clear from the context, $\cdot$ replaces any valid value. E.g. for some oracle O that takes two inputs and a valid first input a, $O(a, \cdot)$ denotes any query $O(a, b)$ for some valid second input b. Finally, we write $\mathcal{A} \Rightarrow b$ to denote the event $\mathcal{A}$ *outputs* b, where $\mathcal{A}$ is an algorithm, and for a game Γ that takes as input an adversary $\mathcal{A}$, we write $\Gamma(\mathcal{A}) \Rightarrow b$ or even $\Gamma \Rightarrow b$ to denote the event $\Gamma(\mathcal{A})$ *outputs* b.

2.2 PKE and security definitions

A Public-Key Encryption (PKE) scheme is defined as follows.

Definition 1 (Public-Key Encryption)**.** Let ℓ_1, ℓ_2, ℓ_3 and ℓ_4 be polynomial functions that describes the length of the inputs and outputs. A Public-Key Encryption scheme is made of three algorithms (G, E, D):

- $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ G(s)$: The key generation algorithm takes a string $s \in \{0,1\}^n$, where n is the security parameter, and outputs a pair of secret and public key. We let $\mathsf{sk} \in \{0,1\}^{\ell_1(n)}$ and $\mathsf{pk} \in \{0,1\}^{\ell_2(n)}$.
- $c \leftarrow\!\!\$\ E(\mathsf{pk}, m \in \{0,1\}^{\ell_4(n)})$: The encryption algorithm takes as inputs the public key pk and a plaintext $m \in \{0,1\}^{\ell_4(n)}$ and it outputs a ciphertext $c \in \{0,1\}^{\ell_3(n)}$.
- $m' \leftarrow D(\mathsf{sk}, c)$: The decryption procedure takes as inputs the secret key sk and the ciphertext $c \in \{0,1\}^{\ell_3(n)}$, and it outputs a plaintext $m' \in \{0,1\}^{\ell_4(n)} \cup \{\perp\}$, where $\perp$ denotes the error symbol.

Both the generation and encryption algorithms are probabilistic, while the decryption function is deterministic. The different length functions $\ell_i(\cdot)$ depend on the security parameter n, which we omit from now on for the sake of simplicity.

In addition, a PKE must be correct. That is, for all $m \in \{0,1\}^{\ell_4}$ and for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow\!\!\$\ G(s)$ for any $s \in \{0,1\}^n$

$$\Pr[D(\mathsf{sk}, E(\mathsf{pk}, m)) = m] = 1\ .$$

$\underline{\text{IND-CPA}_{\mathsf{PKE}}(\mathcal{A})}$

$b \leftarrow\!\$\ \{0,1\}; s \leftarrow\!\$\ \{0,1\}^n$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ G(s)$
$m_0, m_1, st \leftarrow \mathcal{A}_1(\mathsf{pk})$
$c^* \leftarrow\!\$\ E(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}_2(c^*, st)$
return $1_{b'=b}$

$\underline{\text{IND-CCA}_{\mathsf{PKE}}(\mathcal{A})}$

$b \leftarrow\!\$\ \{0,1\}; s \leftarrow\!\$\ \{0,1\}^n$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ G(s)$
$m_0, m_1, st \leftarrow \mathcal{A}_1^{\mathcal{O}^{\mathsf{Dec}_1}}(\mathsf{pk})$
$c^* \leftarrow\!\$\ E(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}_2}(c^*, st)$
return $1_{b'=b}$

Oracle $\mathcal{O}^{\mathsf{Dec}_1}(c)$

$\mathsf{pt}' \leftarrow D(\mathsf{sk}, c)$
return pt'

Oracle $\mathcal{O}^{\mathsf{Dec}_2}(c)$

if $c = c^*$: **return** $\perp$
$\mathsf{pt}' \leftarrow D(\mathsf{sk}, c)$
return pt'

Figure 1: Indistinguishability games.

Then, we recall the definitions of IND-CPA and IND-CCA security (both classical and quantum).

Definition 2 (IND-CPA)**.** We consider the IND-CPA game defined in Figure 1. A PKE scheme $\mathsf{PKE} = (G, E, D)$ is post-quantum (resp. classically) IND-CPA, if for any efficient quantum (resp. classical) adversary $\mathcal{A}$ we have

$$\mathsf{Adv}^{\mathrm{ind-cpa}}_{\mathcal{A},\mathsf{PKE}} = \left| \Pr\left[IND - CPA \Rightarrow 1\right] - \frac{1}{2} \right| = \mathsf{negl} \ .$$

Definition 3 (IND-CCA)**.** We consider the IND-CCA game defined in Figure 1. A PKE scheme $\mathsf{PKE} = (G, E, D)$ is post-quantum (resp. classically) IND-CCA, if for any efficient quantum (resp. classical) adversary $(\mathcal{A}_1, \mathcal{A}_2)$ we have

$$\mathsf{Adv}^{\mathrm{ind-cca}}_{\mathcal{A},\mathsf{PKE}} = \left| \Pr\left[IND - CCA \Rightarrow 1\right] - \frac{1}{2} \right| = \mathsf{negl} \ .$$

We refer to *post-quantum IND-CCA* as simply *IND-CCA* from now on.

2.3 Quantum algorithms

Following similar works (e.g. [HY20]), we consider the quantum circuits model of computation, where a quantum algorithm is a family of quantum circuits. In addition, as our quantum adversaries (algorithms) have access to oracles, we must define oracle-aided quantum algorithms. We use the definition of Hosoyamada et al. [HY20] modified such that we work with uniform quantum circuits. We recall that a family of uniform quantum circuits is a family of quantum circuits that can be generated by a (classical) deterministic Turing machine (see Nishimura et al. [NO02] for more details). First, similarly to previous work [HY20], we make the following assumption on the quantum oracles.

Remark 1. The oracles (classical or quantum) considered in this paper are stateless. In the quantum setting, that means the oracle does not keep a secret register that evolves with queries. Therefore, we assume that having quantum access to an oracle means having an oracle access to the corresponding unitary. The same assumption stays valid when an algorithm has oracle access to another quantum algorithm.

Definition 4 (Oracle-aided quantum algorithms)**.** A quantum oracle is a family of quantum gates $\mathcal{O} = \{\mathcal{O}_n\}_{n\in\mathbb{N}}$. Let $\mathcal{O}_1, \ldots, \mathcal{O}_t$ be a set of t quantum oracles. Then, an oracle-aided quantum algorithm $\mathcal{A}$ is a family of uniform quantum circuits $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ s.t. on a (classical) input $x \in \{0,1\}^n$, $\mathcal{A}$ runs $\mathcal{A}_n^{\mathcal{O}_{1,n},\ldots,\mathcal{O}_{t,n}}$ on the quantum state $|x, 0, 0\rangle$, measures the final

$\mathcal{B}^{\mathcal{A},\Gamma}(x)$

$j \leftarrow\$ [q]$

run $\mathcal{A}^{\Gamma}(x)$ until the j-th query to $\mathcal{O}_i$

$(z, z') \leftarrow\$$ measure first register of $|\phi^i_j\rangle$

return z

Figure 2: Algorithm $\mathcal{B}$ for Lemma 1.

state and returns the result of the output register. In other words, $\mathcal{A}_n^{\mathcal{O}_{1,n},\ldots,\mathcal{O}_{t,n}}$ can be defined as the unitary operator

$$\mathcal{A}_n^{\mathcal{O}_{1,n},\ldots,\mathcal{O}_{t,n}} = \left(\prod_{i=1}^{q}(U_{i,t,n}\mathcal{O}_{t,n}\ldots U_{i,1,n}\mathcal{O}_{1,n})\right)U_{0,n}$$

where $U_{i,j,n}, U_{0,n}$ are some unitary operators and q is the number of queries made by $\mathcal{A}_n$ to the oracles. If an oracle $\mathcal{O}$ is randomized, it is sampled from a given distribution before $\mathcal{A}$ runs $\mathcal{A}_n^{\mathcal{O}_n}$.

Now we can define the notion of query magnitude. Informally, this is the quantum equivalent to the probability that an adversary queries a certain value to an oracle.

Definition 5 (Query magnitude [HY20])**.** Let $\Gamma = (\mathcal{O}_1, \ldots, \mathcal{O}_t)$ be a set of fixed (i.e. not randomized) quantum oracles. In addition, let $|\phi^i_j\rangle$ be the state of $\mathcal{A}^{\Gamma}$ (running on some fixed input x) before the j-th query to an oracle $\mathcal{O}_i$. We can assume w.l.o.g. that the oracle $\mathcal{O}_i$ acts on the first $inp_i + out_i$ qubits of $|\phi^i_j\rangle$ (i.e. inp_i qubits of input and out_i qubits of output). Then there exist $\alpha_z \in \mathbb{C}$ and a state $|\psi_z\rangle$ s.t.

$$|\phi^i_j\rangle = \sum_{z\in\{0,1\}^{inp_i}} \alpha_z|z, \psi_z\rangle \ .$$

The *query magnitude* of z before the j-th query of $\mathcal{A}^{\Gamma}(x)$ to $\mathcal{O}_i$, for an input $x \in \{0,1\}^n$ is

$$\mu^{\mathcal{A},\mathcal{O}_i}_{z,j}(x) := |\alpha_z|^2 \ .$$

Note that if one measures the first inp_i qubits of $|\phi^i_j\rangle$, z will be the result with probability $\mu^{\mathcal{A},\mathcal{O}_i}_{z,j}(x) = |\alpha_z|^2$.

The *total query magnitude* of z is simply the sum of the query magnitude over all queries $1 \le j \le q$ made by the adversary to $\mathcal{O}_i$:

$$\mu^{\mathcal{A},\mathcal{O}_i}_{z}(x) := \sum_{j=1}^{q} \mu^{\mathcal{A},\mathcal{O}_i}_{z,j}(x) \ .$$

Definition 6 (Quantum-accessible oracles)**.** Let $\mathcal{O}$ be any classical oracle. The quantum-accessible oracle O induced by $\mathcal{O}$ is a quantum oracle defined as the unitary operator $O : |x, y\rangle \mapsto |x, y+\mathcal{O}(x)\rangle$ for any classical inputs x and y. For the sake of simplicity, when it is clear from the context, we denote by $\mathcal{O}$ both a classical oracle and its quantum-accessible oracle counterpart.

Now we can state the following lemma, which will be useful in our proof. Informally, this lemma says that if a quantum algorithm can distinguish an oracle $\mathcal{O}$ from the same oracle where all values $\mathcal{O}(z, \cdot)$ for z have been changed, then one can extract z with good probability.

$\Psi'_{\Psi,\mathcal{D}_{x,z,\Gamma}}$

$(x, z, \Gamma) \leftarrow\!\$\ \Psi$
parse $(\mathcal{O}_1, \ldots, \mathcal{O}_t) \leftarrow \Gamma$
for $i \in \{1, \ldots, t\}$:
  $\mathcal{O}'_i \leftarrow \mathcal{O}_i$
  for $z' \in \{0,1\}^{inp_i - k}$:
    $y \leftarrow\!\$\ \mathcal{D}_{x,z,\Gamma}$
      // $\mathcal{O}_i = \mathcal{O}'_i$ except on values of the form $(z, \cdot)$
    $\mathcal{O}'_i(z, z') \leftarrow y$
set $\Gamma' \leftarrow (\mathcal{O}'_1, \ldots, \mathcal{O}'_t)$
return (x, z, Γ, Γ')

Figure 3: Distribution Ψ' induced by Ψ and $\mathcal{D}_{x,z,\Gamma}$ for Lemma 1.

Lemma 1. *Let $n, t \in \mathbb{Z}$ be some integers and Ψ be some distribution that outputs a tuple (x, z, Γ), where $\Gamma = (\mathcal{O}_1, \ldots, \mathcal{O}_t)$ is a sequence of t sub-oracles $\mathcal{O}_i : \{0,1\}^{inp_i} \mapsto \{0,1\}^{out_i}$, $x \in \{0,1\}^n$, and $z \in \{0,1\}^k$ for some $k < inp_i$. In addition, let $\mathcal{D}_{x_d,z_d,\Gamma_d}$ be a distribution parametrized by a tuple (x_d, z_d, Γ_d) that is in the same domain as the output of Ψ defined above.*

Then, the distribution Ψ' induced by Ψ and $\mathcal{D}_{x_d,z_d,\Gamma_d}$ is defined by the sampling algorithm given in Figure 3:

We consider the quantum algorithm $\mathcal{B}$ presented in Figure 2. Then, for any oracle-aided quantum algorithm $\mathcal{A}$ limited to q quantum queries to Γ (or Γ') and any output y

$$\left|\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y] - \Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right| \leq 2q\sqrt{\Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z]}\ ,$$

where $(x, z, \Gamma, \Gamma') \leftarrow\!\$\ \Psi'$ and the probabilities are taken over the internal randomness of the adversaries, the randomness of measurements, and the sampling from Ψ'.

Proof. The proof is an application of the Generalized Swapping Lemma [HY20] and can be found in Appendix A. □

One can observe that the above lemma is a generalised version of the well-known One-Way-to-Hiding (OW2H) lemma [Unr15].

2.4 Post-Quantum reductions

We first define a classical primitive as Baecher et al. [BBF13].

Definition 7 (Algorithm computing a random variable)**.** We say an algorithm $\mathcal{A}$ computes a random variable A if $\mathcal{A}$ produces an output with the same distribution as A. In the following, we often write $\mathcal{A}$ to denote both a random variable and the algorithm that computes it.

Definition 8 (Classical Primitive)**.** A (classical) primitive $\mathcal{P}$ is a tuple $(\mathcal{F}_\mathcal{P}, \mathcal{R}_\mathcal{P})$, where $\mathcal{F}_\mathcal{P}$ is a set of random variables and $\mathcal{R}_\mathcal{P}$ is a relation between two random variables.

A classical algorithm (i.e. Turing machine) *implements* $\mathcal{P}$, or is an *implementation of* $\mathcal{P}$, if it computes f for some $f \in \mathcal{F}_\mathcal{P}$.

A classical/quantum adversary "*breaks* f" if it computes $\mathcal{A}$ s.t. $(f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$.

Finally, let $f \in \mathcal{F}_\mathcal{P}$ be efficiently computable by a classical algorithm, then if there is no efficient classical (resp. quantum) algorithm $\mathcal{A}$ s.t. $(f, \mathcal{A}) \in \mathcal{R}_\mathcal{P}$, we say f is *secure* (resp. *post-quantum secure*).

Remark. In this work, we are interested by classically computable primitives (PKEs) that might resist quantum adversaries. Therefore, we do not consider quantum implementations but only quantum adversaries. That is, any implementation can be computed by a classical algorithm but the set of adversaries is the set of efficient quantum algorithms.

Finally, we define the notion of post-quantum black-box reduction.

Definition 9 (Post-Quantum black-box reduction)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be classical primitives. There exists a post-quantum black-box reduction from $\mathcal{Q}$ to $\mathcal{P}$ if there exist an efficient classical algorithm G and an efficient quantum algorithm $\mathcal{S}$ s.t.

1. For every (classically computable) $f \in \mathcal{F}_{\mathcal{P}}$, then $G^f \in \mathcal{F}_{\mathcal{Q}}$.
2. For every quantum adversary $\mathcal{A}$ and (implementation of) $f \in \mathcal{F}_{\mathcal{P}}$, if $(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{Q}}$ then $(f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{P}}$.

The second condition can be rewritten as

$$\exists \mathsf{EFF_c}\ G\ \exists \mathsf{EFF_q}\ \mathcal{S} \quad \forall \mathcal{A} \quad \forall f \in \mathcal{F}_{\mathcal{P}}$$
$$(G^f, \mathcal{A}^f) \in \mathcal{R}_{\mathcal{Q}} \Rightarrow (f, \mathcal{S}^{\mathcal{A},f}) \in \mathcal{R}_{\mathcal{P}}$$

where $\mathsf{EFF_c}$ and $\mathsf{EFF_q}$ stand for efficient classical and efficient quantum, respectively.

In the post-quantum black-box reduction defined above, we start with a classical primitive $\mathcal{P}$ meant to be post-quantum secure. Then, for a black-box reduction to exist, there must be a classical algorithm that builds a primitive $\mathcal{Q}$ using $\mathcal{P}$. In addition, there must be an efficient quantum reduction algorithm $\mathcal{S}$, which, given quantum black-box access to any (even non efficient) adversary that breaks $\mathcal{Q}$, builds an adversary that breaks $\mathcal{P}$.

Ruling out post-quantum reductions. We show in the following lemma that a two oracles argument as described by Hsiao et al. [HR04] is sufficient to rule out post-quantum reductions. The proof is basically the same as in the classical setting.

Lemma 2. *Let $\mathcal{P}$ and $\mathcal{Q}$ be classical primitives. Then, there is no post-quantum reduction from $\mathcal{Q}$ to $\mathcal{P}$ if there exist oracles (O, R) s.t.*

1. *There exist efficient classical algorithms f s.t. $f^O \in \mathcal{F}_{\mathcal{P}}$.*
2. *For all efficient classical algorithms G:*
   - *there is an efficient quantum adversary $\mathcal{A}$ s.t. $(G^{f^O}, \mathcal{A}^{O,R}) \in \mathcal{R}_{\mathcal{Q}}$*
   - *for all efficient quantum algorithms $\mathcal{S}$ then $(f^O, \mathcal{S}^{f,O,R}) \notin \mathcal{R}_{\mathcal{P}}$*

Proof. For the sake of contradiction, we assume a pair of oracles (O, R) fulfilling the conditions in Lemma 2 exists and a post-quantum reduction from $\mathcal{Q}$ to $\mathcal{P}$ exists as well. Let f be the algorithm s.t. $f \in \mathcal{F}_{\mathcal{P}}$ as specified in condition (1). By condition (2), we have that for all G there is an efficient quantum adversary $\mathcal{A}$ s.t. $(G^{f^O}, \mathcal{A}^{O,R}) \in \mathcal{R}_{\mathcal{Q}}$. By the existence of the post-quantum reduction, it means that there exists an efficient quantum reduction $\mathcal{S}$ s.t. $(f^O, \mathcal{S}^{\mathcal{A}',f}) \in \mathcal{R}_{\mathcal{P}}$ with $\mathcal{A}' := \mathcal{A}^{O,R}$. Now, as $f, \mathcal{A}$ are efficient classical and quantum algorithms, one can embed these in $\mathcal{S}$. Hence, there exists an efficient $\mathcal{S}$ s.t. $(f^O, \mathcal{S}^{O,R}) \in \mathcal{R}_{\mathcal{P}}$. This contradicts the second part of condition (2), which completes the proof. □

Informally, the two oracles technique works as follows. One builds an oracle O that trivially implements the primitive $\mathcal{P}$ (i.e. the primitive exists relative to O). Then, we build another oracle R and we show that the primitive is secure against even unbounded quantum adversaries (with bounded number of quantum queries to (O, R)). In particular, this implies that all the security of the primitive must come from O. In a second step, we show that there exists an inefficient adversary $\mathcal{A}$ (with bounded number of queries to (O, R)) that breaks any implementation of $\mathcal{Q}$ relative to O. Then, in a final step, it is argued that $\mathcal{A}$ can be made efficient. In the classical setting, this is done by assuming P = NP or by embedding a PSPACE oracle in R. Looking ahead, this will be sufficient in our case as $\mathcal{A}$ will be classical in our proof. Lemma 2 then states that this technique is sufficient to rule out post-quantum black-box reductions.

3 The Oracle $\mathcal{O}$

We recall that we want to rule out reductions from IND-CCA to IND-CPA using Lemma 2. That is, we wish to find an oracle $\mathcal{O} = (O, R)$ s.t. an IND-CPA PKE exists relative to this oracle, but IND-CCA PKEs do not. We consider here PKEs that encrypt 1 bit, as they are known to imply PKEs for longer messages [MS09]. We use the same oracle as the one defined by Gertner et al. [GMM07].

Definition 10 (Oracle $\mathcal{O}$)**.** The oracle $\mathcal{O}$ is made of several sub-oracles, more precisely $\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{u}, \mathbf{w})$. Each sub-oracle will play a part in the proof: $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ will correspond to the IND-CPA PKE $(\mathbf{w}, \mathbf{u})$ will help the IND-CCA adversary break the underlying IND-CPA PKE in order to win its own game. More precisely, if we follow the notation of Lemma 2, $O = (\mathbf{g}, \mathbf{e}, \mathbf{d})$ and $R = (\mathbf{u}, \mathbf{w})$.

We now formalize how an oracle

$$\mathcal{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{u}, \mathbf{w}) \leftarrow\!\!\$\ \Psi$$

is sampled. For each $n \in \mathbb{N}$, each sub-oracle is generated as follows.

- $\mathbf{g}$: $\{0,1\}^n \mapsto \{0,1\}^{3n}$ is a random length-tripling one-to-one function. This function will be used as a key-generation function that outputs a public key given a secret key.
- $\mathbf{e}$: $\{0,1\}^{3n} \times \{0,1\} \times \{0,1\}^n \mapsto \{0,1\}^{3n}$ is s.t. $\mathbf{e}(\mathsf{pk}, \cdot, \cdot)$ is a random one-to-one function for all fixed pk. $\mathbf{e}$ will be used as a bit-encryption function.
- $\mathbf{d}$: $\{0,1\}^n \times \{0,1\}^{3n} \mapsto \{0,1,\perp\}$ is deterministically defined as follows. $\mathbf{d}(\mathsf{sk}, c)$ outputs b s.t. $\mathbf{e}(\mathbf{g}(\mathsf{sk}), b, r) = c$ if such r exists. If not, $\mathbf{e}$ outputs $\perp$. This oracle will be used as a decryption function.
- $\mathbf{w}$: $\{0,1\}^{3n} \times \{0,1\}^n \mapsto \{0,1\}^{3n \times n} \cup \{\perp\}$ is defined as follows. The function takes a public key pk and an index i as inputs, and outputs $\perp$ if there is no unique sk' s.t. $\mathbf{g}(\mathsf{sk}') = \mathsf{pk}$. Otherwise, $\mathbf{w}(\mathsf{pk}, i)$ returns a vector of n encryptions of the bits of sk' $(\mathbf{e}(\mathsf{pk}, \mathsf{sk}'_1, r_{1,i,\mathsf{pk}}), \ldots, \mathbf{e}(\mathsf{pk}, \mathsf{sk}'_n, r_{n,i,\mathsf{pk}}))$, where the $r_{k,i,\mathsf{pk}}$ are sampled at random when (pk, i) is queried for the first time. This function returns the bit-by-bit encryption of the secret-key corresponding to the input public key, with different random coins indexed by i.
- $\mathbf{u}$: $\{0,1\}^{3n} \times \{0,1\}^{3n} \mapsto \{\perp, \top\}$ takes a public key pk and a ciphertext c as inputs and returns $\top$ if $\exists b, r$ s.t. $\mathbf{e}(\mathsf{pk}, b, r) = c$. Otherwise it returns $\perp$. This function returns whether a ciphertext is valid or not.

4 Hard Problems

We introduce in this section several quantum hard problems that will be used to prove our main technical result.

First, we recall the definition of the (average) set quality (SETEQ) problem.

Definition 11 (SETEQ)**.** Let $\mathsf{Inj}_{n,m}$ be the set of one-to-one functions from $\{0,1\}^n$ to $\{0,1\}^m$. We define $\mathcal{F}_n^b$ as the following distribution.

- If $b = 0$: Sample $f, g \leftarrow\!\!\$\ \mathsf{Inj}_{n,n+1}$ s.t. $\mathsf{Im}(f) = \mathsf{Im}(g)$.
- If $b = 1$: Sample $f, g \leftarrow\!\!\$\ \mathsf{Inj}_{n,n+1}$ s.t. $\mathsf{Im}(f) \cap \mathsf{Im}(g) = \emptyset$.

The SETEQ problem is hard if for any (possibly unbounded) quantum adversary $\mathcal{A}$ that makes $\mathsf{poly}(n)$ quantum queries to f, g

$$\left|\Pr[\mathcal{A}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^1] - \Pr[\mathcal{A}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^0]\right| = \mathsf{negl}(n)$$

where the probabilities are taken over the quantum randomness and the sampling of f, g.

It turns out the SETEQ problem is hard, according to the following theorem by Zhandry [Zha15].

Theorem 1 (Hardness of SETEQ [Zha15])**.** *Let $\mathcal{F}_n^b$ be as defined above. Then, for any quantum adversary we have*

$$\left|\Pr[\mathcal{A}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^1] - \Pr[\mathcal{A}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^0]\right| = O(q^3/2^n)$$

where q is the number of queries $\mathcal{A}$ makes to f and g.

We now introduce an intermediary problem that we call the IMG problem.

Definition 12 (IMG problem)**.** Let $e_0 : \{0,1\}^n \mapsto \{0,1\}^{3n}$ and $e_1 : \{0,1\}^n \mapsto \{0,1\}^{3n}$ be random one-to-one functions s.t. $\mathsf{Im}(e_0) \cap \mathsf{Im}(e_1) = \emptyset$. I.e. e_0 and e_1 are sampled uniformly at random from the set of pairs of injective functions with disjoint images. Let $f : \{0,1\}^n \mapsto \{0,1\}^n$ be a random function. We define $w_b(\cdot) := e_b(f(\cdot))$. In addition, we define an helper oracle $u(c)$ that returns $\top$ if $c \in \mathsf{Im}(e_0) \cup \mathsf{Im}(e_1)$ and $\bot$ otherwise. The IMG problem is considered hard if for every (possibly unbounded) quantum adversary $\mathcal{A}$ that makes $\mathsf{poly}(n)$ quantum queries to e_0, e_1, w_b, u, we have

$$|\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] - \Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 1]| = \mathsf{negl}(n)\ ,$$

where the probabilities are taken over the quantum randomness and the sampling of e_0, e_1, f. Concretely, this problem is hard if with a polynomial number of quantum queries one cannot say whether w_b has the same image as e_0 or e_1. Note that we could also define w_b as a random function with domain $\{0,1\}^n$ and codomain $\mathsf{Im}(e_b)$.

Jumping ahead, we will use the above problem with e_b defined as $\mathbf{e}(\mathsf{pk}^*, b, \cdot)$, u as $\mathbf{u}$ and w_b as one part of the $\mathbf{w}$ oracle.

Using this result, we prove that the IMG problem is hard by showing that SETEQ reduces to it.

Lemma 3 (SETEQ reduces to IMG)**.** *Let $\mathcal{F}_n^b$ be as defined in the* SETEQ *problem and e_0, e_1, w_b, u as defined in the* IMG *problem. Then, for any* IMG *quantum adversary one can build a* SETEQ *adversary such that*

$$\begin{aligned}&|\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] - \Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 1]| \leq \\ &\left|\Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^1] - \Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^0]\right|\end{aligned}$$

where the number of queries made by $\mathcal{B}$ is roughly twice the number made by $\mathcal{A}$.

Proof. We first state the idea of the proof. In the SETEQ problem, when $b = 1$ (thus $\mathsf{Im}(f) \cap \mathsf{Im}(g) = \emptyset$) one can set $e_0 = f$ and $e_1 = g$ and $w_{b'} = e_{b'} \circ r$ with b' picked at random and r a random function. Minus some technical details, this perfectly simulates an instance of the IMG problem and the probability that the IMG adversary $\mathcal{A}$ outputs b' is the advantage of $\mathcal{A}$ (plus or minus $\frac{1}{2}$) in the IMG problem. Then, if $b = 0$, images of e_0 and e_1 will be the same and it is impossible to distinguish w_0 from w_1. Thus, in this case $\mathcal{A}$ outputs 0 or 1 with probability $\frac{1}{2}$. Hence, if $\mathcal{A}$ makes the correct guess with probability p in a correct instance of the IMG problem, the SETEQ reduction $\mathcal{B}$ has an advantage of $p - \frac{1}{2}$, which is equal to $\mathcal{A}$'s advantage.

More formally, the reduction $\mathcal{B}^{f,g}$ sets $\mathcal{A}$'s oracles as follows. First, $\mathcal{B}$ samples a random one-to-one function $h \leftarrow\!\!\$\ \mathsf{Inj}_{n+1,3n}$, a random function $r : \{0,1\}^n \mapsto \{0,1\}^n$ and a random bit b'. Then, each oracle is set as

- $e_0 := h \circ g$.
- $e_1 := h \circ f$.
- $w_{b'} := e_{b'} \circ r$.
- $u(c)$: return $\top$ if $c \in \mathsf{Im}(h)$, otherwise return $\bot$. Note that the check $c \in \mathsf{Im}(h)$ can be done because $\mathcal{B}$ is an unbounded adversary which sampled h.

Each oracle can be implemented in a quantum circuit that makes 2 calls to the quantum oracles f or g. For instance, the unitary $U_{e_0} : |x, y, z\rangle \mapsto |x, y+e_0(x), z\rangle$ can be implemented as

$$U_{e_0} : |x, y, 0, z\rangle \xrightarrow{g} |x, y, g(x), z\rangle \xrightarrow{h} |x, y + h(g(x)), g(x), z\rangle \xrightarrow{g} |x, y + h(g(x)), 0, z\rangle \ .$$

The adversary $\mathcal{B}^{f,g}$ runs $b'' \leftarrow \mathcal{A}^{e_0,e_1,w_{b'},u}$ and returns $1_{b'=b''}$. We distinguish two cases:

- $b = 1$ ($\mathsf{Im}(f) \cap \mathsf{Im}(g) = \emptyset$): By definition g and f are one-to-one functions from $\{0,1\}^n$ to $\{0,1\}^{n+1}$ and h is a random one-to-one function from $\{0,1\}^{n+1}$ to $\{0,1\}^{3n}$. Moreover, as the images of g and f are distinct, e_0 and e_1 are random one-to-one functions from $\{0,1\}^n$ to $\{0,1\}^{3n}$ s.t. $\mathsf{Im}(e_0) \cap \mathsf{Im}(e_1) = \emptyset$. In addition, $w_{b'}$ is defined as $e_{b'} \circ r$ and $u(c)$ returns whether $c \in \mathsf{Im}(e_0) \cup \mathsf{Im}(e_1)$. Therefore,

$$\begin{aligned}\Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^1] &= \Pr[\mathcal{A}^{e_0,e_1,w_{b'},u} \Rightarrow b' : b' \leftarrow\!\!\$\ \{0,1\}] \\ &= \frac{1}{2}\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] + \frac{1}{2}\Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 0]\end{aligned}$$

  where $(e_0, e_1, w_{b'}, u)$ follow the same distribution as in the IMG problem.

- $b = 0$ ($\mathsf{Im}(f) = \mathsf{Im}(g)$): In this case, $\mathsf{Im}(e_0) = \mathsf{Im}(e_1) = \mathsf{Im}(w_0) = \mathsf{Im}(w_1)$. As r is a random function and cannot be accessed by the adversary, w_0 and w_1 are perfectly indistinguishable. More precisely, given all values of $e_0, e_1, w_{b'}$ (we omit u as it is independent of b'), the optimal distinguisher would output the b that maximizes $\Pr[e_b(r(0)) = w_{b'}(0), \ldots, e_b(r(2^n - 1)) = w_{b'}(2^n - 1)|w_{b'}, e_0, e_1]$. The only randomness here is the one from r, as all values of $e_0, e_1, w_{b'}$ are known. Now,

$$\begin{aligned}&\Pr_r[e_b(r(0)) = w_{b'}(0), \ldots, e_b(r(2^n - 1)) = w_{b'}(2^n - 1)|w_{b'}, e_0, e_1] = \\ &\Pr_r[r(0) = e_b^{-1}(w_{b'}(0)), \ldots, r(2^n - 1) = e_b^{-1}(w_{b'}(2^n - 1))|w_{b'}, e_0, e_1] = \\ &\frac{1}{2^{n2^n}}\end{aligned}$$

for both $b' = 0$ or $b' = 1$, as r is a random function. Hence, even with an unbounded number of queries to $e_0, e_1, w_{b'}$, $\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] = \Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 1]$. Therefore,

$$\Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^0] = \Pr[\mathcal{A}^{e_0,e_1,w_{b'},u} \Rightarrow b' : b' \leftarrow\!\!\$\ \{0,1\}] = \frac{1}{2}\ .$$

Finally, we get that for any IMG adversary $\mathcal{A}$ that makes q quantum queries, there exists an (unbounded) SETEQ adversary $\mathcal{B}$ s.t.

$$\begin{aligned}
&2 \cdot \left|\Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^1] - \Pr[\mathcal{B}^{f,g} \Rightarrow 1 : f, g \leftarrow\!\!\$\ \mathcal{F}_n^0]\right| \\
&= 2 \cdot \left|\Pr[\mathcal{A}^{e_0,e_1,w_{b'},u} \Rightarrow b' : b' \leftarrow\!\!\$\ \{0,1\}] - \frac{1}{2}\right| \\
&= \left|\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] - \Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 1]\right|
\end{aligned}$$

where $\mathcal{B}$ makes at most $2q$ queries, which concludes the proof. □

Corollary 1 (Hardness of IMG)**.** *The* IMG *is hard for quantum algorithms. More precisely, for any* IMG *quantum adversary, we have*

$$\left|\Pr[\mathcal{A}^{e_0,e_1,w_1,u} \Rightarrow 1] - \Pr[\mathcal{A}^{e_0,e_1,w_0,u} \Rightarrow 1]\right| = O(q^3/2^n)$$

where q is the number of quantum queries made by $\mathcal{A}$.

Finally, we define partial inverse functions and recall a Lemma by Cao et al. [CX21].

Definition 13 (Partial inverse function)**.** Let $f : \{0,1\}^n \mapsto \{0,1\}^{n+m}$ be some injective function and $x^* \in \{0,1\}^n$. Then, we define the partial inverse function $f^{-1}_{\neq x^*}$ as

$$f^{-1}_{\neq x^*}(y) = \begin{cases} x, & \text{if } \exists x \neq x^* \text{ s.t. } f(x) = y \\ \perp, & \text{if } \nexists x \text{ s.t. } f(x) = y \\ \perp, & \text{if } y = f(x^*) \end{cases}\ .$$

In other words, $f^{-1}_{\neq x^*}$ inverts f except on $y = f(x^*)$.

Lemma 4 (Lemma 5 [CX21])**.** *Let $f \leftarrow\!\!\$\ \mathsf{Inj}_{n,n+m}$ be a random injective function, $x^* \leftarrow\!\!\$\ \{0,1\}^n$, and $f^{-1}_{\neq x^*}$ be the partial inverse function. Then, for any (possible unbounded) quantum adversary $\mathcal{A}$ making $\mathsf{poly}(n)$ quantum queries to $f, f^{-1}_{\neq x^*}$, we have*

$$\Pr[\mathcal{A}^{f,f^{-1}_{\neq x^*}}(f(x^*)) \Rightarrow x^* : x^* \leftarrow\!\!\$\ \{0,1\}^n, f \leftarrow\!\!\$\ \mathsf{Inj}_{n,n+m}] = \mathsf{negl}(n)$$

where the probability is taken over the randomness of $\mathcal{A}, f$ and x^*. I.e. inverting $f(x^*)$ given f and the partial inverse function is hard.

5 Existence of IND-CPA PKE relative to $\mathcal{O}$

We first define what a (1-bit) *PKE relative to an oracle* is.

Definition 14 (PKE relative to $\mathbf{O}$)**.** Let $\mathbf{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d})$ be an oracle. A *valid* PKE construction relative to $\mathbf{O}$ is of the form $\mathsf{PKE}^{\mathbf{O}} = (\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$, where for all $n \in \mathbb{N}$ and some constants $\rho_0, \rho_1, \rho_2, \rho_3$ $(\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$ are as follows.

- $\mathbf{G}^{\mathbf{O}} : \{0,1\}^n \mapsto \{0,1\}^{n^{\rho_0}} \times \{0,1\}^{n^{\rho_1}}$. We consider $\mathbf{G}^{\mathbf{O}}(S) = (SK, PK)$ as a key generation function that takes a seed S and outputs a pair of secret/public keys (SK, PK).

- $\mathbf{E}^{\mathbf{O}} : \{0,1\}^{n^{\rho_1}} \times \{0,1\} \times \{0,1\}^{n^{\rho_2}} \mapsto \{0,1\}^{n^{\rho_3}}$. We consider $\mathbf{E}^{\mathbf{O}}(PK, M, R) = C$ as an encryption function that takes as inputs a public key PK, a bit M and random coins R, and outputs a ciphertext C.
- $\mathbf{D}^{\mathbf{O}} : \{0,1\}^{n^{\rho_0}} \times \{0,1\}^{n^{\rho_3}} \mapsto \{0,1\} \cup \{\perp\}$. We consider $\mathbf{D}^{\mathbf{O}}(SK, C) = M'$ as a decryption function that takes as inputs a secret-key SK and a ciphertext C, and outputs a plaintext bit M' or the error symbol $\perp$.

We also require perfect correctness, that is for any $M \in \{0,1\}, R \in \{0,1\}^{n^{\rho_2}}$ and $S \in \{0,1\}^n$,

$$\mathbf{D}^{\mathbf{O}}(SK, \mathbf{E}^{\mathbf{O}}(PK, M, R)) = M$$

for $(SK, PK) = \mathbf{G}^{\mathbf{O}}(S)$. In addition, w.l.o.g. we assume there are constants s and q s.t. for any security parameter n, $(\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$ make at most n^q queries to $\mathbf{O}$ and each query is at most of size n^s. In addition, the running time of $(\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$ must be polynomial in n as well.

We now prove the main theorem, that is $(\mathbf{g}, \mathbf{e}, \mathbf{d})$ is IND-CPA relative to the oracle $\mathcal{O}$ (see Definition 10).

Theorem 2. *Let* $\mathsf{PKE_q}^{\mathcal{O}} = (g^{\mathbf{g}}, \mathbf{e}, \mathbf{d})$ *be a PKE relative to* $\mathcal{O}$*, where* $g^{\mathbf{g}}(s)$ *sets* $\mathsf{sk} \leftarrow s$ *and returns* $(\mathsf{sk}, \mathbf{g}(\mathsf{sk}))$*. Then, for any (possibly unbounded) quantum adversary* $\mathcal{A}$ *we have*

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A}^{\mathcal{O}}, \mathsf{PKE_q}^{\mathcal{O}}} = \mathsf{negl}(n) \ ,$$

where the number of quantum queries made by $\mathcal{A}$ *to* $\mathcal{O}$ *is polynomial in* n*.*

Proof. The idea of the proof is the following. We first modify the game such that the decryption oracle $\mathbf{d}$ does not reply when queried with sk^*. We call this new game Γ^1 and we use Lemma 1 to argue that the advantage difference between both games is upper bounded by the probability that some adversary $\mathcal{B}$ inverts pk^* having access to $\mathbf{g}, \mathbf{e}, \mathbf{w}, \mathbf{u}$ and the modified decryption oracle. We then iteratively modify the oracle $\mathbf{w}$ into oracles $\mathbf{w}^1, \dots, \mathbf{w}^n$ s.t. the first i ciphertexts output by $\mathbf{w}^i(\mathsf{pk}^*, \cdot)$ are encryptions of 0's (instead of the bits of sk^*). The advantage difference between these n games hops can be upper bounded by the probability of winning the IMG problem, which is negligible. Then, the advantage of $\mathcal{B}$ in the final game (i.e. with $\mathbf{w}^n$) can be upper bounded by the advantage of another adversary inverting g having access to a partial invert oracle $g^{-1}_{\neq \mathsf{sk}^*}$. The idea is that $g^{-1}_{\neq \mathsf{sk}^*}$ can be used to simulate $\mathbf{w}^n(\mathsf{pk}, \cdot)$ as we can either invert $\mathsf{pk} \neq \mathsf{pk}^*$ and encrypt the result, or encrypt 0 bits if $\mathsf{pk} = \mathsf{pk}^*$.

Next, we modify Γ^1 into another game Γ^2 such that the oracle $\mathbf{e}$ does not reply on queries of the form $\mathbf{e}(\mathsf{pk}^*, \cdot, r^*)$, where r^* are the coins used to compute the challenge ciphertext c^*. We apply Lemma 1 again to deduce that the advantage difference between both games is roughly upper bounded by the advantage of some adversary $\mathcal{B}$ in finding r^* given pk^* and c^*. Then, this advantage can be upper bounded by the probability an adversary inverts a function given access to its partial inverse. Intuitively, $\mathbf{e}$ is the function to invert and the modified $\mathbf{d}$ is the partial inverse as it does not reply on queries of the form $\mathbf{d}(\mathsf{sk}^*, \cdot)$. Finally, in Γ^2, the oracles give no information on c^* as $\mathbf{d}$ cannot decrypt it and $\mathbf{e}(\mathsf{pk}^*, \cdot, r^*)$ returns $\perp$. That concludes the proof.

We proceed with a sequence of hybrid games Γ^0-Γ^2 shown in Figure 4.

Γ^0: It is the original IND-CPA game. We recall that a quantum (sub-)oracle $\mathbf{o}$ is a family of quantum circuits: $\mathbf{o} = \{\mathbf{o}_i\}_{i \in \mathbb{N}}$, where $\mathbf{o} \in (\mathbf{g}, \mathbf{e}, \mathbf{d}, \mathbf{w}, \mathbf{u})$. In the IND-CPA game with security parameter n, we assume the adversary *only* queries oracle circuits $\mathbf{o}_n$. As the adversary's input is independent of any suboracle $\mathbf{o}_i$, $i \neq n$ it does not change the

$\underline{\Gamma^{0\text{-}2}}$

$b \leftarrow\$ \{0,1\}; \mathsf{sk}^* \leftarrow\$ \{0,1\}^n$
$\mathsf{pk}^* \leftarrow \mathbf{g}(\mathsf{sk}^*)$
$r^* \leftarrow\$ \{0,1\}^n$
$c^* \leftarrow \mathbf{e}(\mathsf{sk}^*, b, r^*)$
$\forall \mathsf{pk} \in \{0,1\}^{3n}, i \in \{0,1\}^n :$
  $r_{i,\mathsf{pk}} \leftarrow\$ \{f : \{0,1\}^n \mapsto \{0,1\}^n\}$
$b' \leftarrow\$ \mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d},\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*)$ // Γ^0
$b' \leftarrow\$ \mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*)$ // Γ^1
$b' \leftarrow\$ \mathcal{B}^{\mathbf{g},\mathbf{e}',\mathbf{d}',\mathbf{w}',\mathbf{u}}(\mathsf{pk}^*, c^*)$ // Γ^2
return $b' = b$

$\underline{\mathbf{d}'(\mathsf{sk}, c)}$

if $\mathsf{sk} = \mathsf{sk}^*$:
  return $\perp$
return $\mathbf{d}(\mathsf{sk}, c)$

$\underline{\mathbf{w}'(\mathsf{pk}, i)}$

if $\exists \mathsf{sk}$ s.t. $g(\mathsf{sk}) = \mathsf{pk}$:
  $r \leftarrow (\mathbf{e}'(\mathsf{pk}, \mathsf{sk}_1, r_{1,\mathsf{pk}}(i)), \ldots,$
    $\mathbf{e}'(\mathsf{pk}, \mathsf{sk}_n, r_{n,\mathsf{pk}}(i)))$
  return r
return $\perp$

$\underline{\mathbf{e}'(\mathsf{pk}, b, r)}$

if $\mathsf{pk} = \mathsf{pk}^*$ and $r = r^*$:
  return $\perp$
return $\mathbf{e}(\mathsf{pk}, b, r)$

Figure 4: Games Γ^0-Γ^2 for the proof of Thm 2.

distribution of the output. For the sake of simplicity, we write $\mathbf{o}$ for $\mathbf{o}_n$.

$\mathbf{\Gamma^1}$: We modify the $\mathbf{d}$ oracle into an identical oracle $\mathbf{d}'$ except that $\mathbf{d}'(\mathsf{sk}^*, \cdot) = \perp$, where $\cdot$ denotes any value in $\{0,1\}^{3n}$ and sk^* is the challenge secret key (i.e. $\mathbf{g}(\mathsf{sk}^*) = \mathsf{pk}^*$). That is, the $\mathbf{d}'$ oracle does not reply to decryption queries that could help the adversary decrypt the challenge ciphertext c^*. By Lemma 1, we have

$$\left|\Pr[\mathcal{A}^{\mathbf{g},\mathbf{e},\mathbf{d},\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b] - \Pr[\mathcal{A}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b]\right| \leq 2q\sqrt{\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow \mathsf{sk}^*]}$$

where $\mathcal{B}$ runs $\mathcal{A}$ until some random quantum query q_i, measures the input register and outputs the first n bits of the result. Now we prove the following lemma.

Lemma 5. $\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow \mathsf{sk}^*] = \mathsf{negl}(n)$.

Proof. We proceed by building a sequence of hybrid games where the oracle $\mathbf{w}$ is modified. We first recall that

$$\mathbf{w}(\mathsf{pk}, i) := (\mathbf{e}(\mathbf{g}(\mathsf{sk}), \mathsf{sk}_1, r_{1,i,\mathsf{pk}}), \ldots, \mathbf{e}(\mathbf{g}(\mathsf{sk}), \mathsf{sk}_n, r_{n,i,\mathsf{pk}}))$$

where the values $r_{k,i,\mathsf{pk}}$ are sampled at random and pk is s.t. $\mathbf{g}(\mathsf{sk}) = \mathsf{pk}$. Equivalently, we can write

$$\mathbf{w}(\mathsf{pk}, i) := (\mathbf{e}(\mathbf{g}(\mathsf{sk}), \mathsf{sk}_1, r_{1,\mathsf{pk}}(i)), \ldots, \mathbf{e}(\mathbf{g}(\mathsf{sk}), \mathsf{sk}_n, r_{n,\mathsf{pk}}(i)))$$

where $r_{k,\mathsf{pk}} : \{0,1\}^n \mapsto \{0,1\}^n$ are random functions.

$\underline{\mathbf{w}^1}$: Let $e_b(\cdot) := \mathbf{e}(\mathsf{pk}^*, b, \cdot)$. We modify $\mathbf{w}$ into an oracle $\mathbf{w}^1$ s.t.

$$\mathbf{w}^1(\mathsf{pk}, i) = \begin{cases} \mathbf{w}(\mathsf{pk}, i), & \text{if } \mathsf{pk} \neq \mathsf{pk}^* \\ (e_0(r_{1,\mathsf{pk}}(i)), e_{\mathsf{sk}_2^*}(r_{2,\mathsf{pk}}(i)), \ldots, e_{\mathsf{sk}_n^*}(r_{n,\mathsf{pk}}(i))), & \text{if } \mathsf{pk} = \mathsf{pk}^* \end{cases}$$

In other words, when pk^* is queried, the encryption of the first bit of sk^* is replaced by the encryption of a zero. All other values returned are the same as in the original $\mathbf{w}$ oracle. We now wish to upper bound

$$\begin{aligned}&\left|\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]-\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^1,\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]\right|\\&=\frac{1}{2}\Big|\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*|\mathsf{sk}_1^*=1]\\&-\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^1,\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*|\mathsf{sk}_1^*=1]\Big| \qquad (1)\end{aligned}$$

where the equality follows from the fact that $\mathbf{w}$ is identically distributed to $\mathbf{w}^1$ if $\mathsf{sk}_1^*=0$. We show that for any adversary $\mathcal{B}$, one can construct a IMG adversary $\mathcal{C}$ s.t. Eq. 1 is upper bounded by the advantage of $\mathcal{C}$. We show the reduction in Figure 5.

First, we see $\mathcal{C}$ simulates perfectly the oracles for queries independent of $(\mathsf{sk}^*,\mathsf{pk}^*)$. Indeed, $\mathcal{C}$ samples a valid function g and random injective functions $e(g(\mathsf{sk}),\cdot,\cdot)$ for each $\mathsf{sk}\neq\mathsf{sk}^*$. Then, $\mathcal{C}$ can use its knowledge of these functions to reply to any query $e'(\mathsf{pk},\cdot,\cdot)$, $d'(\mathsf{sk},\cdot)$, $w'(\mathsf{pk},\cdot)$ or $u'(\mathsf{pk},\cdot)$ with $\mathsf{pk}\neq\mathsf{pk}^*,\mathsf{sk}\neq\mathsf{sk}^*$ as in the original game played by $\mathcal{B}$.

Then, $\mathcal{C}$ sets the encryption oracle for pk^* as

$$e'(\mathsf{pk}^*,b,r)=\begin{cases}e_0(r), & \text{if } b=0\\ e_1(r) & \text{if } b=1\end{cases}$$

where e_0,e_1 are $\mathcal{C}$'s own oracles. As e_0,e_1 are random one-to-one functions s.t. their image do not intersect, $e'(\mathsf{pk}^*,\cdot,\cdot)$ is also a random one-to-one function $\{0,1\}^{n+1}\mapsto\{0,1\}^{3n}$. Therefore, e' simulates perfectly $\mathbf{e}$. Then, d' simulates perfectly $\mathbf{d}'$ as $\perp$ is returned if it is queried on $(\mathsf{sk}^*,\cdot)$. Similarly, u' perfectly simulates $\mathbf{u}$ by using $\mathcal{C}$'s own u oracle to reply to queries of the form $u'(\mathsf{pk}^*,\cdot)$. Finally, $w'(\mathsf{pk}^*,\cdot)$ perfectly simulates $\mathbf{w}$ when $w_b:=e_1(r(\cdot))$ and perfectly simulates $\mathbf{w}^1$ when $w_b:=e_0(r(\cdot))$, where r is a random function. Indeed, when $\mathcal{C}$ plays the IMG game with $b=1$, on a query $w'(\mathsf{pk}^*,\cdot)$ made by $\mathcal{B}$, $\mathcal{C}$ outputs a ciphertext with the first component set to $e_1(r(\cdot))=e_{\mathsf{sk}_0^*}(r(\cdot))$ (i.e. the "encryption" of the first bit of sk^*, which is equal to 1). Similarly, when $\mathcal{C}$ plays the IMG game with $b=0$, the returned ciphertext has a first component set to $e_0(r(\cdot))$, as in the $\mathbf{w}^1$ oracle. Hence, $\mathcal{C}$ playing the IMG game with bit $b=1$ (resp. $b=0$) perfectly simulates $\mathcal{B}$'s view with oracle $\mathbf{w}$ (resp. $\mathbf{w}^1$) and we have

$$\begin{aligned}&\left|\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]-\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^1,\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]\right|\\&=\frac{1}{2}\Big|\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*|\mathsf{sk}_1^*=1]\\&-\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^1,\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*|\mathsf{sk}_1^*=1]\Big|\\&=|\Pr[\mathcal{C}^{e_0,e_1,w_1,u}\Rightarrow 1]-\Pr[\mathcal{C}^{e_0,e_1,w_0,u}\Rightarrow 1]|=\mathsf{negl}(n)\end{aligned}$$

where the last equality follows from Corollary 1.

$\underline{\mathbf{w}^j}$: We successively modify the oracle $\mathbf{w}^1$ into an oracle $\mathbf{w}^j, j\in[n]$ s.t. on a query $(\mathsf{pk}^*,\cdot)$, the i-th first components of the resulting ciphertexts are encryption of a 0 instead of the i-th bit of the challenge secret key. Formally, we have

$$\mathbf{w}^j(\mathsf{pk},i)=\begin{cases}\mathbf{w}(\mathsf{pk},i), & \text{if } \mathsf{pk}\neq\mathsf{pk}^*\\ (\ldots,e_0(r_{j,\mathsf{pk}}(i)),e_{\mathsf{sk}_{j+1}^*}(r_{j+1,\mathsf{pk}}(i))),\ldots), & \text{if } \mathsf{pk}=\mathsf{pk}^*\end{cases}.$$

By a similar reduction to the IMG problem as before, we have for all $j\in\{1,\ldots,n-1\}$

$$\begin{aligned}&\left|\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^j,\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]-\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^{j+1},\mathbf{u}}(\mathsf{pk}^*,c^*)\Rightarrow\mathsf{sk}^*]\right|\\&=\mathsf{negl}(n)\ .\end{aligned}$$

$\underline{\mathbf{w}^n}$: Now, $\mathbf{w}^n(\mathsf{pk}^*, \cdot)$ returns a vector of ciphertexts encrypting 0 which means we do not use sk^* in $\mathbf{w}^n$ anymore. In order to conclude the proof of the lemma, we wish to show that

$$\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^n,\mathbf{u}}(g(\mathsf{sk}^*), c^*) \Rightarrow \mathsf{sk}^*] = \mathsf{negl}(n) \ .$$

One can see that the oracles $(\mathbf{g}, \mathbf{e}, \mathbf{d}', \mathbf{w}^n, \mathbf{u})$ never invert pk^* or use the secret key sk^* anymore. The only exception is the decryption oracle that returns $\perp$ whenever sk^* is equal to the queried sk. However, this condition can be checked by verifying whether $g(\mathsf{sk}) = \mathsf{pk}^*$, as g is one-to-one. Hence, we are going to show that if $\mathcal{B}$ outputs sk^*, one can build an adversary $\mathcal{D}$ that inverts g on a random image, having access to a partial inverse oracle. We show the adversary in Figure 6. As in Lemma 4, $\mathcal{D}^{g, g^{-1}_{\neq \mathsf{sk}^*}}$ receives $g(\mathsf{sk}^*)$, where g is a random injective function, sk^* is sampled at random, and the goal is to recover sk^*. Note that sk^* and $\mathsf{pk}^* = g(\mathsf{sk}^*)$ are distributed as in $\mathcal{B}$'s game. Then, $\mathcal{D}$ generates a challenge ciphertext c^* using pk^* and runs $\mathcal{B}$ while simulating the oracles $(\mathbf{g}, \mathbf{e}, \mathbf{d}', \mathbf{w}^n, \mathbf{u})$ as follows.

- $g(\mathsf{sk})$: $\mathcal{D}$ uses its own g oracle to reply to $\mathcal{B}$'s queries to $\mathbf{g}$. As they are similarly distributed and $\mathsf{pk}^* = g(\mathsf{sk})^*$, the simulation is perfect.
- $e'(\mathsf{pk}, b, r)$: It simply returns the evaluation of $e(\mathsf{pk}, b, r)$, where $e(\mathsf{pk}, \cdot, \cdot)$ is a random one-to-one function sampled by $\mathcal{D}$. This simulates perfectly $\mathbf{e}$.
- $d'(\mathsf{sk}, c)$: It returns the decryption of c or $\perp$ if $\mathsf{sk} = \mathsf{sk}^*$, as in the oracle $\mathbf{d}'$. Note that $\mathcal{D}$ uses its own oracle g to check whether $g(\mathsf{sk}) = \mathsf{pk}^*$.
- $w'(\mathsf{pk}, i)$: It simulates $\mathbf{w}^n$ perfectly. Indeed, if $\mathsf{pk} = \mathsf{pk}^*$ it returns a vector of ciphertexts encrypting 0. Otherwise, $\mathcal{D}$ uses its own $g^{-1}_{\neq \mathsf{sk}^*}$ to invert pk and encrypts the bits of the corresponding secret key.
- $u(\mathsf{pk}, c)$: It simulates perfectly $\mathbf{u}$ as $\mathcal{D}$ uses its knowledge of $e(\mathsf{pk}, \cdot, \cdot)$ to check whether c is a valid image.

Note that while the simulated oracles are described in a classical way, $\mathcal{D}$ implements them as quantum accessible oracles. This can be done with a polynomial number of quantum queries to its own oracles, as described before. Finally, we get

$$\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w}^n,\mathbf{u}}(g(\mathsf{sk}^*), c^*) \Rightarrow \mathsf{sk}^*] = \Pr[\mathcal{D}^{g, g^{-1}_{\neq \mathsf{sk}^*}}(g(\mathsf{sk}^*)) \Rightarrow \mathsf{sk}^*] = \mathsf{negl}(n)$$

where the last equality follows from Lemma 4. Collecting the probabilities as in a standard hybrid argument holds the proof of Lemma 5. □

$\mathbf{\Gamma^2}$: We recall that Γ^1 is the IND-CPA game except the oracle $\mathbf{d}$ has been modified into an oracle $\mathbf{d}'$ that returns $\perp$ on a query $(\mathsf{sk}^*, \cdot)$. Now, we modify Γ^1 into a game Γ^2 by building another "encryption" oracle $\mathbf{e}'$ that returns $\perp$ whenever queried on (pk^*, b, r^*) for any bit b, where r^* is the randomness used to compute the challenge ciphertext (i.e. $c^* = \mathbf{e}(\mathsf{pk}^*, b, r^*)$). Formally,

$$\mathbf{e}'(\mathsf{pk}, b, r) = \begin{cases} \perp, & \text{if } \mathsf{pk} = \mathsf{pk}^* \wedge r = r^* \\ \mathbf{e}(\mathsf{pk}, b, r), & \text{otherwise} \end{cases} \ .$$

In addition, we modify $\mathbf{w}$ into a $\mathbf{w}'$ oracle s.t. it queries $\mathbf{e}'$ instead of $\mathbf{e}$. Note that as $\mathbf{w}$ (resp. $\mathbf{w}'$) encrypts n bits in parallel, one quantum query to $\mathbf{w}$ (resp. $\mathbf{w}'$) can be computed with n quantum queries to $\mathbf{e}$ (resp. $\mathbf{e}'$). Thus, in total, there are at most $q + qn$ queries made to $\mathbf{e}$ or $\mathbf{e}'$, where q is the number of queries made by $\mathcal{A}$. Then, Γ^2 is the same as Γ^1

$\mathcal{C}^{\mathcal{B},e_0,e_1,w_b,u}$

$b' \leftarrow\$ \{0,1\}; \mathsf{sk}^* \leftarrow\$ \{0,1\}^n$
$\mathsf{sk}_0^* \leftarrow 1$
sample $g \leftarrow\$ \mathsf{Inj}_{n,3n}$
$\mathsf{pk}^* \leftarrow g(\mathsf{sk}^*)$
$\forall \mathsf{pk} \in \{0,1\}^{3n}$ s.t. $\mathsf{pk} \neq \mathsf{pk}^*$:
  sample $e(\mathsf{pk},\cdot,\cdot) \leftarrow\$ \mathsf{Inj}_{n+1,3n}$
$\forall \mathsf{pk} \in \{0,1\}^{3n}, i \in [n]$:
  $r_{i,\mathsf{pk}} \leftarrow\$ \{f : \{0,1\}^n \mapsto \{0,1\}^n\}$
$r^* \leftarrow\$ \{0,1\}^n; c^* \leftarrow e_{b'}(r^*)$
run $\mathsf{sk}' \leftarrow\$ \mathcal{B}^{g,e',d',w',u'}(\mathsf{pk}^*, c^*)$
return $1_{\mathsf{sk}'=\mathsf{sk}^*}$

$e'(\mathsf{pk}, b, r)$

if $\mathsf{pk} = \mathsf{pk}^*$:
  return $e_b(r)$
return $e(\mathsf{pk}, b, r)$

$d'(\mathsf{sk}, c)$

if $\mathsf{sk} = \mathsf{sk}^*$:
  return $\perp$
if $\exists (b,r)$ s.t. $e(g(\mathsf{sk}), b, r) = c$:
  return b
return $\perp$

$w'(\mathsf{pk}, i)$

if $\mathsf{pk} = \mathsf{pk}^*$:
  $r \leftarrow (w_b(i), e_{\mathsf{sk}_2^*}(r_{2,\mathsf{pk}}(i)), \ldots, e_{\mathsf{sk}_n^*}(r_{n,\mathsf{pk}}(i)))$
  return r
if $\exists \mathsf{sk}$ s.t. $g(\mathsf{sk}) = \mathsf{pk}$:
  $r \leftarrow (e(\mathsf{pk}, \mathsf{sk}_1, r_{1,\mathsf{pk}}(i)), \ldots, e(\mathsf{pk}, \mathsf{sk}_n, r_{n,\mathsf{pk}}(i)))$
  return r
return $\perp$

$u'(\mathsf{pk}, c)$

if $\mathsf{pk} = \mathsf{pk}^*$:
  return $u(c)$
if $\exists (b,r)$ s.t. $e(\mathsf{pk}, b, r) = c$:
  return $\top$
return $\perp$

Figure 5: $\mathcal{C}$ adversary.

except $\mathcal{A}$ has quantum oracle access to $\mathbf{e}'$ and $\mathbf{w}'$ instead of $\mathbf{e}$ and $\mathbf{w}$. As in the previous transition $\Gamma^0 \to \Gamma^1$, one can apply Lemma 1 to get

$$\left|\Pr[\mathcal{A}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b] - \Pr[\mathcal{A}^{\mathbf{g},\mathbf{e}',\mathbf{d}',\mathbf{w}',\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b]\right| \leq 2(q+qn)\sqrt{\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow r^*]}$$

where $(\mathsf{pk}^*, c^* = \mathbf{e}(\mathsf{pk}^*, b, r^*))$ is as in the IND-CPA game, and $\mathcal{B}$ runs $\mathcal{A}$ until some random quantum query q_i to $\mathbf{e}$ (made by $\mathcal{A}$ or $\mathbf{w}$), measures the input register and outputs the last n bits of the result. As before, we are going to upper bound the right-hand side of the equation by the probability a quantum adversary $\mathcal{F}$ inverts a random one-to-one length-tripling function with the help of a partial inverse oracle. This time, $\mathcal{F}$ will simulate queries of the form $\mathbf{e}(\mathsf{pk}^*, b, \cdot)$ using the function $e \in \mathsf{Inj}_{n,3n}$ it wants to invert. We show $\mathcal{F}$ in Figure 7. As in previous reductions, $\mathcal{F}$ samples its own functions $g, r_{\mathsf{pk},i}$ and $e(\mathsf{pk},\cdot,\cdot)$ for $\mathsf{pk} \neq \mathsf{pk}^*$. Using these, it can reply consistently to $\mathcal{B}$'s queries that do not involve sk^* or pk^*. In addition, $\mathcal{F}$ samples a random challenge bit b' that plays the role of the challenge bit of the IND-CPA game. Then, it sets

$$e'(\mathsf{pk}^*, b, r) = \begin{cases} e(r), & \text{if } b = b' \\ e_{1-b'}(r), & \text{if } b = 1-b' \end{cases}$$

where e is the function $\mathcal{F}$ wants to invert and $e_{1-b'} \in \mathsf{Inj}_{n,3n}$ is sampled by $\mathcal{F}$. Now, as both $e, e_{1-b'}$ are injective functions in $\mathsf{Inj}_{n,3n}$, the probability that $e'(\mathsf{pk}^*,\cdot,\cdot)$ is not a

$\mathcal{D}^{\mathcal{B},g,g^{-1}_{\neq \mathsf{sk}^*}}(\mathsf{pk}^* = g(\mathsf{sk}^*))$

$b' \leftarrow\!\$\ \{0,1\}$

$\forall \mathsf{pk} \in \{0,1\}^{3n}$:

  sample $e(\mathsf{pk},\cdot,\cdot) \leftarrow\!\$\ \mathsf{Inj}_{n+1,3n}$

  $\forall i \in \{0,1\}^n$:

    $r_{i,\mathsf{pk}} \leftarrow\!\$\ \{f : \{0,1\}^n \mapsto \{0,1\}^n\}$

$r^* \leftarrow\!\$\ \{0,1\}^n; c^* \leftarrow e(\mathsf{pk}^*, b', r^*)$

$\mathsf{sk}' \leftarrow\!\$\ \mathcal{B}^{g,e',d',w',u'}(\mathsf{pk}^*, c^*)$

return sk'

$e'(\mathsf{pk}, b, r)$

return $e(\mathsf{pk}, b, r)$

$d'(\mathsf{sk}, c)$

if $g(\mathsf{sk}) = \mathsf{pk}^*$:

  return $\perp$

if $\exists (b,r)$ s.t. $e(g(\mathsf{sk}), b, r) = c$:

  return b

return $\perp$

$w'(\mathsf{pk}, i)$

if $\mathsf{pk} = \mathsf{pk}^*$:

  $r \leftarrow (e(0, r_{1,\mathsf{pk}}(i)), \ldots, e(0, r_{n,\mathsf{pk}}(i)))$

  return r

$\mathsf{sk} \leftarrow g^{-1}_{\neq \mathsf{sk}^*}(\mathsf{pk})$

if $\mathsf{sk} = \perp$: **return** $\perp$

$r \leftarrow (e(\mathsf{pk}, \mathsf{sk}_1, r_{1,\mathsf{pk}}(i)), \ldots, e(\mathsf{pk}, \mathsf{sk}_n, r_{n,\mathsf{pk}}(i)))$

return r

$u'(\mathsf{pk}, c)$

if $\exists (b,r)$ s.t. $e(\mathsf{pk}, b, r) = c$:

  return $\top$

return $\perp$

Figure 6: $\mathcal{D}$ adversary.

random function from $\mathsf{Inj}_{n+1,3n}$ is $\Pr[\mathsf{coll}] = \Pr[\mathsf{Im}(e) \cap \mathsf{Im}(e_{1-b'}) \neq \emptyset] = O(\frac{1}{2^n})$. Thus, assuming coll does not occur, $e'(\mathsf{pk}^*,\cdot,\cdot)$ follows the same distribution as $\mathbf{e}$. In addition, $\mathcal{F}$ can simulate perfectly $\mathbf{d}$ and $\mathbf{w}$ using its knowledge of sk^* and its own oracles/functions. In particular, each quantum query $\mathbf{w}(\mathsf{pk}^*,\cdot)$ can be simulated with at most n quantum queries to its oracle e. Finally, queries of the form $u'(\mathsf{pk}^*, c)$ for some $c \in \{0,1\}^{3n}$ can be simulated perfectly, as:

- if $c = c^*$: $\mathcal{F}$ can return $\top$ as c^* is a valid ciphertext.
- if $c \neq c^*$: $\mathcal{F}$ can query its oracle $e^{-1}_{\neq r^*}$ to check whether c is a valid ciphertext of the form $c = e'(\mathsf{pk}^*, b', r)$, for some r. If that is not the case, $\mathcal{F}$ further checks whether $c = e'(\mathsf{pk}^*, 1-b', r)$ for some r using its knowledge of $e_{1-b'}$.
- if the two previous conditions are not fulfilled, then c is not a valid ciphertext.

Hence, if coll does not occur, $\mathcal{F}$ simulates perfectly $\mathcal{B}$'s view and we get

$$\Pr[\mathcal{B}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow r^*] \leq O\left(\frac{1}{2^n}\right) + \Pr[\mathcal{F}^{e,e^{-1}_{\neq r^*}}(e(r^*)) \Rightarrow r^*] = \mathsf{negl}(n)$$

where the last equality follows from Lemma 4. Thus,

$$\left|\Pr[\mathcal{A}^{\mathbf{g},\mathbf{e},\mathbf{d}',\mathbf{w},\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b] - \Pr[\mathcal{A}^{\mathbf{g},\mathbf{e}',\mathbf{d}',\mathbf{w}',\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b]\right| = \mathsf{negl}(n).$$

Finally, we argue that

$$\Pr[\mathcal{A}^{\mathbf{g},\mathbf{e}',\mathbf{d}',\mathbf{w}',\mathbf{u}}(\mathsf{pk}^*, c^*) \Rightarrow b] = \frac{1}{2}\ .$$

$\mathcal{F}^{\mathcal{B},e,e^{-1}_{\neq r^*}}(c^* = e(r^*))$

$b' \leftarrow\$ \{0,1\}; \mathsf{sk}^* \leftarrow\$ \{0,1\}^n$
sample $g \leftarrow\$ \mathsf{Inj}_{n,3n}$
$\mathsf{pk}^* \leftarrow g(\mathsf{sk}^*)$
$\forall \mathsf{pk} \neq \mathsf{pk}^* \in \{0,1\}^{3n}$:
  sample $e(\mathsf{pk},\cdot,\cdot) \leftarrow\$ \mathsf{Inj}_{n+1,3n}$
$\forall \mathsf{pk} \in \{0,1\}^{3n}, \forall i \in \{0,1\}^n$:
  $r_{i,\mathsf{pk}} \leftarrow\$ \{f : \{0,1\}^n \mapsto \{0,1\}^n\}$
$e_{1-b'} \leftarrow\$ \mathsf{Inj}_{n,3n}$
$\mathsf{sk}' \leftarrow\$ \mathcal{B}^{g,e',d',w',u'}(\mathsf{pk}^*, c^*)$
return sk'

$e'(\mathsf{pk}, b, r)$

if $\mathsf{pk} = \mathsf{pk}^*$:
  if $b = b'$: **return** $e(r)$
  else : **return** $e_{1-b}(r)$
return $e(\mathsf{pk}, b, r)$

$d'(\mathsf{sk}, c)$

if $\mathsf{sk} = \mathsf{sk}^*$:
  return $\perp$
if $\exists (b,r)$ s.t. $e(g(\mathsf{sk}), b, r) = c$:
  return b
return $\perp$

$w'(\mathsf{pk}, i)$

if $\exists \mathsf{sk}$ s.t. $g(\mathsf{sk}) = \mathsf{pk}$:
  $r \leftarrow (e'(\mathsf{pk}, \mathsf{sk}_1, r_{1,\mathsf{pk}}(i)), \ldots, e'(\mathsf{pk}, \mathsf{sk}_n, r_{n,\mathsf{pk}}(i)))$
  return r
return $\perp$

$u'(\mathsf{pk}, c)$

if $\mathsf{pk} = \mathsf{pk}^*$:
  if $c = c^*$: **return** $\top$
  if $e^{-1}_{\neq r^*}(c) \neq \perp$: **return** $\top$
  if $\exists r$ s.t. $e_{1-b'}(r) = c$: **return** $\top$
  return $\perp$
if $\exists (b,r)$ s.t. $e(\mathsf{pk}, b, r) = c$:
  return $\top$
return $\perp$

Figure 7: $\mathcal{F}$ adversary.

Indeed, we recall that the challenge ciphertext is $c^* = \mathbf{e}(\mathsf{pk}^*, b, r^*)$, where $\mathbf{e}(\mathsf{pk}^*, \cdot, \cdot)$ is a random injective function and b is a random bit. Then, the decryption oracle $\mathbf{d}'$ is useless as $\mathbf{d}'(\mathsf{sk}^*, \cdot)$ returns $\perp$, thus $\mathcal{A}$ cannot invert c^*. In addition, no oracle (i.e. $\mathbf{e}'$ or $\mathbf{w}'$) ever returns $\mathbf{e}(\mathsf{pk}^*, b, r^*)$ for any bit b (i.e. $\perp$ is returned in both cases). Finally, $\mathbf{u}(\mathsf{pk}^*, \mathbf{e}(\mathsf{pk}^*, b, r^*))$ returns $\top$ for both $b = 0$ and $b = 1$. Hence, given $\mathcal{A}$'s view, $\Pr[c^* = \mathbf{e}(\mathsf{pk}^*, 0, r^*)] = \Pr[c^* = \mathbf{e}(\mathsf{pk}^*, 1, r^*)]$ and $\mathcal{A}$ cannot distinguish. Therefore, $\Pr[\Gamma^2 \Rightarrow 1] = \frac{1}{2}$ and collecting the probabilities holds the result. □

Corollary 2. *Let* $\mathsf{PKE_q}^{\mathcal{O}} = (g^{\mathbf{g}}, \mathbf{e}, \mathbf{d})$ *be a PKE relative to* $\mathcal{O}$*, where* $g^{\mathbf{g}}(s)$ *sets* $\mathsf{sk} \leftarrow s$ *and returns* $(\mathsf{sk}, \mathbf{g}(\mathsf{sk}))$*. Then, we have*

$$\Pr_{\mathcal{O} \leftarrow\$ \Psi}[\forall \mathsf{EFF_q}\ \mathcal{A} : \mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A}^{\mathcal{O}}, \mathsf{PKE_q}^{\mathcal{O}}} = \mathsf{negl}(n)] = 1$$

where $\mathsf{EFF_q}$ *stands for "efficient quantum". In other words, for measure* 1 *of oracles,* $\mathsf{PKE_q}^{\mathcal{O}}$ *is* IND-CPA *secure.*

Proof. This follows from a now standard trick in impossibility results based on the Borel-Cantelli lemma, Markov inequalities, and a counting argument (e.g. see Lemma 2 and 5 by Buldas et al. [BN13]). Note, however, that for the proof to work, the set of efficient quantum adversaries must be countable. This is the case here, as we consider *uniform* quantum circuits, which are countable (as they can be generated by deterministic Turing Machines). □

6 Non-existence of IND-CCA PKE relative to $\mathcal{O}$

We first recall which type of constructions we will rule out, namely *shielding constructions*.

Definition 15 (Shielding construction)**.** A valid PKE construction relative to $\mathbf{O} = (\mathbf{g}, \mathbf{e}, \mathbf{d})$ $\mathsf{PKE}^{\mathbf{O}} = (\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$ is *shielding* iff the decryption function $\mathbf{D}$ never queries the oracles $\mathbf{e}$. In other words, we can write $\mathsf{PKE}^{\mathcal{O}} = (\mathbf{G}^{\mathbf{g},\mathbf{e},\mathbf{d}}, \mathbf{E}^{\mathbf{g},\mathbf{e},\mathbf{d}}, \mathbf{D}^{\mathbf{g},\mathbf{d}})$.

Informally, the decryption function of a PKE resulting from a shielding transform never queries the encryption function of the underlying PKEs.

Now, in order to complete the proof of the impossibility result, we need to show that any shielding black-box construction

$$\mathsf{PKE}^{\mathcal{O}} = (\mathbf{G}^{\mathbf{g},\mathbf{e},\mathbf{d}}, \mathbf{E}^{\mathbf{g},\mathbf{e},\mathbf{d}}, \mathbf{D}^{\mathbf{g},\mathbf{d}})$$

is not IND-CCA secure. We can simply reuse Gertner et al.'s result [GMM07], as they showed there exists a classical IND-CCA adversary that breaks any shielding PKE construction. This implies that there is such a quantum adversary as well.

This is stated in the following theorem.

Theorem 3 (Theorem 2 [GMM07])**.** *Let* $\mathsf{PKE} = (\mathbf{G}, \mathbf{E}, \mathbf{D})$ *be any shielding construction. Then, there exists a (non-efficient) adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *making a polynomial number of queries to* $(\mathbf{O}, \mathbf{R})$ *s.t.*

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathcal{A}^{\mathbf{O},\mathbf{R}},\mathsf{PKE}^{\mathbf{O}}} \geq 1 - \frac{1}{n}$$

where the probability of the advantage is taken over the randomness of the game and of the adversary, and the sampling of $(\mathbf{O}, \mathbf{R}) \hookleftarrow \Psi$*, where* Ψ *is defined as in Definition 10.*

Proof sketch. We recall the idea of the proof here.

1. In the first step, the public keys $\mathbf{g}(\mathsf{sk})$ for some sk's embedded into the public key PK (which is output by $\mathbf{G}^{\mathbf{O}}$) are collected. In order to do this, the adversary executes $\mathbf{E}^{\mathbf{O}}(PK, M, R)$ for many different M and R, collecting all pk in queries $\mathbf{e}(\mathsf{pk}, \cdot, \cdot)$ made by $\mathbf{E}$. Obviously not all pk's possibly embedded in PK are recovered as some could never be used, but the useful ones (most likely) are. Indeed, the secret keys sk's that are going to be used in decryption should correspond to the public keys used in encryption. Thus, the main goal of the next steps will be to invert the public keys pk's that have been collected in this part.

2. In this step, the public keys corresponding to the IND-CPA scheme are inverted. This is the only part where the decryption oracle provided to the classical adversary in the IND-CCA game is used. The approximate idea is the following. Many ciphertexts $C = \mathbf{E}^{\mathbf{O}}(PK, M, R)$ for a random bit M and coins R are generated. Then, the process is repeated but in each encryption, some query $\mathbf{e}(\mathsf{pk}, b, r)$ (for some b and r) made by $\mathbf{E}$ is replaced by some value $\mathbf{e}(\mathsf{pk}, \mathsf{sk}_i, r')$ obtained through the $\mathbf{w}$ oracle, where $\mathsf{pk} = \mathbf{g}(\mathsf{sk})$. Let C' be such a modified ciphertext and $C = \mathbf{E}^{\mathbf{O}}(PK, M, R)$ the original one. Then, C' is queried to the decryption oracle to get $M' = \mathbf{D}^{\mathbf{O}}(SK, C')$. We first observe that if $\mathsf{sk}_i = b$, then M' should be equal to M. Indeed, we replaced $c := \mathbf{e}(\mathsf{pk}, b, r)$ by $c' := \mathbf{e}(\mathsf{pk}, b, r')$, but since $\mathbf{D}$ cannot query $\mathbf{e}$, it cannot distinguish c from c'. Now we can distinguish two cases:

   - $M \neq M'$: By the previous observation, it means that (most likely) $b \neq \mathsf{sk}_i$ and thus $\mathsf{sk}_i = 1 - b$ can be recovered, as b is known.

- $M = M'$: Either $\mathsf{sk}_i = b$ or the ciphertext corresponding to the modified query (or the decryption of the ciphertext) does not impact the decryption result. However, by repeating many times the experiment with different (M, R), it is possible to distinguish both cases with high probability and one can recover the corresponding bit of the secret key sk.

Note that if no ciphertext of the form $\mathbf{e}(\mathsf{pk}, \cdot, \cdot)$ ever impacts the decryption, the secret-key sk s.t. $\mathbf{g}(\mathsf{sk}) = \mathsf{pk}$ will not be recovered using this technique. However, it also means that recovering such secret-key is not important as it is not used in decryption. Hence, after this step, all useful sk's should be recovered with high probability.

3. In the last step, using the knowledge of the secret-keys recovered and of the queries made throughout the different experiments, the adversary builds a key SK' and simulates the decryption algorithm $\mathbf{D}^{\mathbf{O}}$ using its own version $\hat{\mathbf{D}}^{\hat{\mathbf{O}}}$. Then, with high probability we will have $\hat{\mathbf{D}}^{\hat{\mathbf{O}}}(SK', C^*) = M^*$, where C^* is the challenge ciphertext and M^* the challenge bit of the IND-CCA game (remember we consider 1-bit PKEs). This step is the only non-efficient one, as the adversary needs to sample an oracle $\hat{\mathbf{O}}$ consistent with the values observed in the previous step.

□

Corollary 3. *If* P = NP*, for measure one of oracles* $(\mathbf{O}, \mathbf{R})$*, there exists an* efficient *adversary* $\mathcal{A}$ *that breaks the IND-CCA security of every shielding construction* $\mathsf{PKE}^{\mathbf{O}} = (\mathbf{G}^{\mathbf{O}}, \mathbf{E}^{\mathbf{O}}, \mathbf{D}^{\mathbf{O}})$*.*

Proof. This follows from Theorem 3 and the fact that the adversary defined in the proof is efficient if P = NP. Indeed, the adversary is efficient except in the last step, where it samples an oracle that must be consistent with the queries seen. Sampling such an oracle is equivalent to sampling an NP witness, which can be done efficiently if P = NP. More details can be found in the original proof [GMM07]. □

It follows that that disproving the previous result would imply proving P ≠ NP. However, we note that the assumption P=NP is not necessary. One can also embed a PSPACE oracle in the breaking oracle $\mathbf{R}$, then the proof holds as $\mathrm{P}^{\mathrm{PSPACE}} = \mathrm{NP}^{\mathrm{PSPACE}}$.

The main result of the paper then follows.

Theorem 4. *There is no post-quantum shielding black-box construction of IND-CCA PKE from IND-CPA PKEs.*

Proof. From Corollaries 2 and 3 we know that for measure one of oracles $(\mathbf{O}, \mathbf{R})$, IND-CPA PKEs exist but IND-CCA PKEs do not. Thus, there exists a tuple of oracle $(\mathbf{O}, \mathbf{R})$ s.t. IND-CPA PKEs exist but IND-CCA PKEs do not. Hence, the conditions for Lemma 2 to hold are fulfilled and that concludes the proof. □

Acknowledgement. LHD is supported by a grant (project no. 192364) of the Swiss National Science Foundation (SNSF).

References

[AS16] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. *SIAM J. Comput.*, 45(6):2117–2176, 2016. doi:10.1137/15M1034064.

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315. Springer, 2013. doi:10.1007/978-3-642-42033-7_16.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011. doi:10.1007/978-3-642-25385-0_3.

[BN13] Ahto Buldas and Margus Niitsoo. Black-box separations and their adaptability to the non-uniform model. In Colin Boyd and Leonie Simpson, editors, *Information Security and Privacy - 18th Australasian Conference, ACISP 2013, Brisbane, Australia, July 1-3, 2013. Proceedings*, volume 7959 of *Lecture Notes in Computer Science*, pages 152–167. Springer, 2013. doi:10.1007/978-3-642-39059-3_11.

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379. Springer, 2013. doi:10.1007/978-3-642-40084-1_21.

[CX21] Shujiao Cao and Rui Xue. Being a permutation is also orthogonal to one-wayness in quantum world: Impossibilities of quantum one-way permutations from one-wayness primitives. *Theor. Comput. Sci.*, 855:16–42, 2021. URL: https://doi.org/10.1016/j.tcs.2020.11.013, doi:10.1016/J.TCS.2020.11.013.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554. Springer, 1999. doi:10.1007/3-540-48405-1_34.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptol.*, 26(1):80–101, 2013. URL: https://doi.org/10.1007/s00145-011-9114-1, doi:10.1007/S00145-011-9114-1.

[GMM07] Yael Gertner, Tal Malkin, and Steven A. Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455. Springer, 2007. doi:10.1007/978-3-540-70936-7_24.

[HR04] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2004. doi:10.1007/978-3-540-28628-8_6.

[HY20] Akinori Hosoyamada and Takashi Yamakawa. Finding collisions in a quantum world: Quantum black-box separation of collision-resistance and one-wayness. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2020. doi:10.1007/978-3-030-64837-4_1.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 44–61. ACM, 1989. doi:10.1145/73007.73012.

[MS09] Steven A. Myers and Abhi Shelat. Bit encryption is complete. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 607–616. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.65.

[NO02] Harumichi Nishimura and Masanao Ozawa. Computational complexity of uniform quantum circuit families and quantum turing machines. *Theor. Comput. Sci.*, 276(1-2):147–181, 2002. doi:10.1016/S0304-3975(01)00111-6.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004. doi:10.1007/978-3-540-24638-1_1.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998. URL: https://doi.org/10.1007/BFb0054137, doi:10.1007/BFB0054137.

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, 2015. doi:10.1145/2817206.

[Vaz98] Umesh Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1743):1759–1768, 1998. doi:10.1098/rsta.1998.0247.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Inf. Comput.*, 15(7&8):557–567, 2015. doi:10.26421/QIC15.7-8-2.

A Proof of Lemma 1

Proof. We first recall a generalized version of the *Swapping Lemma* [Vaz98], proven by Hosoyamada et al. [HY20].

Lemma 6 (Generalized Swapping Lemma [HY20])**.** *Let* $\Gamma = (\mathcal{O}_1, \ldots, \mathcal{O}_t)$ *and* $\Gamma' = (\mathcal{O}'_1, \ldots, \mathcal{O}'_t)$ *be sequences of fixed (i.e. not randomized) quantum-accessible oracles. In addition, for any pair of quantum-accessible oracles* $\mathcal{O}, \mathcal{O}'$*, we define* $\Delta(\mathcal{O}, \mathcal{O}') := \{x : \mathcal{O}(x) \neq \mathcal{O}'(x)\}$*. Then, for any oracle-aided quantum algorithm* $\mathcal{A}$ *and any input* $x \in \{0,1\}^n$

$$\left|\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y] - \Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right| \leq 2\sum_{i=1}^{t} \sqrt{q \sum_{z \in \Delta(\mathcal{O}_i, \mathcal{O}'_i)} \mu_z^{\mathcal{A},\mathcal{O}_i}(x)}$$

for any output y*.*

We first note that sampling Γ' is the same as sampling (Γ, z, x) and then the set of differing outputs $D \leftarrow\!\!\$\ \mathcal{D}_{x,z,\Gamma}$. Hence, the left-hand side of the equation can be written as

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y]\right] - \mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y] - \Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right]\right| \\
&\leq \mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\left|\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y] - \Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right|\right]
\end{aligned}$$

where we used the linearity of expectation and the inequality $|\mathbb{E}[X]| \leq \mathbb{E}[|X|]$. Now, Γ, Γ', x are fixed in the probabilities above (i.e. we conditioned on Γ, D, z and x). Thus, we can apply Lemma 6 to get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\left|\Pr[\mathcal{A}^{\Gamma}(x) \Rightarrow y] - \Pr[\mathcal{A}^{\Gamma'}(x) \Rightarrow y]\right|\right] \\
&\leq 2\sqrt{q}\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\sqrt{\sum_{(z,z') \in \Delta(\mathcal{O}_i, \mathcal{O}'_i)} \mu_{(z,z')}^{\mathcal{A},\mathcal{O}_i}(x)}\right] \\
&\leq 2\sqrt{q\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\sum_{(z,z') \in \Delta(\mathcal{O}_i, \mathcal{O}'_i)} \mu_{(z,z')}^{\mathcal{A},\mathcal{O}_i}(x)\right]} \\
&= 2\sqrt{q\mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\sum_{j=1}^{q} \mu_{(z,\cdot),j}^{\mathcal{A},\mathcal{O}_i}(x)\right]}
\end{aligned}$$

where we used the inequality $\mathbb{E}[\sqrt{X}] \leq \sqrt{\mathbb{E}[X]}$ and we set $\mu_{(z,\cdot),j}^{\mathcal{A},\mathcal{O}_i}(x) = \sum_{(z,z') \in \Delta(\mathcal{O}_i, \mathcal{O}'_i)} \mu_{(z,z'),j}^{\mathcal{A},\mathcal{O}_i}(x)$ for some query j. Now, let Q be the query number sampled uniformly at random by $\mathcal{B}$ and let's assume $Q = j$. Then, the probability $\mathcal{B}$ outputs z is the probability that the result of measuring the j-th query made by $\mathcal{A}$ is of the form (z, z') for some z'. By the definition of query magnitude, it is at least $\mu_{(z,\cdot),j}^{\mathcal{A},\mathcal{O}_i}(x)$, thus $\Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z | Q = j] \geq \mu_{(z,\cdot),j}^{\mathcal{A},\mathcal{O}_i}(x)$. Hence,

$$\Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z] = \frac{1}{q}\sum_{j=1}^{q} \Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z | Q = j] \geq \frac{1}{q}\sum_{j=1}^{q} \mu_{(z,\cdot),j}^{\mathcal{A},\mathcal{O}_i}(x)\ .$$

Finally, we get

$$2\sqrt{q \mathop{\mathbb{E}}_{\Gamma,x,z,D}\left[\sum_{j=1}^{q} \mu^{\mathcal{A},\mathcal{O}_i}_{(z,\cdot),j}(x)\right]} \leq 2\sqrt{q^2 \mathop{\mathbb{E}}_{\Gamma,x,z,D}[\Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z]]}$$
$$= 2q\sqrt{\Pr[\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z]}$$

where the last probability is taken over the internal randomness of $\mathcal{B}$, and the randomness of the measurement, Γ, x and z. Note that we can remove the dependence over D as the event $\{\mathcal{B}^{\mathcal{A},\Gamma}(x) \Rightarrow z\}$ is fully determined by Γ, x, z and the randomness of $\mathcal{B}$. Collecting the inequalities concludes the proof. □