Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security*

Veronika Kuchta[1], Amin Sakzad[1], Damien Stehlé[2,3], Ron Steinfeld[1], and Shi-Feng Sun[1,4]

[1] Faculty of Information Technology, Monash University, Australia
[2] Univ. Lyon, EnsL, UCBL, CNRS, Inria, LIP, F-69342 Lyon Cedex 07, France.
[3] Institut Universitaire de France.
[4] Data61, CSIRO, Australia

Abstract. We introduce a new technique called 'Measure-Rewind-Measure' (MRM) to achieve tighter security proofs in the quantum random oracle model (QROM). We first apply our MRM technique to derive a new security proof for a variant of the 'double-sided' quantum One-Way to Hiding Lemma (O2H) of Bindel et al. [TCC 2019] which, for the first time, avoids the square-root advantage loss in the security proof. In particular, it bypasses a previous 'impossibility result' of Jiang, Zhang and Ma [IACR eprint 2019]. We then apply our new O2H Lemma to give a new tighter security proof for the Fujisaki-Okamoto transform for constructing a strong (IND-CCA) Key Encapsulation Mechanism (KEM) from a weak (IND-CPA) public-key encryption scheme satisfying a mild injectivity assumption.

Keywords. QROM, security proof, public-key encryption.

1 Introduction

Background. Correctly selecting secure parameters for quantum-resistant cryptosystems requires understanding both the concrete quantum cost of attacks against the underlying intractability assumption (e.g., LWE [19]), as well as the concrete quantum cost of attacks against the cryptosystem itself. Ideally, one would like a cryptosystem whose security is *tightly* related via a *security proof* (or security reduction) to the intractability of a well-studied problem, so that attacks against the cryptosystem of lower cost than those against the problem are ruled out. Such tight proofs give confidence in the concrete security of practical parameter choices based on the best known attacks against the underlying problem. Unfortunately, due to existing gaps in the understanding

* This is the full version of the EUROCRYPT 2020 conference paper.

of security proofs in the context of quantum adversaries, there are many practical post-quantum cryptosystem candidates that lack such tight security proofs.

A case in point is the Fujisaki-Okamoto (FO) CCA transform [9], which is commonly applied in the design of practical public-key cryptosystems to strengthen their security from chosen-plaintext security ($\mathsf{IND\text{-}CPA}$) to chosen-ciphertext security ($\mathsf{IND\text{-}CCA}$), assuming the random oracle model (ROM) for the underlying cryptographic hash functions. This transform and its variants [8, 21, 10, 20] are used in all public-key encryption schemes and key-establishment algorithms of the second round of the NIST PQC standardisation process [18]. Tight security proofs are known for FO variants against classical adversaries (in the classical ROM), meaning that an adversary breaking the FO-transformed scheme in time T and advantage ε can be used to break the underlying scheme in time $\approx T$ and advantage $\approx \varepsilon$. Oppositely, no such tight security proof for an all-purpose FO transform is known against *quantum* attacks in the quantum random oracle model [6]. In the QROM, the adversary is given *quantum* access to those hash functions modeled by random oracles. Note that [20, 25] described a transform from a deterministic encryption scheme that enjoys a so-called disjoint simulatability property, to an $\mathsf{IND\text{-}CCA}$ public-key encryption scheme, which is tight in the QROM. The assumptions for this tight QROM transform are more stringent than those of the all-purpose FO transform: only 2.5 out of 17 second round NIST proposals for public-key encryption schemes claim that it is applicable to them [3, 7, 4],[1] and at the cost of additional assumptions.

Although a series of works [21, 10, 11, 12, 15, 13, 5] have provided improved analyses of the FO transform, the existing QROM reductions are still not tight. The state-of-the-art reductions essentially preserve the runtime, but the advantage degradation only satisfies $\mathsf{Adv}(\mathcal{A}_{\text{CCA}}) \leq O(q^c \cdot (\mathsf{Adv}(\mathcal{B}_{\text{CPA}}))^\delta)$, where $(c, \delta) = (1/2, 1/2)$ (versus the ideal tight result $(c, \delta) = (0, 1)$ that one could hope for), where $\mathsf{Adv}(\mathcal{A}_{\text{CCA}})$ and $\mathsf{Adv}(\mathcal{B}_{\text{CPA}})$ respectively denote the distinguishing advantages of the $\mathsf{IND\text{-}CCA}$ attack against the FO-transformed scheme and $\mathsf{IND\text{-}CPA}$ attack against the original scheme, and q denotes the number of QROM queries made by the attacker $\mathcal{A}$. We note that previous techniques have mainly improved the value of c, reducing it gradually from $c = 3/2$ down to $c = 1/2$. Regarding δ, while it has been improved from $1/4$ to $1/2$, going further

[1] In the case of [4], this holds for Streamlined NTRU Prime, but not for NTRU LPRime.

towards $\delta = 1$ has seemed challenging. Recently, it has even been conjectured infeasible, based on an 'impossibility result' [14].

At the heart of these prior results has been the use of the 'One-way to Hiding' (O2H) lemma, first given in [23]. All its versions so far inherently lead to a 'square-root advantage' loss in the proofs of the FO transforms. The O2H lemma can be formulated informally as follows. A quantum distinguisher $\mathcal{A}_{\mathrm{O2H}}$ is given quantum access to an oracle O that implements either a random oracle $H : X \to Y$ or a modified random oracle $G : X \to Y$, where H and G are identical on all except a single secret point $x \in X$: we have $H(x') = G(x')$ for all $x' \neq x$ and $H(x) = y_H$ and $G(x) = y_G$ where y_H, y_G are independent uniformly chosen random strings. The distinguisher is also given $z = (z_x = \mathsf{enc}(x), z_H = y_H, z_G = y_G)$, where enc is a one-way function (a deterministic encryption scheme in the FO scenario).[2] The goal of $\mathcal{A}_{\mathrm{O2H}}$ is to distinguish whether the oracle O implements G or H, while making up to q queries to O with depth at most d (where a depth of d means that $\mathcal{A}_{\mathrm{O2H}}$ splits its queries into d bunches and all queries within each bunch are queried in *parallel*, so queries in each bunch may depend on the answer to $d-1$ previous query bunches, and the total number of queries over all d bunches is at most q). An algorithm that computes x from z_x (by breaking the one-wayness of enc), queries $O(x)$ and compares the result to z_H achieves an advantage $\mathsf{Adv}(\mathcal{A}^{O}_{\mathrm{O2H}})$ negligibly close to 1. In the case of a classical access to O, no algorithm can do better. In the quantum access case, all variants of the O2H lemma known so far suffer from a square-root advantage loss. For example, the recent [5, Lemma 5] states that $\mathsf{Adv}(\mathcal{A}^{O}_{\mathrm{O2H}}) \leq 2 \cdot \sqrt{\mathsf{Adv}(\mathcal{B}^{G,H}_{\mathrm{OW}})}$. Here $\mathcal{B}^{G,H}_{\mathrm{OW}}(z)$ is a quantum attacker against the one-wayness of enc, which is given oracle access for both G and H (these oracles can be simulated given z_x, and thus such an attacker implies an attacker against the one-wayness of enc). The one-wayness attacker $\mathcal{B}^{G,H}_{\mathrm{OW}}$ constructed in the proof of this O2H lemma (and all prior variants thereof) 'only' runs $\mathcal{A}_{\mathrm{O2H}}$ and measures its queries. In particular, it does not 'rewind' $\mathcal{A}_{\mathrm{O2H}}$ to an earlier state. Rewinding the state of an attacker to an earlier state is often considered tricky in the quantum setting, due to the fact that measurement operations are not reversible. The 'impossibility result' of [14] states that any O2H lemma based on a one-wayness attacker that runs the distinguisher only once and

[2] We use this definition of z for simplicity in this introduction. The actual formulation of most prior O2H lemmas, as well as our new one, is more general and allows z to have an arbitrary joint distribution with G, H, x, as well as allowing a set S of any number of x's on which G and H may differ, rather than just one.

Table 1. Comparison of security bounds and features of our new O2H lemma with earlier variants of the O2H lemma. The 'Bound' column shows the dependence of the upper bound on the distinguisher advantage $\mathsf{Adv}(\mathcal{A})$ in terms of the One-Wayness attacker advantage ε and $\mathcal{A}$'s oracle query depth $d \leq q$ (where q is the total number of queries). The '$|S|$' column indicates the number of points on which G and H may differ, the '$\mathcal{B}_{\mathrm{OW}}$ must know' column shows the oracles available to the one-wayness attacker, and the 'Event' column indicates the type of event used to define $\mathcal{A}$'s advantage. Here $H \setminus S$ (resp. $G \setminus S$) refers to the restriction of H (resp. G) to the complementary set of S, and 1_S refers to the indicator function of S.

O2H variant	Bound	$\vert S\vert$	$\mathcal{B}_{\mathrm{OW}}$ must know	Event
Original [23, 1]	$2d\varepsilon^{1/2}$	Arbitrary	H or G	Arbitrary
Semi-classical [1]	$2d^{1/2}\varepsilon^{1/2}$	Arbitrary	$(H \setminus S$ or $G \setminus S)$ and 1_S	Arbitrary
Double-sided [5]	$2\varepsilon^{1/2}$	One	H and G	Arbitrary
This work	$4d\varepsilon$	Arbitrary	H and G	Efficiently checkable

involves no rewinding, must incur a square-root advantage loss. Thus, it has been suggested in [5, 14] that the square-root advantage loss in the O2H lemma may be unavoidable in the quantum setting.

Contributions. We present a novel quantum O2H lemma that, for the first time, does not suffer from the square-root advantage loss in the reduction. Concretely, we obtain a security bound of the form $\mathsf{Adv}(\mathcal{A}) \leq 4 \cdot d \cdot \mathsf{Adv}(\mathcal{B}^{G,H})$, where $\mathcal{B}$ is the one-wayness attacker against the underlying one-way function enc.

To circumvent the 'impossibility result' of [14], we introduce a Measure-Rewind-Measure (MRM) proof technique, which provides a new way to extract the one-wayness secret x from the distinguisher. Rather than extracting x directly by measuring the oracle queries of the distinguisher (as in prior works), the MRM technique may also extract x from the *distinguishing measurement* of the distinguisher. The latter distinguishing measurement knowledge extraction is achieved by letting the distinguisher perform its distinguishing measurement, and then rewinding *the collapsed measured state* back to the state of the oracle query stage, to perform a second measurement and extract x. A comparison of our O2H lemma security bounds and features with earlier O2H lemma variants is provided in Table 1.

Compared to prior O2H lemmas, our variant is the first to avoid the square-root advantage loss. On the other hand, it constructs a one-wayness attacker which in general requires oracle accesses to both G

and H. Therefore, our lemma is in the same setting as the 'double-sided' O2H lemma of [5], which makes it less general than the semi-classical or original O2H lemmas. Nevertheless, it still suffices for important applications (see below). Compared to the 'double-sided' O2H lemma in [5], our variant is slightly less general in one respect and more general in another. On the one hand, the classical event distinguished by the O2H attacker $\mathcal{A}$ in [5] can be arbitrary, while we assume this event to be efficiently checkable by $\mathcal{A}$. 'Efficiently checkable' means that the distinguishing advantage in the definition of the O2H Lemma is defined as the advantage of $\mathcal{A}$ in the usual way, i.e., $\mathsf{Adv}(\mathcal{A}) = |\Pr[1 \leftarrow \mathcal{A}^G(z)] - \Pr[1 \leftarrow \mathcal{A}^H(z)]|$. This is in contrast to the more general definition used in [5], which uses the advantage $|\Pr[\mathsf{Ev} : \mathcal{A}^G(z)] - \Pr[\mathsf{Ev} : \mathcal{A}^H(z)]|$ for any classical event Ev over the view of $\mathcal{A}$. There may not exist a computationally efficient algorithm to check whether Ev has occurred. On the other hand, our O2H variant allows $|S|$ (the number of points on which G and H may differ) to be arbitrary, while in [5] it must contain a single point.

As an important application of our O2H lemma, we present the first security proof for the FO transform in the QROM which does not suffer from a 'square-root' advantage loss for non-deterministic schemes, i.e., it has the form $\mathsf{Adv}(\mathcal{A}_{\text{CCA}}) \leq O(q^c \cdot \mathsf{Adv}(\mathcal{B}_{\text{CPA}})^\delta)$, where $\delta = 1$ rather than $\delta = 1/2$ as in previous results (on the other hand, our proof currently gives a larger value of c compared to earlier works, see below). A comparison of our FO security proof bounds with earlier ones starting from $\mathsf{IND\text{-}CPA}$ non-deterministic weak schemes is provided in Table 2. The 'Security loss' column of that table shows the number of extra bits of security required for the 'weak scheme' in order to guarantee (via the security proof bound) a desired bit security of λ for the FO-transformed scheme. To obtain the 'security loss' L, we define the indistinguishability bit security of a scheme (against distinguishers that never output $\perp$, which is the class of attacks considered here) [16] as λ if the time to squared (conditional) advantage ratio T/ε^2 of any attack with time $T \leq 2^\lambda$ is $\geq 2^\lambda$.[3] We then choose the smallest bit security S_{weak} of the 'weak scheme' so that the CCA security bound for the CCA scheme implies a CCA bit security of the FO scheme to be $\geq \lambda$, and define the 'security loss' as $L := S_{weak} - \lambda$. We remark that our bit security loss estimates in Table 2 assume that the classical bit security definitions in [16] are appropriate in the quantum setting, as we are not aware of any research on bit security notions in the quantum setting. Note also that this latter assumption does not impact the security

[3] We note that [16] calls ε the 'conditional advantage' while ε^2 is referred to as the 'advantage'; we always refer to 'conditional advantage' ε as 'advantage'.

bounds we prove in this paper (which do not depend on this assumption); it only affects their interpretation in Table 2 in terms of bit security. We refer the reader to Appendix B for the security loss computation details for the entries of Table 2.

We make the following remarks about Table 2. Whereas all previous proofs for FO applied to non-deterministic $\mathsf{IND\text{-}CPA}$ weak schemes incurred at least a λ bit security loss (due to the square-root advantage loss in the CCA bound), our proof removes this λ bit overhead, and instead incurs a loss $4 \log d$ that depends only on the query depth d of the CCA distinguisher. In particular, this means that our security proof is nearly tight for low query depth attacks (i.e., when $\log d$ is much smaller than λ), its loss is less than λ bits for $\log d < \lambda/4$.The case of (relatively) low query depth attacks ruled out by our proof tends to be of high practical interest, since it corresponds, for instance, to massively parallelized attacks, which are the standard approach to deal with high computation costs in practical cryptanalyses. An additional requirement of our scheme is injectivity, but it turns out that it is commonly satisfied by many practical weak schemes, as argued in [5]. We leave a detailed investigation of injectivity of the second round PQC NIST KEM candidates [18] to future work (see [5, Appendix D] for a short discussion). We also remark that although our work and [5] need the extra injectiveness assumption, it gives a better bound than prior works for *modular* FO proofs (those that decompose into a composition of two proofs: one for the T transform and one for the U transform). The prior works in Table 2 can get the same bound overall for FO but only via a direct proof for whole FO transform (combining the T and U transforms). The reason we do not adapt prior FO proofs that do not rely on the injectiveness property is that those proofs also seem to require an O2H Lemma where the extractor works with single-sided oracles for either G *or* H, rather than the G *and* H requirement we (and [5]) have in our 'double-sided' O2H Lemma.

Techniques. To explain our MRM security proof technique, we consider the following example and explain the difficulty encountered by previous O2H proofs, and then our observations leading to our MRM technique for resolving this difficulty.

Consider the following O2H distinguisher $\mathcal{A}^O$ that makes 1 query (with depth 1) to its quantum oracle and makes a measurement on the resulting state to distinguish whether $O = H$ or $O = G$. The oracle input (first) and output (second) registers are denoted by *in* and *out*. Given $z = (\mathsf{enc}(x), H(x))$, the distinguisher $\mathcal{A}^O$ prepares in the input register *in* a superposition of the form $\sum_{x' \in X} \sqrt{p_{x'}} |x'\rangle$ and queries O to get the

Table 2. Comparison of security bounds for FO-type non-deterministic $\mathsf{IND\text{-}CPA}$ to $\mathsf{IND\text{-}CCA}$ transforms in the QROM. The 'CCA bound' column shows the dependence of the upper bound on CCA attacker advantage $\mathsf{Adv}(\mathcal{A})$ against the FO-transformed scheme in terms of the attacker advantage ε against the weak scheme transformed by FO, and $\mathcal{A}$'s oracle query depth $d \leq q$ (where q is the total number of random oracle queries). For simplicity, in this table, we only take into account the dependence in ε, and neglect other additive terms and (small) multiplicative constants. In all cases listed, the run-time of the weak scheme attacker is within a constant factor of the run-time of the CCA scheme. The required weak scheme security notion is shown in column 'Weak scheme'. The 'Security loss' column indicates the bit security loss of the CCA bound (see text). Note that all the weak schemes are not required to enjoy perfect correctness of decryption.

	CCA bound	Security loss	Weak scheme
[10]	$q^{3/2} \cdot \varepsilon^{1/4}$	$3\lambda + 9 \log q$	IND-CPA
[11, 15, 13]	$d^{1/2} \cdot \varepsilon^{1/2}$	$\lambda + \log d$	IND-CPA
[5]	$d^{1/2} \cdot \varepsilon^{1/2}$	$\lambda + \log d$	IND-CPA injective
This work	$d^2 \cdot \varepsilon$	$4 \log d$	IND-CPA injective

state

$$|\psi^O\rangle = \sum_{x' \in X} \sqrt{p_{x'}} |x', O(x')\rangle = \sqrt{p_x} |x, O(x)\rangle + \sum_{x' \neq x} \sqrt{p_{x'}} |x', O(x')\rangle,$$

where $\sum_{x' \in X} p_{x'} = 1$. Let $|\psi_{\neq x}\rangle := \sum_{x' \neq x} \sqrt{\frac{p_{x'}}{1-p_x}} |x', H(x')\rangle$. Recalling that G and H differ only on x, we are in one of the following two cases:

$$|\psi^H\rangle = \sqrt{p_x} |\psi_x^H\rangle + \sqrt{1 - p_x} |\psi_{\neq x}\rangle \text{ and } |\psi^G\rangle = \sqrt{p_x} |\psi_x^G\rangle + \sqrt{1 - p_x} |\psi_{\neq x}\rangle,$$

with $|\psi_x^H\rangle := |x, H(x)\rangle$ and $|\psi_x^G\rangle := |x, G(x)\rangle$.

Since the amplitude of $in = |x\rangle$ in $|\psi^H\rangle$ is $\sqrt{p_x}$, measuring the input register in for $\mathcal{A}$'s query would give the secret x with probability $\mathsf{Adv}(\mathcal{B}) = \Pr[M_{in=|x\rangle} |\psi^O\rangle] = p_x$. This is in fact the strategy of the one-wayness adversary $\mathcal{B}$ constructed from $\mathcal{A}$ in prior O2H security proofs.

On the other hand, as observed in [14], the trace distance between $|\psi^G\rangle$ and $|\psi^H\rangle$ is $\sqrt{1 - (|\psi^G\rangle, |\psi^H\rangle)^2} = \sqrt{1 - (1 - p_x)^2} = \sqrt{(2 - p_x) p_x}$ and therefore there exists a projective measurement $\mathbb{M}_V = (M_V, I - M_V)$ (where M_V is a projector on a subspace V of the state space)[4] that $\mathcal{A}$ can

[4] Here, we assume that $\mathcal{A}$ outputs 1 when the result of measurement space is a state in subspace V.

perform on $|\psi^O\rangle$ to distinguish the case $O = H$ from $O = G$ with distinguishing advantage $\mathsf{Adv}(\mathcal{A}) = \|M_V|\psi^H\rangle\|^2 - \|M_V|\psi^G\rangle\|^2 = \sqrt{(2-p_x)p_x}$ (see [17, Chapter 9]). The existence of such a distinguisher with a square root advantage $\approx \sqrt{2p_x}$ led the authors of [14, 5] to the suggestion that removing the square-root loss from the O2H security reduction may be impossible in the quantum setting.

Let us exhibit such the worst-case M_V that $\mathcal{A}$ could use. Consider $M_V = |v\rangle\langle v|$ that projects the state on a single unit vector $|v\rangle$, with $|v\rangle$ defined as lying on the plane spanned by $|\psi^G\rangle$ and $|\psi^H\rangle$, and at angle $\pi/4 + \theta/2$ from $|\psi^G\rangle$ if $|\psi^H\rangle$ is at angle θ from $|\psi^G\rangle$. Then $\mathsf{Adv}(\mathcal{A}) = \cos^2(\pi/4 + \theta/2) - \cos^2(\pi/4 - \theta/2) = \sin\theta = \sqrt{1-(1-p_x)^2} = \sqrt{(2-p_x)p_x} \approx \sqrt{2p_x}$.

Our MRM technique for resolving the above conundrum stems from the observation that to achieve its high $\approx \sqrt{2p_x}$ advantage, the above example distinguisher $\mathcal{A}$ uses a measurement $\mathbb{M}_V$ that itself *encodes the secret* x. Indeed, in the measurement vector $|v\rangle$ the state $in = |x\rangle$ has amplitude $\approx 1/\sqrt{2}$ when p_x is small. Hence, as $\mathcal{A}$ can measure along $|v\rangle$, it must somehow store it and we should be able to extract x from $\mathcal{A}$ with high probability by simply measuring in of $|v\rangle$ in the computational basis.

The above idea raises the question of how to set up the system state to be $|v\rangle$. The answer is simply to let $\mathcal{A}$ perform its distinguishing measurement $\mathbb{M}_V$ on $|\psi^H\rangle$.[5] If the measurement is M_V, the state collapses to the state $M_V|\psi^H\rangle/\|M_V|\psi^H\rangle\|$. In the above example, this is $|v\rangle$ with probability $\approx 1/2$ when p_x is small. In the standard quantum computational model, since $\mathcal{A}$'s measurement $\mathbb{M}_V$ is not performed with respect to the computational basis (note that $|v\rangle$ is a superposition of computational basis vectors), applying $\mathbb{M}_V$ to the oracle output state is implemented by $\mathcal{A}$ as a composition of a unitary U_V followed by a computational basis measurement $\mathbb{M}_\beta$ of a qubit register β corresponding to $\mathcal{A}$'s output bit (where U_V is designed so that it maps the state $|v\rangle$ to a state with $\beta = 1$). Then, setting up the system state to be $|v\rangle$ actually consists in running $\mathcal{A}$ with oracle H to obtain the state $|\psi^H\rangle$, applying U_V followed by $\mathcal{A}$'s output qubit measurement $\mathbb{M}_\beta$, and if the result of the latter measurement is $\beta = 1$, then *rewinding* the collapsed output state of $\mathcal{A}$ to the step before the measurement by applying the inverse unitary U_V^{-1} (so that effectively the measurement projector $M_V = U_V^{-1} M_{\beta=|1\rangle} U_V$ is applied on the state $|\psi^H\rangle$).

Overall, we obtain an efficient MRM-based quantum algorithm $\mathcal{C}$ to extract x from $\mathcal{A}$ that works as follows for $q = d = 1$: run $\mathcal{A}^H$ and query

[5] Our actual general reduction applies it to a uniform superposition $\frac{1}{2}(|\psi^H\rangle + |\psi^G\rangle)$; see below.

the H oracle to set up the state $|\psi^H\rangle$, continue running $\mathcal{A}$ until it performs its measurement $\mathbb{M}_\beta U_V$ and, if the result is $\beta = |1\rangle$, rewind $\mathcal{A}$ back to just after the query by running U_V^{-1} and apply measurement M_{in} on the in register to extract x, achieving overall success probability $\approx 1/4$ for the above example distinguisher $\mathcal{A}$ when p_x is small.

In our new O2H security proof, we show that (a slight variant of) the above MRM extraction technique works for $q = d = 1$ in the case where $\mathbb{M}_V$ is a general measurement. More precisely, we show that the advantage of any distinguisher $\mathcal{A}$ cannot exceed $4 \cdot \max(\mathsf{Adv}(\mathcal{B}), \mathsf{Adv}(\mathcal{C}))$, where $\mathsf{Adv}(\mathcal{C})$ is the probability that our MRM-based extractor recovers x, and $\mathsf{Adv}(\mathcal{B}) = p_x$ is the probability that the direct query measurement algorithm $\mathcal{B}$ recovers x. Our actual extraction algorithm $\mathcal{D}$ therefore runs $\mathcal{A}$ *twice*: in the first run of $\mathcal{A}$, algorithm $\mathcal{D}$ runs the direct query measurement algorithm $\mathcal{B}$ to attempt to compute x, and in the second run of $\mathcal{A}$, algorithm $\mathcal{D}$ runs our MRM-based algorithm $\mathcal{C}$ to attempt to compute x. By the above bound, the advantage of $\mathcal{A}$ is at most 4 times the success probability of $\mathcal{D}$.

The proof of our new O2H bound is based on re-writing $\mathsf{Adv}(\mathcal{A}) := |\|M_V|\psi^G\rangle\|^2 - \|M_V|\psi^H\rangle\|^2|$ as an inner product of the form

$$\mathsf{Adv}(\mathcal{A}) \leq \left|\Big(|\psi^G\rangle - |\psi^H\rangle, M_V(|\psi^G\rangle + |\psi^H\rangle)\Big)\right|.$$

At this point, we use the crucial fact that since G and H differ only on x, $|\psi^G\rangle - |\psi^H\rangle = |\psi^G_x\rangle + |\psi^H_x\rangle$ is a vector in the subspace $E_{|x\rangle}$ of vectors with $in = |x\rangle$, so it is unchanged by applying a projection $M_{in=|x\rangle}$ onto $E_{|x\rangle}$. Consequently, the inner-product above can be rewritten as

$$\mathsf{Adv}(\mathcal{A}) \leq \left|\Big(M_{in=|x\rangle}(|\psi^G\rangle - |\psi^H\rangle), M_{in=|x\rangle}M_V(|\psi^G\rangle + |\psi^H\rangle)\Big)\right|.$$

Now, we observe that the norm $\|M_{in=|x\rangle}(|\psi^G\rangle - |\psi^H\rangle)\|$ of the vector on the left of the inner-product is (up to a factor of 2) the square-root of the advantage p_x of the direct measurement extraction algorithm $\mathcal{B}$, whereas the norm

$$\|M_{in=|x\rangle}M_V(|\psi^G\rangle + |\psi^H\rangle)\| = \|M_{in=|x\rangle}U_V^{-1}M_{\beta=|1\rangle}U_V(|\psi^G\rangle + |\psi^H\rangle)\|$$

of the vector on the right of the inner-product is (up to a factor of 2) the square-root of the advantage of a variant of the MRM-based extraction algorithm $\mathcal{C}$. Applying the Cauchy-Schwarz inequality gives our bound

$$\mathsf{Adv}(\mathcal{A}) \leq 4 \cdot \sqrt{\mathsf{Adv}(\mathcal{B})} \cdot \sqrt{\mathsf{Adv}(\mathcal{C})} \leq 4 \cdot \max(\mathsf{Adv}(\mathcal{B}), \mathsf{Adv}(\mathcal{C})),$$

for $q = d = 1$. We extend our O2H security proof to the case of any depth $d \geq 1$ by applying a standard hybrid argument over d hybrid distributions in which the oracle O is used only to answer the i-th depth of $\mathcal{A}$, which leads to an additional loss of a factor d in our bound on $\mathsf{Adv}(\mathcal{A})$.

We apply the new O2H lemma to the FO transform, by showing that a slight variant of the proof of security for the $\mathsf{FO}^{\not\perp}$ ('implicit rejection') variant based on the 'double-sided' O2H lemma from [5] suffices for use with our new O2H lemma, without any significant reduction cost. The reason we cannot directly plug in our new 'double-sided' O2H lemma in the FO security proof of [5] is the limitation of our new O2H lemma to 'efficiently checkable' events for the definition of distinguisher $\mathcal{A}$. Our modified proof applies the lemma with the event '$\mathcal{A}$ outputs 1' instead. By the general tight equivalence results of [5, Theorem 5], we also obtain an improved security proof for other variants $\mathsf{FO}^{\perp}$ ('explicit rejection') and $\mathsf{FO}_m^{\not\perp}$ (key derived from message only).

Open problems. Our new O2H security proof for $q = d = 1$ oracle queries crucially makes use of the fact that $|\psi^G\rangle - |\psi^H\rangle$ is in the subspace of vectors with $in = |x\rangle$. This property may no longer be satisfied after $q > 1$ queries, and currently, we handle this difficulty via a hybrid argument that loses a factor q in the advantage (in the presentation of our reduction we actually only lose a factor $d \leq q$ that is the query depth, but in the worst-case we have $d = q$). The security proofs of [1, 5] make use of semi-classical oracles or a variant of Zhandry's quantum query recording technique [26] to reduce (or even eliminate) the loss factor q in the advantage, but they do not seem to be easily compatible with our MRM technique. An interesting open problem is to find an even tighter security proof that combines our MRM technique with those techniques to give a fully tight reduction for O2H in the quantum setting. Relaxing the 'double-sided' aspect of our O2H Lemma to a 'single-sided' variant (like the original O2H Lemma [23]) is also left as an interesting question. Removing the injectivity assumption and finding other applications for our O2H Lemma and the underlying MRM technique are further questions left open by our work.

Additional related work. To the best of our knowledge, the use of quantum circuit rewinding is novel in the context of the O2H Lemma, but there is a body of work using different forms of quantum circuit rewinding in other applications, notably in the analysis of quantum security of zero-knowledge protocols. Watrous [24] presented a *quantum rewinding lemma*, which is a procedure involving multiple 'measure-rewind' iterations with interleaved unitary gates, in order to approximate a desired collapsed measured state with any desired fidelity. The procedure assumes a near independence of the measurement probabilities on the input state, which suffices to prove the zero-knowledge property of certain protocols. Our MRM technique does not make such near independence assumptions (indeed the measurement distribution of the distinguisher may strongly depend on the input state), but only applies one 'measure-rewind-measure' iteration. Unruh [22] presented a form of rewinding extraction technique for proving soundness of zero-knowledge proof of knowledge protocols against quantum attacks. However, the purpose of rewinding there is to approximate the previous state of the attacker while minimising the disturbance of the measurement, whereas in our MRM technique, we actually *want* the measurement to disturb the state in order to extract knowledge from the measurement vector. Later work by Ambianis et al. [2] showed the necessity of restrictions of Unruh's rewinding in the context of quantum-secure proofs of knowledge.

2 Preliminaries

For a finite set $\mathcal{H}$, we denote by $H \overset{\$}{\leftarrow} \mathcal{H}$ the sampling of a uniformly random element H from $\mathcal{H}$. If $\mathcal{A}$ is an algorithm, we denote by $b \leftarrow \mathcal{A}(z)$ the assignment to b of the output of $\mathcal{A}$ run on input z.

Let $\mathbb{C}$ denote the set of complex numbers. For $z \in \mathbb{C}$, we denote the absolute value of z by $|z|$ and the complex conjugate of z by $\bar{z}$. The (complex) inner product between two vectors $|u\rangle = (u_0, \ldots, u_{n-1})$ and $|v\rangle = (v_0, \ldots, v_{n-1})$ in $\mathbb{C}^n$ is denoted by $(|u\rangle, |v\rangle) := \sum_i \bar{u}_i \cdot v_i$. Let $|v\rangle \in \mathbb{C}^n$, then $\||v\rangle\| = \sqrt{(|v\rangle, |v\rangle)}$ denotes its Euclidean norm. For a linear transformation M, the Hermitian (adjoint) operation on M is denoted by $M^\dagger$.

2.1 Quantum Computations

A qubit is a quantum system defined over $\{0, 1\}$. Given two orthonormal vectors $|0\rangle, |1\rangle$, let $\mathbb{S}$ be the state space of a single qubit, namely

$$\mathbb{S} = \left\{ \alpha_0|0\rangle + \alpha_1|1\rangle : |\alpha_0|^2 + |\alpha_1|^2 = 1,\ \alpha_0, \alpha_1 \in \mathbb{C} \right\}.$$

For an integer $N \geq 1$, the state space of a quantum system (register) of N qubits is the N-fold tensor product of $\mathbb{S}$ and is denoted by

$$\mathbb{S}^{\otimes N} = \left\{ \sum_{\mathbf{in} \in \{0,1\}^N} \alpha_{\mathbf{in}} |in_1\rangle \cdots |in_N\rangle : \sum_{\mathbf{in} \in \{0,1\}^N} |\alpha_{\mathbf{in}}|^2 = 1,\ \alpha_{\mathbf{in}} \in \mathbb{C} \right\}.$$

For $\mathbf{x} = (x_1, \ldots, x_N) \in \{0,1\}^N$, the associated *computational basis vector* of $\mathbb{S}^{\otimes N}$ is $\mathbf{x} = |x_1\rangle|x_2\rangle \cdots |x_N\rangle$, and is denoted by $|\mathbf{x}\rangle$. The set of all 2^N computational basis states $\{|\mathbf{x}\rangle\}$ forms an orthonormal basis for $\mathbb{S}^{\otimes N}$. A linear combination $|\phi\rangle = \sum_{\mathbf{x}\in\{0,1\}^N} \alpha_{\mathbf{x}}|\mathbf{x}\rangle$ of computational basis states $|\mathbf{x}\rangle$ is referred to as a *superposition* of computational basis states. We refer to the weight $\alpha_{\mathbf{x}}$ as the *amplitude* of $|\mathbf{x}\rangle$ in state $|\phi\rangle$.

Given the state $|\phi_{in}\rangle \in \mathbb{S}^{\otimes N}$ of an N-qubit register *in* and a value $y \in \{0,1\}^N$, we denote by $M_{in=|y\rangle} : \mathbb{S}^{\otimes N} \to \mathbb{S}^{\otimes N}$ the operator that applies the projection $|y\rangle\langle y|$ map to the state $|\phi_{in}\rangle$ of register *in* to get the new state $|y\rangle\langle y||\phi_{in}\rangle$. This projector can be generalized to a projector M_{E_V} onto a subspace $E_V = \{\sum_{in\in V} \alpha_{in}|in\rangle : \alpha_{in} \in \mathbb{C}\}$ defined by a subset $V \subseteq \{0,1\}^N$, which applies the projection map $\sum_{y\in V} |y\rangle\langle y|$ to a state $|\phi_{in}\rangle \in \mathbb{S}^{\otimes N}$. For example, for a subset $S \subseteq \{0,1\}^N$, we define $S^{\oplus n} := \{\mathbf{in} \in (\{0,1\}^N)^n : \ \exists\ i \ \text{ with } in_i \in S\}$, and then $M_{E_{S^{\oplus n}}}$ is the projector onto subspace $E_{S^{\oplus n}} := \{\sum_{\mathbf{in}\in S^{\oplus n}} \alpha_{\mathbf{in}}|\mathbf{in}\rangle : \alpha_{\mathbf{in}} \in \mathbb{C}\}$. We use the same notation for operators and projectors even if they are applied to non-normalized vectors in $\mathbb{C}^N$. It can be checked that any projector operator M_{E_V} is Hermitian (i.e., we have $M^\dagger = M$) and idempotent (i.e., we have $M^2 = M$).

A *measurement* in the computational basis on a register *in* that is in state $|\phi_{in}\rangle \in \mathbb{S}^{\otimes N}$ returns the measurement result $y \in \{0,1\}^N$ with probability $P = \|M_{in=|y\rangle}|\phi_{in}\rangle\|^2$ and changes ('collapses') the state of *in* to $|\phi'_{in}\rangle = \frac{M_{in=|y\rangle}|\phi_{in}\rangle}{\|M_{in=|y\rangle}|\phi_{in}\rangle\|}$. Such a measurement of register *in* is denoted by $\mathbb{M}_{in}$. A general projective measurement is defined by a set of projection operators $\{M_1, \ldots, M_n\}$ where M_i's project onto subspaces V_i that are mutually orthogonal and whose sum is the whole state space. For example, for any subspace V of $\mathbb{S}^{\otimes N}$, we can define the projective measurement $\mathbb{M}_V = (M_V, I - M_V)$ where M_V is the projector onto V and $I - M_V$ is the projector onto the orthogonal complement of V. Any general projective measurement can be implemented by composing a unitary operation followed by a measurement in computational basis. Each measurement costs one time unit.

A quantum algorithm executes a sequence of unitary gate operations for a fixed set F containing Hadamard, phase, CNOT and $\pi/8$ gates. Each gate is also counted as one unit of time. The overall time taken to perform a quantum algorithm $\mathcal{A}$ is denoted by $\mathcal{T}_{\mathcal{A}}$. An efficient quantum algorithm runs a polynomial-time (in N) sequence of gate operations or measurements.

Given a function $H : X \to Y = \{0,1\}^N$, a quantum-accessible oracle O of H is modeled by a unitary transformation U_H operating on two registers in, out with state spaces $\mathbb{S}^{\otimes N}$, in which $|x, y\rangle$ is mapped to $|x, y \oplus H(x)\rangle$, where $\oplus$ denotes XOR group operation on Y. A quantum algorithm with quantum random oracle O performs a mix of classical and quantum unitary algorithms. This can be efficiently converted, up to a constant factor overhead and same number of oracle queries [17], to a purely unitary algorithm that applies a unitary followed by a final set of measurements. A purely unitary algorithm making q oracle queries to O is denoted by $(OU_i)_{i=1}^{q}$, where U_i is a unitary operation applied before the i-th call to oracle O. Following [5], we model a quantum algorithm $\mathcal{A}$ making parallel queries to a quantum oracle O as a quantum algorithm making $d \leq q$ queries to an oracle $O^{\otimes n}$ consisting of $n = q/d$ parallel copies of oracle O. Given an input state of n pairs of *in/out* registers $|x_1\rangle|y_1\rangle \cdots |x_n\rangle|y_n\rangle$, the oracle of $O^{\otimes n}$ maps it to the state $|x_1\rangle|y_1 \oplus O(x_1)\rangle \cdots |x_n\rangle|y_n \oplus O(x_n)\rangle$. We call d the algorithm's *query depth*, n the parallelization factor, and $q = n \cdot d$ the total number of oracle queries.

2.2 Original One-Way to Hiding (O2H) Lemma

We now recall the One-Way to Hiding (O2H) Lemma, as stated in [1] (this formulation generalizes Unruh's original O2H Lemma [23]).

Lemma 2.1 ([1, Theorem 3]). *Let $G, H \colon X \to Y$ be random functions, z be a random value, and $S \subseteq X$ be a random set such that $G(x) = H(x)$ for every $x \notin S$. The tuple (G, H, S, z) may have an arbitrary joint distribution. Furthermore, let $\mathcal{A}^H$ be a quantum oracle algorithm which queries H with depth at most d. Let* Event *be an arbitrary classical event. Define the oracle algorithm $\mathcal{B}^H(z)$ as follows: sample $i \xleftarrow{\$} \{0, \dots, d-1\}$; run $\mathcal{A}^H(z)$ until just before its i-th round of queries to H; measure all query input registers in the computational basis, and output the set T of measurement outcomes. Then*

$$\mathsf{Adv}(\mathcal{A}) \leq 2d\sqrt{\mathsf{Adv}(\mathcal{B})} \quad \textit{and} \quad |\sqrt{P_{\mathsf{left}}} - \sqrt{P_{\mathsf{right}}}| \leq 2d\sqrt{\mathsf{Adv}(\mathcal{B})},$$

where $\mathsf{Adv}(\mathcal{A}) := |P_{\mathsf{left}} - P_{\mathsf{right}}|$ *with*

$$P_{\mathsf{left}} := \Pr[\mathsf{Event} : \mathcal{A}^H(z)], \quad P_{\mathsf{right}} := \Pr[\mathsf{Event} : \mathcal{A}^G(z)],$$

and

$$\mathsf{Adv}(\mathcal{B}) := \Pr[S \cap T \neq \emptyset : T \leftarrow \mathcal{B}^H(z)].$$

3 Main Results

The following result will prove useful later on in the proof of Lemma 3.2.

Lemma 3.1. *For any vectors $|\phi_1\rangle$ and $|\phi_2\rangle$, we have*

$$|\||\phi_1\rangle\|^2 - \||\phi_2\rangle\|^2| \leq |(|\phi_1\rangle - |\phi_2\rangle, |\phi_1\rangle + |\phi_2\rangle)|.$$

Proof. Let $x_1 = |\phi_1\rangle - |\phi_2\rangle$ and $x_2 = |\phi_1\rangle + |\phi_2\rangle$. Then, we have:

$$\begin{aligned}\left|\frac{\|x_1 + x_2\|^2}{4} - \frac{\|x_1 - x_2\|^2}{4}\right| &= \frac{|(x_1 + x_2, x_1 + x_2) - (x_1 - x_2, x_1 - x_2)|}{4} \\ &= |\mathsf{Real}((x_1, x_2))| \leq |(x_1, x_2)|,\end{aligned}$$

where $\mathsf{Real}(z)$ denotes the real part of a complex number z. □

3.1 O2H with Measure-Rewind-Measure (MRM)

We first describe the fixed input version of our result, where G, H, S, z are all fixed, and then we extend it to case of random G, H, S, z. Note that below, the value z can depend on G, H, S, so can serve to provide the adversary with a 'hint' about G, H, S (for instance, in our application later on, the value z contains an encryption of S).

Lemma 3.2 (Fixed O2H with MRM). *Let $G, H \colon X \to Y$ be fixed functions, z be a fixed value, and $S \subseteq X$ be a fixed set such that $G(x) = H(x)$ for every $x \notin S$. Furthermore, let $\mathcal{A}^O$ be a quantum oracle algorithm which queries an oracle O with depth d. Then we can construct unitary algorithms $\{\mathcal{A}_i^O(z)\}_{0 \leq i < d}$, $\{\mathcal{B}_i^{G,H}(z)\}_{0 \leq i < d}$, and $\{\mathcal{C}_i^{G,H}(z)\}_{0 \leq i < d}$ with $\mathcal{T}_{\mathcal{A}_i^O} \approx \mathcal{T}_{\mathcal{A}^O}$, $\mathcal{T}_{\mathcal{B}_i^{G,H}} \lesssim \mathcal{T}_{\mathcal{A}_i^O}$ and $\mathcal{T}_{\mathcal{C}_i^{G,H}} \approx 2 \cdot \mathcal{T}_{\mathcal{A}_i^O}$ (for all i) and such that*

$$\mathsf{Adv}(\mathcal{A}^O) \leq \sum_{i=0}^{d-1} \mathsf{Adv}(\mathcal{A}_i^O), \tag{1}$$

and (for all i):

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}_i^O) &\leq 4\sqrt{\mathsf{Adv}(\mathcal{B}_i^{G,H}) \cdot \mathsf{Adv}(\mathcal{C}_i^{G,H})} \\ &\leq 4 \max\{\mathsf{Adv}(\mathcal{B}_i^{G,H}), \mathsf{Adv}(\mathcal{C}_i^{G,H})\}.\end{aligned} \tag{2}$$

Here $\mathsf{Adv}(\mathcal{A}^O) := |P_{\mathsf{left}} - P_{\mathsf{right}}|$ *with*

$$P_{\mathsf{left}} := \Pr[1 \leftarrow \mathcal{A}^H(z)],\ P_{\mathsf{right}} := \Pr[1 \leftarrow \mathcal{A}^G(z)],$$

$$\mathsf{Adv}(\mathcal{A}_i^O) := |\Pr[1 \leftarrow \mathcal{A}_i^H(z)] - \Pr[1 \leftarrow \mathcal{A}_i^G(z)]|,$$
$$\mathsf{Adv}(\mathcal{B}_i^{G,H}) := \Pr[S \cap T_{\mathcal{B}_i} \neq \emptyset : T_{\mathcal{B}_i} \leftarrow \mathcal{B}_i^{G,H}(z)],$$

and

$$\mathsf{Adv}(\mathcal{C}_i^{G,H}) := \Pr[S \cap T_{\mathcal{C}_i} \neq \emptyset : T_{\mathcal{C}_i} \leftarrow \mathcal{C}_i^{G,H}(z)].$$

Proof. Let $O_G^{\otimes n}$ and $O_H^{\otimes n}$ be the n-wise parallel quantum oracles for G and H, respectively. As in [5, Lemma 5], we define another quantum oracle $O_{G,H}^{\otimes n}$, which is used to put the sum and difference of $O_G^{\otimes n}$ and $O_H^{\otimes n}$ in superposition, entangled with another bit b. This can be configured so that the additional bit register b decides which oracle is in use. Concretely, we define

$$O_{G,H}^{\otimes n} := (O_H^{\otimes n} \otimes |+\rangle\langle +|) + (O_G^{\otimes n} \otimes |-\rangle\langle -|),$$

where $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$.Therefore, the oracle $O_{G,H}^{\otimes n}$ maps the state $|\psi\rangle|+\rangle$ to the state $O_H^{\otimes n}(|\psi\rangle)|+\rangle$ and the state $|\psi\rangle|-\rangle$ to the state $O_G^{\otimes n}(|\psi\rangle)|-\rangle$. As observed in [5], it can be efficiently implemented by applying a Hadamard gate before and after a conditional evaluation map that applies O_H if $b = 0$ and O_G if $b = 1$. By setting the b bit register to start in the superposition state $\frac{|+\rangle+|-\rangle}{\sqrt{2}} = |0\rangle$, and applying $O_{G,H}^{\otimes n}$ we get a state with the sum and differences of the oracle output states entangled with the bit b:

$$\begin{aligned} O_{G,H}^{\otimes n}(|\psi\rangle|0\rangle) &= \frac{1}{\sqrt{2}} \cdot \Big(O_H^{\otimes n}(|\psi\rangle)|+\rangle + O_G^{\otimes n}(|\psi\rangle)|-\rangle\Big) \\ &= \frac{1}{2} \cdot \Big(O_H^{\otimes n}|\psi\rangle + O_G^{\otimes n}|\psi\rangle\Big) \otimes |0\rangle \\ &+ \frac{1}{2} \cdot \Big(O_H^{\otimes n}|\psi\rangle - O_G^{\otimes n}|\psi\rangle\Big) \otimes |1\rangle. \end{aligned} \tag{3}$$

Looking ahead, we will use the above bit b in algorithms $\mathcal{B}_i$ and $\mathcal{C}_i$ and aim to measure $b = 1$ in the former and $b = 0$ in the latter, so that we get the difference and sum states, respectively, in the remaining registers.

We now present our hybrid algorithms for $i \in \{0, \ldots, d-1\}$. The i-th hybrid pair of algorithms for $\mathcal{A}$ corresponds to running $\mathcal{A}$ with its first i oracle calls answered with $O_H^{\otimes n}$, $\mathcal{A}$'s $(i + 1)$-th call answered by $O_O^{\otimes n}$ where $O \in \{G, H\}$ is $\mathcal{A}$'s oracle, and $\mathcal{A}$'s final $d - (i + 1)$ calls answered using $O_G^{\otimes n}$. The extraction algorithms $\mathcal{B}_i$ and $\mathcal{C}_i$ detailed below will run $\mathcal{A}$ similarly except with the $(i+1)$-th query answered with the superposition oracle $O_{G,H}^{\otimes n}$. We define the four hybrid algorithms below. Recall that the total number of quantum oracle queries of $\mathcal{A}$ equals $q = n \cdot d$, where n is the parallelization factor, and that $\mathcal{A}$ applies a unitary U_j in between its $(j - 1)$-th and j-th oracle call.

- Algorithm $\mathcal{A}_i^O$ for $O \in \{O_H^{\otimes n}, O_G^{\otimes n}\}$. This algorithm starts with 0's in registers $|aux\rangle \bigotimes_{i=1}^n (|in_i\rangle|out_i\rangle)|\beta\rangle$, where aux is $\mathcal{A}$'s auxiliary working register, and $\beta \in \{0,1\}$ is $\mathcal{A}$'s output bit. Algorithm $\mathcal{A}_i^O$ first runs $(O_H^{\otimes n} U_j)_{j=1}^i$ to get to state $|st_{2i,i}\rangle$, then runs OU_{i+1} to get to state $|st_{2i+2,i}\rangle$, and finally performs $(O_G^{\otimes n} U_j)_{j=i+2}^d$, which takes us to state $|st_{2d,i}\rangle$. This is finalized by a unitary operation U_{d+1}, which gives state $|st_{2d+1,i}\rangle$, to which the output bit measurement $\mathbb{M}_\beta$ is applied. The algorithm outputs the measurement result bit β.

- Algorithm $\mathcal{B}_i^{G,H}$. This algorithm starts with one extra bit register as input compared to previous algorithm. The first $2n+2$ registers are exactly the same as those in $\mathcal{A}_i^O$ and the last register is devoted to bit b to implement $O_{G,H}^{\otimes n}$. All registers are initialized to 0. Then, this algorithm runs $(O_H^{\otimes n} U_j)_{j=1}^i$ (giving a state $|st'_{2i,i}\rangle$), then applies $O_{G,H}^{\otimes n} U_{i+1}$ (giving a state $|st'_{2i+2,i}\rangle$), and then performs a measurement $\mathbb{M}_b$ of the b register (i.e., just after the $(i+1)$-th oracle call). If the result of this measurement is 1, then a measurement $\mathbb{M}_{\mathbf{in}}$ of the oracle's input register $\mathbf{in} = (in_1, \ldots, in_n)$ is conducted. This can also be seen as n parallel measurements $\mathbb{M}_{in_1} \ldots \mathbb{M}_{in_n}$. The algorithm terminates by outputting the results of the measurements.

- Algorithm $\mathcal{C}_i^{G,H}$. This algorithm has the same registers as the previous one. All registers are initialized to 0. This algorithm applies $(O_H^{\otimes n} U_j)_{j=1}^i$, $O_{G,H}^{\otimes n} U_{i+1}$, $(O_G^{\otimes n} U_j)_{j=i+2}^d$ and U_{d+1}. The states after applying those operations are called $|st''_{2i,i}\rangle$, $|st''_{2i+2,i}\rangle$, $|st''_{2d,i}\rangle$ and $|st''_{2d+1,i}\rangle$, respectively. Then the measurements $\mathbb{M}_b$, and $\mathbb{M}_\beta$ are applied. If the result of $\mathbb{M}_b$ equals 0 and the result of $\mathbb{M}_\beta$ equals 1, then the following (rewinding) transformations are applied back to the point just after the $(i+1)$-th oracle call: $U_{d+1}^\dagger$, $((O_G^{\otimes n} U_j)^\dagger)_{j=i+2}^d$, resulting in states called $|st'''_{2d,i}\rangle$, and $|st'''_{2i+2,i}\rangle$, respectively. Finally, a measurement with respect to $\mathbf{in}$ is performed, and the algorithm outputs the result of the measurement.

One can check that $\mathcal{T}_{\mathcal{A}_i^O} \approx \mathcal{T}_{\mathcal{A}^O}$, $\mathcal{T}_{\mathcal{B}_i^{G,H}} \lesssim \mathcal{T}_{\mathcal{A}_i^O}$ and that $\mathcal{T}_{\mathcal{C}_i^{G,H}} \approx \mathcal{T}_{\mathcal{B}_i^{G,H}} + 2(\mathcal{T}_{\mathcal{A}_i^O} - \mathcal{T}_{\mathcal{B}_i^{G,H}}) \leq 2 \cdot \mathcal{T}_{\mathcal{A}_i^O}$.

We have $\mathcal{A}_0^{O=G} = \mathcal{A}^G$, $\mathcal{A}_{d-1}^{O=H} = \mathcal{A}^H$ and $\mathcal{A}_i^{O=H} = \mathcal{A}_{i+1}^{O=G}$ for $0 \leq i \leq d-2$ (here and in the following, we use the shorthand $O = G$ and $O = H$

for $O = O_G^{\otimes n}$ and $O = O_H^{\otimes n}$ respectively). This implies that:

$$\begin{aligned}
\mathsf{Adv}(\mathcal{A}) &= |\Pr[1 \leftarrow \mathcal{A}^G] - \Pr[1 \leftarrow \mathcal{A}^H]| \\
&= |\Pr[1 \leftarrow \mathcal{A}_0^{O=G}] - \Pr[1 \leftarrow \mathcal{A}_{d-1}^{O=H}]| \\
&= \left| \sum_{i=0}^{d-1} \Big(\Pr[1 \leftarrow \mathcal{A}_i^{O=G}] - \Pr[1 \leftarrow \mathcal{A}_i^{O=H}] \Big) \right| \\
&\leq \sum_{i=0}^{d-1} \Big| \Pr[1 \leftarrow \mathcal{A}_i^{O=G}] - \Pr[1 \leftarrow \mathcal{A}_i^{O=H}] \Big| \\
&= \sum_{i=0}^{d-1} \mathsf{Adv}(\mathcal{A}_i^O),
\end{aligned}$$

where the first and the last equalities are obtained based on the definitions, the second equality is the result of a simple telescopic argument, and the only inequality follows from the triangle inequality. This proves (1).

We now proceed to prove (2). Fix $0 \leq i \leq d-1$. Let

$$\begin{aligned}
W_i &:= U_{d+1}(O_G^{\otimes n} U_j)_{j=i+2}^d, \\
|\psi_{i,F}\rangle &:= |st_{2i+2,i}^{O=H}\rangle - |st_{2i+2,i}^{O=G}\rangle, \\
|\psi_{i,B}\rangle &:= W_i^\dagger M_{\beta=|1\rangle} W_i (|st_{2i+2,i}^{O=H}\rangle + |st_{2i+2,i}^{O=G}\rangle).
\end{aligned}$$

We first study $\mathsf{Adv}(\mathcal{A}_i^O)$. We have:

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathcal{A}_i^{O=H}] - \Pr[1 \leftarrow \mathcal{A}_i^{O=G}]| \\
&\quad = \Big| \|M_{\beta=|1\rangle} |st_{2d+1,i}^{O=H}\rangle\|^2 - \|M_{\beta=|1\rangle} |st_{2d+1,i}^{O=G}\rangle\|^2 \Big| \\
&\quad \leq \Big| \Big(M_{\beta=|1\rangle} (|st_{2d+1,i}^{O=H}\rangle - |st_{2d+1,i}^{O=G}\rangle), M_{\beta=|1\rangle} (|st_{2d+1,i}^{O=H}\rangle + |st_{2d+1,i}^{O=G}\rangle) \Big) \Big| && (4) \\
&\quad = \Big| \Big(M_{\beta=|1\rangle} W_i |\psi_{i,F}\rangle, M_{\beta=|1\rangle} W_i (|st_{2i+2,i}^{O=H}\rangle + |st_{2i+2,i}^{O=G}\rangle) \Big) \Big| && (5) \\
&\quad = \Big| \Big(|\psi_{i,F}\rangle, W_i^\dagger M_{\beta=|1\rangle}^\dagger M_{\beta=|1\rangle} W_i (|st_{2i+2,i}^{O=H}\rangle + |st_{2i+2,i}^{O=G}\rangle) \Big) \Big| \\
&\quad = |(|\psi_{i,F}\rangle, |\psi_{i,B}\rangle)| && (6) \\
&\quad = |(M_{\mathbf{in} \in S^{\oplus n}} |\psi_{i,F}\rangle, |\psi_{i,B}\rangle)| && (7) \\
&\quad = \Big| \Big(M_{\mathbf{in} \in S^{\oplus n}} |\psi_{i,F}\rangle, M_{\mathbf{in} \in S^{\oplus n}}^\dagger |\psi_{i,B}\rangle \Big) \Big| && (8) \\
&\quad \leq \|M_{\mathbf{in} \in S^{\oplus n}} |\psi_{i,F}\rangle\| \cdot \|M_{\mathbf{in} \in S^{\oplus n}}^\dagger |\psi_{i,B}\rangle\|, && (9)
\end{aligned}$$

where (4) follows from Lemma 3.1, (5) is obtained based on the definitions of $\mathcal{A}_i^O$ and $|\psi_{i,F}\rangle$, (6) employs the fact that $M_{\beta=|1\rangle}$ is a Hermitian and idempotent transformation and the definition of $|\psi_{i,B}\rangle$, (8) uses the fact that $M_{\mathbf{in}\in S^{\oplus n}}$ is idempotent, and (9) follows from the Cauchy-Schwarz inequality. Finally, the equality in (7) exploits the fact that $|\psi_{i,F}\rangle$ may have non-zero amplitudes only for computational basis vectors $\mathbf{in} \in S^{\oplus n}$ (we recall that $S^{\oplus n}$ is the set of n-dimensional vectors **in** having at least one component in the set S on which H and G differ). To see the latter fact, one can write

$$\begin{aligned}|st^{O}_{2i+2,i}\rangle = &\sum_{\mathbf{in}\in S^{\oplus n},\mathbf{out}} \alpha_{\mathbf{in},\mathbf{out}} |in_1\rangle|out_1 \oplus O(in_1)\rangle \cdots |in_n\rangle|out_n \oplus O(in_n)\rangle \\ &+ \sum_{\mathbf{in}\in \overline{S^{\oplus n}},\mathbf{out}} \alpha_{\mathbf{in},\mathbf{out}} |in_1\rangle|out_1 \oplus O(in_1)\rangle \cdots |in_n\rangle|out_n \oplus O(in_n),\end{aligned}$$

with $\overline{S^{\oplus n}} = \{0,1\}^{N\cdot n} \setminus S^{\oplus n}$. From this, we deduce that difference vector $|\psi_{i,F}\rangle$ only has a component along $S^{\oplus n}$, as the sum over $\overline{S^{\oplus n}}$ (and **out**) is identical for both $|st^{G}_{2i+2,i}\rangle$ and $|st^{H}_{2i+2,i}\rangle$.

Based on the definitions of $O^{\otimes n}_{G,H}$, $\mathcal{B}^{G,H}_i$ and $\mathcal{C}^{G,H}_i$, and the superposition property (3), the following holds:

$$|st'_{2i+2,i}\rangle = |st''_{2i+2,i}\rangle = \frac{1}{2}\Big(|\psi_{i,F}\rangle|1\rangle + (|st^{O=H}_{2i+2,i}\rangle + |st^{O=G}_{2i+2,i}\rangle)|0\rangle\Big). \quad (10)$$

On the one hand, we have

$$\begin{aligned}\mathsf{Adv}(\mathcal{B}^{G,H}_i) &= \Pr[S \cap T_{\mathcal{B}_i} \neq \emptyset,\ T_{\mathcal{B}_i} \leftarrow \mathcal{B}^{G,H}_i(z)] \\ &= \left\| M_{\mathbf{in}\in S^{\oplus n}} \frac{M_{b=|1\rangle}|st'_{2i+2,i}\rangle}{\|M_{b=|1\rangle}|st'_{2i+2,i}\rangle\|} \right\|^2 \cdot \|M_{b=|1\rangle}|st'_{2i+2,i}\rangle\|^2 \\ &= \left\| M_{\mathbf{in}\in S^{\oplus n}} \frac{|\psi_{i,F}\rangle|1\rangle}{\||\psi_{i,F}\rangle|1\rangle\|} \right\|^2 \cdot \left\| \frac{1}{2}|\psi_{i,F}\rangle|1\rangle \right\|^2 \quad (11) \\ &= \frac{1}{4}\|M_{\mathbf{in}\in S^{\oplus n}}|\psi_{i,F}\rangle\|^2, \quad (12)\end{aligned}$$

where (11) follows from (10). On the other hand, by definition of $\mathcal{C}^{G,H}_i$, we have that:

$$\begin{aligned}|st'''_{2i+2,i}\rangle &= \frac{W_i^\dagger M_{\beta=|1\rangle} M_{b=|0\rangle} W_i |st''_{2i+2,i}\rangle}{\|M_{\beta=|1\rangle} M_{b=|0\rangle} W_i |st''_{2i+2,i}\rangle\|} \\ &= \frac{W_i^\dagger M_{\beta=|1\rangle} W_i M_{b=|0\rangle} |st''_{2i+2,i}\rangle}{\|M_{\beta=|1\rangle} W_i M_{b=|0\rangle} |st''_{2i+2,i}\rangle\|}, \quad (13) \\ &= \frac{|\psi_{i,B}\rangle|0\rangle}{\||\psi_{i,B}\rangle|0\rangle\|}, \quad (14)\end{aligned}$$

where (13) holds since $M_{b=|0\rangle}$ does not have any effect on U_{d+1} nor on $(U_j O_G^{\otimes n})_{j=i+2}^{d}$ and hence it commutes with W_i, and (14) is obtained using (10) and the definition of $|\psi_{i,R}\rangle$. Finally, one can write:

$$\begin{aligned}
\mathsf{Adv}(\mathcal{C}_i^{G,H}) &= \Pr[S \cap T_{\mathcal{C}_i} \neq \emptyset,\ T_{\mathcal{C}_i} \leftarrow \mathcal{C}_i^{G,H}(z)] \\
&= \|M^{\dagger}_{\mathbf{in}\in S^{\oplus n}} |st'''_{2i+2,i}\rangle\|^2 \cdot \|M_{\beta=|1\rangle} M_{b=|0\rangle} W_i |st''_{2i+2,i}\rangle\|^2 \\
&= \|M^{\dagger}_{\mathbf{in}\in S^{\oplus n}} |st'''_{2i+2,i}\rangle\|^2 \cdot \|W_i^{\dagger} M_{\beta=|1\rangle} W_i M_{b=|0\rangle} |st''_{2i+2,i}\rangle\|^2 && (15) \\
&= \left\| M^{\dagger}_{\mathbf{in}\in S^{\oplus n}} \frac{|\psi_{i,B}\rangle|0\rangle}{\||\psi_{i,B}\rangle|0\rangle\|} \right\|^2 \cdot \left\| \frac{1}{2} |\psi_{i,B}\rangle|0\rangle \right\|^2 && (16) \\
&= \frac{1}{4} \|M^{\dagger}_{\mathbf{in}\in S^{\oplus n}} |\psi_{i,B}\rangle\|^2, && (17)
\end{aligned}$$

where (15) holds true as $W_i^{\dagger}$ is a unitary operation and $M_{b=|0\rangle}$ commutes with W_i, and (16) follows from (14). Substituting (12) and (17) into (9) proves (2). □

We now extend our O2H Lemma to the random case.

Lemma 3.3 (Random O2H with MRM). *Let $G, H \colon X \to Y$ be random functions, z be a random value, and $S \subseteq X$ be a random set such that $G(x) = H(x)$ for every $x \notin S$. The tuple (G, H, S, z) may have arbitrary joint distribution. Furthermore, let $\mathcal{A}^O$ be a quantum oracle algorithm which queries oracle O with query depth d. Then we can construct an algorithm $\mathcal{D}^{G,H}(z)$ such that $\mathcal{T}_{\mathcal{D}^{G,H}} \lesssim 3 \cdot \mathcal{T}_{\mathcal{A}^O}$ and*

$$\mathsf{Adv}(\mathcal{A}^O) \leq 4d \cdot \mathsf{Adv}(\mathcal{D}^{G,H}).$$

Here $\mathsf{Adv}(\mathcal{A}^O) := |P_{\mathsf{left}} - P_{\mathsf{right}}|$ *with*

$$P_{\mathsf{left}} := \Pr_{H,z}[1 \leftarrow \mathcal{A}^H(z)],\ P_{\mathsf{right}} := \Pr_{G,z}[1 \leftarrow \mathcal{A}^G(z)],$$

and

$$\mathsf{Adv}(\mathcal{D}^{G,H}) := \Pr_{G,H,S,z}[T \cap S \neq \emptyset : T \leftarrow \mathcal{D}^{G,H}(z)].$$

Proof. We first construct $\mathcal{D}^{G,H}$ on input z as follows:

- Sample $i \xleftarrow{\$} \{0, \dots, d-1\}$,
- Run $\mathcal{B}_i^{G,H}(z)$ and $\mathcal{C}_i^{G,H}(z)$ to obtain $T_{\mathcal{B}_i}$ and $T_{\mathcal{C}_i}$, respectively, and
- Return $T := T_{\mathcal{B}_i} \cup T_{\mathcal{C}_i}$.

The run-time bound follows from Lemma 3.2, which states that $\mathcal{T}_{\mathcal{B}_i^{G,H}} \lesssim \mathcal{T}_{\mathcal{A}^O}$ and $\mathcal{T}_{\mathcal{C}^{G,H}} \approx 2 \cdot \mathcal{T}_{\mathcal{A}^O}$. In the following, when we do not explicitly state the subscripts of probabilities or expectations, it means that they are over the internal randomness of the quantum algorithms only. Now, for fixed G, H, S, z, let

$$P_i^{\mathcal{B}_i \vee \mathcal{C}_i}(G, H, S, z) := \Pr[(T_{\mathcal{B}_i} \cap S \neq \emptyset) \vee (T_{\mathcal{C}_i} \cap S \neq \emptyset) : \\ T_{\mathcal{B}_i} \leftarrow \mathcal{B}_i^{G,H}(z), T_{\mathcal{C}_i} \leftarrow \mathcal{C}_i^{G,H}(z)].$$

With the above definition, we can write:

$$\begin{aligned} \mathop{\mathbb{E}}_{G,H,S,z}\Big[P_i^{\mathcal{B}_i \vee \mathcal{C}_i}(G, H, S, z)\Big] &\geq \mathop{\mathbb{E}}_{G,H,S,z}\Big[\max\left\{\mathsf{Adv}(\mathcal{B}_i^{G,H}), \mathsf{Adv}(\mathcal{C}_i^{G,H})\right\}\Big] \\ &\geq \frac{1}{4} \mathop{\mathbb{E}}_{G,H,S,z}[\mathsf{Adv}(\mathcal{A}_j^O)], \end{aligned} \tag{18}$$

where the first inequality uses the fact that, for any two events E_1 and E_2, we have $\Pr[E_1 \vee E_2] \geq \max\{\Pr[E_1], \Pr[E_2]\}$, and the second one follows from Lemma 3.2. We now investigate the advantage of algorithm $\mathcal{D}$:

$$\begin{aligned} \mathsf{Adv}(\mathcal{D}^{G,H}) &= \sum_j \Pr[i = j] \cdot \mathop{\mathbb{E}}_{G,H,S,z}\Big[P_j^{\mathcal{B}_j \vee \mathcal{C}_j}(G, H, S, z)\Big] \\ &\geq \frac{1}{4d} \sum_j \mathop{\mathbb{E}}_{G,H,S,z}[\mathsf{Adv}(\mathcal{A}_j^O)] \\ &\geq \frac{1}{4d} \cdot \mathsf{Adv}(\mathcal{A}^O), \end{aligned}$$

where the first and second inequalities follow from (18) and Lemma 3.2, respectively. □

4 Tighter IND-CCA Proofs for Fujisaki-Okamoto KEMs

Here, we apply our results from Section 3 to prove IND-CCA security of the Fujisaki-Okamoto $\mathsf{FO}^{\not\perp}$ transform, which takes an IND-CPA secure public-key encryption scheme (PKE) and applies a composition of the T transform [10] and the $U^{\not\perp}$ transform [10, 13] to produce an IND-CCA secure Key Encapsulation Mechanism (KEM). Our QROM security proof for $\mathsf{FO}^{\not\perp}$ is obtained by adapting the proof in [5] to work with our new O2H lemma.

4.1 Security Definitions

We recall standard definitions related to PKEs, KEMs and pseudo-random functions (PRFs) in Appendix A. Here we recall less standard definitions that will be needed in the analysis of the transform to an IND-CCA KEM.

We start with the definitions of a valid ciphertext and a security property called "finding failing ciphertext" (FFC). The latter was introduced in [5] to capture a decryption error requirement on the dPKE scheme needed for the IND-CCA security of the $U^{\not\perp}$ transform (recalled below). Notice that the success event of the FFC experiment is *not* efficiently checkable, which may at first sight seem incompatible with our O2H lemma; looking ahead, this event corresponds to the Fail event in the proof of Theorem 4.6, which we handle without invoking our O2H lemma.

Definition 4.1 (Valid Ciphertext). *Let* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *be a deterministic* PKE. *We call a ciphertext* $c \in \mathcal{C}$ *valid for a public key* pk *if there exists a message* $m \in \mathcal{M}$ *such that* $c = \mathsf{Encr}(\mathsf{pk}, m)$*.*

Definition 4.2 (Finding Failing Ciphertext). *Let* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *be a* PKE *and* $\mathcal{A}$ *be an adversary executing an attack against the finding failing ciphertext property (*FFC*), as specified by the following experiment:*

```
$H \xleftarrow{\$} \mathcal{H}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$
$L \leftarrow \mathcal{A}^H(\mathsf{pk})$
return $[\exists m \in \mathcal{M}, c \in L : \mathsf{Encr}(\mathsf{pk}, m) = c \ \wedge \ \mathsf{Decr}(\mathsf{sk}, c) \neq m]$
```

The advantage of $\mathcal{A}$ *in the above experiment is defined as:*

$$\mathsf{Adv}^{\mathsf{FFC}}_{\mathsf{P}}(\mathcal{A}) := \Pr[1 \leftarrow \mathsf{Expt}^{\mathsf{FFC}}_{\mathsf{P}}(\mathcal{A})].$$

In the analysis of the $U^{\not\perp}$ transform, we will also need a dPKE satisfying the following injectivity property.

Definition 4.3 (Injectivity of a dPKE). *Let* $\eta \geq 0$*. A* dPKE *scheme* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *is* η*-injective if*

$$\Pr[\mathsf{Encr}(\mathsf{pk}, \cdot) \textit{ is not injective: } (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda), H \xleftarrow{\$} \mathcal{H}] \leq \eta.$$

4.2 Transforms

In [10], the authors showed how to build a transform T which converts any rPKE scheme $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ into a dPKE scheme $T(\mathsf{P}, G) = (\mathsf{KeyGen}, \mathsf{Encr}_d, \mathsf{Decr})$ using a hash function $G : \mathcal{M} \to \mathcal{R}$, where $\mathcal{R}$ is the space of random coins of rPKE's Encr algorithm. In [5], the authors proved the following security reduction from $\mathsf{IND\text{-}CPA}$ security of rPKE to $\mathsf{OW\text{-}CPA}$ security of $T(\mathsf{P}, G)$. We use this result as is, since it does not suffer from a square-root advantage loss.

Theorem 4.4 ([5, Theorem 1]). *Let* P *be an* rPKE *with message space* $\mathcal{M}$ *and randomness space* $\mathcal{R}$*. Let* $G : \mathcal{M} \to \mathcal{R}$ *be a quantum-accessible random oracle. Let* $\mathcal{A}$ *be a* $\mathsf{OW\text{-}CPA}$ *adversary against* $\mathsf{P}' = T(\mathsf{P}, G)$*. Suppose that* $\mathcal{A}$ *queries* G *at most* q *times with query depth at most* d*. Then we can construct an* $\mathsf{IND\text{-}CPA}$ *adversary* $\mathcal{B}$*, running in time* $\approx \mathcal{T}_{\mathcal{A}}$*, such that:*

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{P}'}(\mathcal{A}) \leq (d+2) \cdot \left(\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P}}(\mathcal{B}) + \frac{8 \cdot (q+1)}{|\mathcal{M}|} \right).$$

The following result provides a bound on the FFC advantage for a scheme obtained via the transform above.

Lemma 4.5 ([5, Lemma 6]). *Let* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *be a* δ*-correct* rPKE *with messages in* $\mathcal{M}$ *and randomness in* $\mathcal{R}$*. Let* $G : \mathcal{M} \to \mathcal{R}$ *be a random oracle, so that* $T(\mathsf{P}, G) := (\mathsf{KeyGen}, \mathsf{Encr}_1, \mathsf{Decr})$ *is a derandomized version of* P*. Suppose that* $T(\mathsf{P}, G)$ *is* η*-injective. Let* $\mathcal{A}$ *be an FFC adversary against* $T(\mathsf{P}, G)$ *which makes at most* q *queries to* G *with query depth at most* d *and returns a list of at most* q_{dec} *ciphertexts. Then*

$$\mathsf{Adv}^{\mathsf{FFC}}_{T(\mathsf{P},G)}(\mathcal{A}) \leq ((4d+1)\delta + \sqrt{3\eta}) \cdot (q + q_{dec}) + \eta.$$

We now recall the $U^{\not\perp}$ transform from [10]. It converts a dPKE $\mathsf{P} = (\mathsf{KeyGen}_{\mathsf{P}}, \mathsf{Encr}, \mathsf{Decr})$ into a KEM $\mathsf{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ using a pseudorandom function $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times \mathcal{C} \to \mathcal{K}$ and a hash function $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ for given key spaces $\mathcal{K}_{\mathsf{F}}$ and $\mathcal{K}$. Here $\mathcal{M}$ and $\mathcal{C}$ denote the message and cipher spaces of P. The PRF is used in case the ciphertext happens to be invalid. The transform is defined by the following three algorithms:

- $\mathsf{KeyGen}(1^\lambda)$. On input a security parameter λ, this algorithm runs $(\mathsf{pk}, \mathsf{sk}_{\mathsf{P}}) \leftarrow \mathsf{KeyGen}_{\mathsf{P}}(1^\lambda)$, samples a random key $\mathsf{prfk} \overset{\$}{\leftarrow} \mathcal{K}_{\mathsf{F}}$ and sets $\mathsf{sk} = (\mathsf{sk}_{\mathsf{P}}, \mathsf{prfk})$. The algorithm returns a pair of public and secret keys $(\mathsf{pk}, \mathsf{sk})$.

- $\mathsf{Encaps}(\mathsf{pk})$. On input a public key pk, this algorithm samples a random message $m \xleftarrow{\$} \mathcal{M}$, encrypts it running the encryption algorithm of P, i.e., $c \leftarrow \mathsf{Encr}(\mathsf{pk}, m)$, and computes a hash value $\mathsf{k} \leftarrow H(m, c)$. It outputs (k, c).
- $\mathsf{Decaps}(\mathsf{sk}, c)$. This algorithm parses sk as $\mathsf{sk} = (\mathsf{sk_P}, \mathsf{prfk})$ and runs the decryption algorithm of P to decrypt c, i.e., $m' \leftarrow \mathsf{Decr}(\mathsf{sk_P}, c)$. If $m' = \perp$, then it returns $\mathsf{F}(\mathsf{prfk}, c)$. If $m' \neq \perp$ but $\mathsf{Encr}(\mathsf{pk}, m') \neq c$, then it also returns $\mathsf{F}(\mathsf{prfk}, c)$. In all other cases (i.e., if $m' \neq \perp$ and $\mathsf{Encr}(\mathsf{pk}, m') = c$), it returns $H(m', c)$.

4.3 Analysis of the $U^{\not\perp}$ Transform

We are now ready to state our main application of the O2H lemma from Section 3. In the following theorem, we state that $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ is an $\mathsf{IND\text{-}CCA}$ secure KEM as long as the following four conditions are satisfied: (i) the dPKE scheme P is $\mathsf{OW\text{-}CPA}$ secure, (ii) it is η-injective for a negligible η, (iii) it is FFC secure, and (iv) F is a secure PRF. The latter is as in prior works: the improvement is in the security loss.

Theorem 4.6. *Let $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ be a quantum-accessible random oracle, $\mathsf{F} : \mathcal{K}_\mathsf{F} \times \mathcal{C} \to \mathcal{K}$ be a PRF and P be an η-injective dPKE which does not depend of H. Let $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ be the KEM obtained by applying the $U^{\not\perp}$ transform to P, F and H. Let $\mathcal{A}$ be an adversary against the $\mathsf{IND\text{-}CCA}$ security of $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$ issuing at most q (quantum oracle) queries to H with query depth at most d, and q_{dec} classical queries to the decapsulation oracle.*

Then, we can construct three algorithms whose run-times are $\lesssim 3\mathcal{T}_\mathcal{A}$. These algorithms are:

- *a $\mathsf{OW\text{-}CPA}$-adversary $\mathcal{B}_1$ against P,*
- *an FFC-adversary $\mathcal{B}_2$ against P, returning a list of at most q_{dec} ciphertexts,*
- *a PRF-adversary $\mathcal{B}_3$ against F making q_{dec} queries.*

These algorithms satisfy the following:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{U^{\not\perp}(\mathsf{P},\mathsf{F},H)}(\mathcal{A}) \leq 4d \cdot \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{P}}(\mathcal{B}_1) + 6\mathsf{Adv}^{\mathsf{FFC}}_{\mathsf{P}}(\mathcal{B}_2) + 2\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{F}}(\mathcal{B}_3)\\ + (4d + 6) \cdot \eta.\end{aligned}$$

Proof. Our proof uses a sequence of games. All six games in our proof are essentially the same as in the proof of [5, Theorem 2], the only difference being the analysis of Game 5 to apply our new O2H lemma instead of the O2H lemma from [5]. For the sake of completeness, we present all the games.

In each of the following games, the probability space is partitioned into three mutually exclusive classical outcomes (events) called Win, Lose and Draw, respectively corresponding to $\mathcal{A}$ succeeding in its IND-CCA attack ($b' = b$), failing ($b' \neq b$) and a kind of intermediate outcome between the two, defined precisely in Game 2. Outcome Draw is defined to have probability 0 in Games 0 and 1, but in later games, whenever Draw occurs, the game continues and returns a Draw in the end regardless of b and b'. In Game i (for $i \in \{0, \ldots, 5\}$), we define the attacker's 'score' w_i as

$$\begin{aligned} w_i := &\Pr[\mathsf{Win} : \mathsf{Game}\ i] + \frac{1}{2}\Pr[\mathsf{Draw} : \mathsf{Game}\ i] \\ = &\frac{1}{2} + \frac{1}{2}\left(\Pr[\mathsf{Win} : \mathsf{Game}\ i] - \Pr[\mathsf{Lose} : \mathsf{Game}\ i]\right), \end{aligned}$$

where the last equality comes from the fact that Win, Lose and Draw partition the probability space in each game.

Game 0 (IND-CCA). This game is the original IND-CCA experiment against $U^{\not\perp}(\mathsf{P}, \mathsf{F}, H)$.

Game 1 (PRF is random). This game is the same as Game 0, except that the simulator replaces the PRF $\mathsf{F}(\mathsf{prfk}, \cdot)$ in the decapsulation algorithm by a random function $\mathsf{R} \xleftarrow{\$} \mathcal{K}^{\mathcal{C}}$. We construct a PRF adversary $\mathcal{B}_3$ by replacing calls to $\mathsf{F}(\mathsf{prfk}, \cdot)$ by calls to $\mathcal{B}_3$'s oracle. Adversary $\mathcal{B}_3$ runs $\mathcal{A}$ and outputs 1 if $\mathcal{A}$ wins the IND-CCA game and 0 otherwise. If $\mathcal{B}_3$'s oracle is F, then it simulates Game 0, and if $\mathcal{B}_3$'s oracle is R, then it simulates Game 1. Therefore, we have $\Pr[\mathcal{B}_3^{\mathsf{F}(\mathsf{k},\cdot)} = 1] = \Pr[\mathsf{Win} : \mathsf{Game}\ 0]$ and $\Pr[\mathcal{B}_3^{\mathsf{R}(\cdot)} = 1] = \Pr[\mathsf{Win} : \mathsf{Game}\ 1]$, and hence

$$|w_1 - w_0| = \mathsf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3).$$

Game 2 (Draw on fail). We let Fail be the (classical) event that at least one query of $\mathcal{A}$ to the decapsulation oracle $\mathcal{O}^D$ fails to decrypt a valid ciphertext., i.e., adversary $\mathcal{A}$ queries a c such that there exists some message $m \in \mathcal{M}$ such that $c = \mathsf{Encr}(\mathsf{pk}, m)$, but with $\mathsf{Decr}(\mathsf{sk}, c) \neq m$. We also let Inj denote the (classical) event that the encryption mapping $\mathsf{Encr}(\mathsf{pk}, \cdot)$ is injective over the message space $\mathcal{M}$. In Game 2 and the subsequent games, we define the Draw event as $\mathsf{Draw} := \mathsf{Fail} \vee \neg\mathsf{Inj}$ (which implies $\neg\mathsf{Draw} := \neg\mathsf{Fail} \wedge \mathsf{Inj}$). We define $d_i := \Pr[\mathsf{Draw} : \mathsf{Game}\ i]$, for $i \geq 2$. For $i < 2$, we define Draw as the empty event and $d_i = 0$.

We have:

$$|w_2 - w_1| = \left| \Pr[\mathsf{Win} : \ \mathsf{Game}\ 2] - \Pr[\mathsf{Win} : \ \mathsf{Game}\ 1] + \frac{d_2}{2} \right| \leq \frac{d_2}{2},$$

where the first equality holds since $d_1 = 0$ and the inequality holds true as $-d_2 \leq \Pr[\mathsf{Win} : \ \mathsf{Game}\ 2] - \Pr[\mathsf{Win} : \ \mathsf{Game}\ 1] \leq 0$. Note that the simulator may not be able to efficiently check whether Draw occurs, but the games will not require the simulator to perform this check.

Game 3 (Reprogram $H(m, c)$ to $\mathsf{R}(c)$). This game differs from Game 2 by reprogramming the hash function return value $H(m, c)$ on input (m, c) to $\mathsf{R}(c)$ if $c = \mathsf{Encr}(\mathsf{pk}, m)$.

The change from Game 2 to Game 3 does not affect the probability of Win and Draw so that $w_3 = w_2$ and $d_3 = d_2$. This is because in Game 3, the joint distribution of the oracle H and the attacker's view remains the same as in Game 2, as long as Draw does not occur. In particular, the distribution of $H(m, c)$ for each (m, c) remains uniformly random thanks to the uniformly random choice of $\mathsf{R}(c)$. The $H(m, c)$ values also remain independent for distinct pairs $(m, c) \neq (m', c')$ if Draw does not occur, since the latter implies that Inj occurs (i.e., there do not exist two distinct messages $m \neq m'$ with $c = \mathsf{Encr}(\mathsf{pk}, m) = \mathsf{Encr}(\mathsf{pk}, m') = c'$) Also, if Draw does not occur, then for any ciphertext c queried to and failing decryption by the Decaps oracle (meaning that $\mathsf{Encr}(\mathsf{pk}, \mathsf{Decr}(sk, c)) \neq c$), the Decaps oracle returns a value $\mathsf{R}(c)$ that is statistically independent of the value of $H(m, c)$ for all messages $m \in \mathcal{M}$ (since if there would exist some m with $H(m, c) = R(c)$, i.e., $\mathsf{Encr}(\mathsf{pk}, m) = c$, it would imply that c is a valid ciphertext which failed to decrypt in Decaps, so that Draw occurred).

Game 4 (Decapsulation oracle returns $\mathsf{R}(c)$). This game is the same as Game 3 except that Decaps is modified to output $\mathsf{R}(c)$ for all ciphertexts but the challenge ciphertext (for the challenge ciphertext, it still outputs $\perp$). Since in Game 3, Decaps already responds in this way (as both F and H have been reprogrammed to respond with $\mathsf{R}(c)$), the values of w_4, d_4 are not affected, i.e., $w_4 = w_3$ and $d_4 = d_3$. The only change is that in Game 4 and onwards, the secret key is not used anymore in the simulation. We conclude that all probabilities d_i of Draw in Games 2 to 4 are the same.

To bound this Draw probability, we construct an adversary $\mathcal{B}_2$ which, given a public key pk, simulates Game 4 with $\mathcal{A}$, and outputs the list L of $\mathcal{A}$'s decapsulation queries. Note that if the event Fail occurs, then L contains a valid ciphertext c that fails decryption by Decr. Therefore, according to Definition 4.2, algorithm $\mathcal{B}_2$ is an FFC adversary against P which runs in almost the same time as $\mathcal{A}$ and has FFC advantage

$$\begin{aligned}\mathsf{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) &= \Pr[\mathsf{Fail} : \mathsf{Game}\ 4] \\ &\geq \Pr[\mathsf{Draw} : \mathsf{Game}\ 4] - \Pr[\neg\mathsf{Inj} : \mathsf{Game}\ 4] \\ &= d_4 - \eta,\end{aligned}$$

using the fact that P is η-injective. We conclude that

$$d_2 = d_3 = d_4 \leq \mathsf{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + \eta. \tag{19}$$

Game 5 (Change shared secret). This game differs from Game 4 by changing the challenge shared secret k_b^* given to $\mathcal{A}$ to always be an independent uniformly random value r (whereas in Game 4, the challenge shared secret k_b^* was chosen as an independent random value $r = \mathsf{k}_1^*$ if $b = 1$ but chosen as $\mathsf{R}(c^*)$ if $b = 0$). Additionally, if $b = 0$ then $\mathsf{R}(c^*)$ is reprogrammed to return r (i.e., $H(m, c^*) = r$ for all messages m such that $\mathsf{Encr}(\mathsf{pk}, m) = c^*$; we denote by S^* the set of such messages m), but if $b = 1$ then $\mathsf{R}(c^*)$ is not reprogrammed.

In fact, the change from Game 4 to Game 5 is purely conceptual and does not change the joint distribution of the view of $\mathcal{A}$. Indeed, in both games, if $b = 0$, the input shared key k_0^* to $\mathcal{A}$ is uniformly random and equal to $H(m, c^*) = \mathsf{R}(c^*)$ for all $m \in S^*$. And in both games, if $b = 1$, the input shared key k_1^* to $\mathcal{A}$ is uniformly random and statistically independent of the uniformly random value of $H(m, c^*) = \mathsf{R}(c^*)$ for all $m \in S^*$. Therefore, we have $w_5 = w_4$ and $d_5 = d_4$.

In Game 5, the distribution of the input $z = (\mathsf{pk}, c^*, \mathsf{k}_b^* = r)$ to $\mathcal{A}$ is independent of b, and the random oracle queried by $\mathcal{A}$ and the simulator is either H if $b = 1$ (where $H(m, c) = \mathsf{R}(c)$ if $\mathsf{Encr}(\mathsf{pk}, m) = c$) or H' if $b = 0$, where H' is equal to H on all inputs except those in the set $S := \{(m, c^*) : m \in S^*\}$; for inputs in S, H' returns r. The simulation in Game 5 runs in time $\approx \mathcal{T}_{\mathcal{A}}$. Therefore, the algorithm $\mathcal{A}$ together with the simulator in Game 5 constitutes an O2H distinguisher algorithm for distinguishing oracle H from H' with run-time $\approx \mathcal{T}_{\mathcal{A}}$. Therefore, applying Lemma 3.3, we can construct algorithm $\mathcal{D}$, with run-time $\lesssim 3\mathcal{T}_{\mathcal{A}}$ and making oracle calls to H' and H, such that

$$\begin{aligned}\Delta &:= |\Pr[0 \leftarrow \mathcal{A} : b = 0] - \Pr[0 \leftarrow \mathcal{A} : b = 1]| \\ &= \left|\Pr[0 \leftarrow \mathcal{A}^{H'}] - \Pr[0 \leftarrow \mathcal{A}^{H}]\right| \\ &\leq 4d \cdot \Pr[T \cap S \neq \emptyset : T \xleftarrow{\$} \mathcal{D}^{H',H}(z)]. \qquad (20)\end{aligned}$$

Using $\mathcal{D}$, we can construct an algorithm $\mathcal{B}_1$ against the $\mathsf{OW\text{-}CPA}$ security of P that given (pk, c^*, r), runs $\mathcal{D}^{H',H}$ and when $\mathcal{D}$ returns its output set T of candidates for m^*, algorithm $\mathcal{B}_1$ tests each $m \in T$ to check whether $m \in S$, i.e., whether $\mathsf{Encr}(\mathsf{pk}, m) = c^*$, and returns any such m if it is found. Note that $\mathcal{T}_{\mathcal{B}_1} \approx \mathcal{T}_{\mathcal{D}}$. Further, algorithm $\mathcal{B}_1$ succeeds (i.e., outputs m^*) if $T \cap S \neq \emptyset$, unless $\neg\mathsf{Inj}$ occurs (in the latter case, the output of $\mathcal{B}_1$ may be a different decryption of c^* than m^*). Since P is η-injective, we have

$$\Pr[T \cap S \neq \emptyset : T \xleftarrow{\$} \mathcal{D}^{H',H}(\mathsf{pk}, c^*)] \leq \mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1) + \eta. \qquad (21)$$

On the other hand, in Game 5 we have:

$$\begin{aligned}2\left|w_5 - \frac{1}{2}\right| &= |\Pr[\mathsf{Win} : \text{Game 5}] - \Pr[\mathsf{Lose} : \text{Game 5}]| \\ &= \left|\frac{1}{2}\Pr[0 \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 0] + \frac{1}{2}\Pr[0 \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 1]\right. \\ &\quad \left.-\frac{1}{2}\Pr[0 \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 1] - \frac{1}{2}\Pr[1 \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 0]\right| \\ &\leq \frac{1}{2}|\Delta_{0,\neg\mathsf{Draw}}| + \frac{1}{2}|\Delta_{1,\neg\mathsf{Draw}}|, \qquad (22)\end{aligned}$$

where we define, for $v \in \{0, 1\}$,

$$\Delta_{v,\neg\mathsf{Draw}} := \Pr[v \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 0] - \Pr[v \leftarrow \mathcal{A} \wedge \neg\mathsf{Draw} : b = 1].$$

We further define:

$$\Delta_{v,\mathsf{Draw}} := \Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} : b = 0] - \Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} : b = 1],$$

which satisfies

$$\begin{aligned}|\Delta_{v,\mathsf{Draw}}| &\leq \Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} : b = 0] + \Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} : b = 1] \\ &= 2 \cdot (\Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} \wedge b = 0] + \Pr[v \leftarrow \mathcal{A} \wedge \mathsf{Draw} \wedge b = 1]) \\ &\leq 4 \cdot \Pr[\mathsf{Draw}] = 4 \cdot d_5. \qquad (23)\end{aligned}$$

Now, for $v \in \{0,1\}$, observe that $\mathit{\Delta}_{v,\neg\mathsf{Draw}} + \mathit{\Delta}_{v,\mathsf{Draw}} = \mathit{\Delta}$, so we have, by the triangle inequality, (23), (20) and (21):

$$\begin{aligned} \mathit{\Delta}_{v,\neg\mathsf{Draw}} &\le |\mathit{\Delta}| + |\mathit{\Delta}_{v,\mathsf{Draw}}| \\ &\le 4d \cdot \Big(\mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1) + \eta\Big) + 4d_5. \end{aligned} \tag{24}$$

and plugging (24) into (22) for $v \in \{0,1\}$ gives

$$\left| w_5 - \frac{1}{2} \right| \le 2d \cdot \Big(\mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1) + \eta\Big) + 2d_5.$$

Summing up the differences of w_i's over all games, we get

$$\begin{aligned} \mathsf{Adv}_{\mathsf{U}^{\not\perp}(\mathsf{P},\mathsf{F},H)}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) &= 2|w_0 - 1/2| \\ &\le 4d \cdot \Big(\mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1) + \eta\Big) + 4d_5 + 2d_2 + 2\mathsf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3) \\ &\le 4d \cdot \mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_1) + 6\mathsf{Adv}_{\mathsf{P}}^{\mathsf{FFC}}(\mathcal{B}_2) + 2\mathsf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{B}_3) \\ &\quad + (4d+6) \cdot \eta, \end{aligned}$$

where in the last line we plugged in the bound on $d_5 = d_2$ from (19). □

Combining Theorem 4.6 with Theorem 4.4 and Lemma 4.5, we immediately obtain the following result for the IND-CCA security of the FO-transformed scheme $\mathsf{FO}^{\not\perp}(\mathsf{P},\mathsf{F},G,H) = \mathsf{U}^{\not\perp}(T(\mathsf{P},G),\mathsf{F},H)$ from the IND-CPA security of scheme P.

Corollary 4.7. *Let* P *be a δ-correct* rPKE *with message space $\mathcal{M}$ and randomness space $\mathcal{R}$. Let $G : \mathcal{M} \to \mathcal{R}$ and $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ be quantum-accessible random oracles, and $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times \mathcal{C} \to \mathcal{K}$ be a* PRF*. Suppose that $\mathsf{P}' = T(\mathsf{P},G)$ is η-injective and let $\mathsf{FO}^{\not\perp}(\mathsf{P},\mathsf{F},G,H) = \mathsf{U}^{\not\perp}(T(\mathsf{P},G),\mathsf{F},H)$. Let $\mathcal{A}$ be an adversary against the* IND-CCA *security of $\mathsf{FO}^{\not\perp}(\mathsf{P},\mathsf{F},G,H)$ issuing at most q_G (resp. q_H) quantum queries to G (resp. H) with query depth at most d_G (resp. d_H) and at most q_{dec} classical queries to the decapsulation oracle of $\mathsf{FO}^{\not\perp}(\mathsf{P},\mathsf{F},G,H)$.*

Then, we can construct two algorithms whose run-times are $\lesssim 3\mathcal{T}_{\mathcal{A}}$. These algorithms are:

- *an* IND-CPA*-adversary $\mathcal{B}_1$ against* P*,*
- *a* PRF*-adversary $\mathcal{B}_2$ against* F *issuing at most q_{dec} queries.*

These algorithms satisfy the following:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{FO}^{\not\perp}(\mathsf{P},\mathsf{F},G,H)}(\mathcal{A}) \leq 8d_H \cdot (d_G+1) \cdot &\left(\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{P}}(\mathcal{B}_1) + \frac{8\cdot(3q_G+1)}{|\mathcal{M}|}\right)\\ &+ 6\cdot(3q_G+q_{dec})\cdot\left((8d_G+1)\cdot\delta+\sqrt{3\eta}\right)\\ &+ (4d_H+12)\cdot\eta + 2\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{F}}(\mathcal{B}_2).\end{aligned}$$

Acknowledgments. This work was supported in part by BPI-France in the context of the national project RISQ (P141580), by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701) and the Australian Research Council Discovery Grant DP180102199. Part of this work was done while Damien Stehlé was visiting the Simons Institute for the Theory of Computing.

References

1. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *Advances in Cryptology - CRYPTO 2019*, pages 269–295, 2019.
2. A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 474–483, 2014.
3. D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, and W. Wang. Classic McEliece – supporting documentation, 2019. Submitted to [18], available at `https://classic.mceliece.org/nist/mceliece-20190331.pdf`.
4. D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal. NTRU Prime: Round 2 – supporting documentation, 2019. Submitted to [18], available at `https://ntruprime.cr.yp.to/nist/ntruprime-20190330.pdf`.
5. N. Bindel, M. Hamburg, K. Hövelmanns, A. Hülsing, and E. Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *Theory of Cryptography - 17th International Conference, TCC 2019*, pages 61–90, 2019.
6. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *Advances in Cryptology - ASIACRYPT 2011*, pages 41–69, 2011.
7. C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, J. M. Schanck, P. Schwabe, W. Whyte, and Z. Zhang. NTRU – supporting documentation, 2019. Submitted to [18], available at `https://ntru.org/f/ntru-20190330.pdf`.
8. A. W. Dent. A designer's guide to KEMs. In *Cryptography and Coding, 9th IMA International Conference*, pages 133–151, 2003.
9. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO*, pages 537–554, 1999.
10. D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *Theory of Cryptography - 15th International Conference, TCC 2017*, pages 341–371, 2017.

11. K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. *IACR Cryptology ePrint Archive*, 2018:928, 2018.
12. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *Advances in Cryptology - CRYPTO 2018*, pages 96–125, 2018.
13. H. Jiang, Z. Zhang, and Z. Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *Public-Key Cryptography - PKC 2019*, pages 618–645, 2019.
14. H. Jiang, Z. Zhang, and Z. Ma. On the non-tightness of measurement-based reductions for key encapsulation mechanism in the quantum random oracle model. *IACR Cryptology ePrint Archive*, 2019:494, 2019.
15. H. Jiang, Z. Zhang, and Z. Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 227–248, 2019.
16. D. Micciancio and M. Walter. On the bit security of cryptographic primitives. In *Advances in Cryptology - EUROCRYPT 2018*, pages 3–28, 2018.
17. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.
18. NIST. Post-quantum cryptography standardization. Available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/`.
19. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th Annual ACM Symposium on Theory of Computing*, pages 84–93, 2005.
20. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Advances in Cryptology - EUROCRYPT 2018*, pages 520–551, 2018.
21. E. E. Targhi and D. Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *14th International Conference on Theory of Cryptography TCC 2016-B*, pages 192–216, 2016.
22. D. Unruh. Quantum proofs of knowledge. In *Advances in Cryptology - EUROCRYPT 2012*, pages 135–152, 2012.
23. D. Unruh. Revocable quantum timed-release encryption. In *Advances in Cryptology - EUROCRYPT 2014*, pages 129–146, 2014.
24. J. Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.
25. K. Xagawa and T. Yamakawa. (Tightly) QCCA-secure key-encapsulation mechanism in the quantum random oracle model. In *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 249–268, 2019.
26. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology - CRYPTO 2019*, pages 239–268, 2019.

A Standard Cryptographic Definitions

Definition A.1 (Public-Key Encryption). *Given a finite message space $\mathcal{M}$, a cipher space $\mathcal{C}$, a secret key space $\mathcal{SK}$ and a public-key space $\mathcal{PK}$, a randomized public-key encryption scheme* (rPKE) *consists of three algorithms* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$*:*

- $\mathsf{KeyGen}(1^\lambda)$*: On input a security parameter* λ*, this randomized algorithm outputs a secret key* $\mathsf{sk} \in \mathcal{SK}$ *and a public key* $\mathsf{pk} \in \mathcal{PK}$*.*
- $\mathsf{Encr}(\mathsf{pk}, m)$*: On input a public key* pk *and a message* m*, this randomized algorithm outputs a ciphertext* $c \in \mathcal{C}$*.*
- $\mathsf{Decr}(\mathsf{sk}, c)$*: On input a secret key* $\mathsf{sk} \in \mathcal{SK}$ *and a ciphertext* $c \in \mathcal{C}$*, this deterministic algorithm outputs either a message* $m' \in \mathcal{M}$ *or a failure* $\perp \notin \mathcal{M}$*.*

The definition of a deterministic public-key encryption scheme (dPKE) *is the same except that* Encr *is a deterministic algorithm.*

Definition A.2 (Correctness of PKEs). *We say that a* PKE $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *is* δ*-correct if the following holds:*

$$\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{Decr}(\mathsf{sk}, \mathsf{Encr}(\mathsf{pk}, m)) \neq m] : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\right] \leq \delta.$$

In case of a deterministic PKE, the probability $\Pr[\mathsf{Decr}(\mathsf{sk}, \mathsf{Encr}(\mathsf{pk}, m) \neq m]$ is either 0 or 1 for each key pair $(\mathsf{pk}, \mathsf{sk})$.

We provide definitions for the two security properties of a PKE scheme that are relevant for this work. They match the corresponding definitions in [5]. We use notation $\mathcal{H}$ to denote a space of random hash functions. If security of a PKE scheme is given in the random oracle model, we sample a random hash function from this space, i.e., $H \xleftarrow{\$} \mathcal{H}$.

Definition A.3 (OW-CPA Advantage). *Let* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *be a* dPKE *or an* rPKE*, and* $\mathcal{A}$ *be an adversary executing a one-way chosen-plaintext attack as specified by the following experiment:*

```
$\mathsf{Expt}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}$

$H \xleftarrow{\$} \mathcal{H}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$m^* \xleftarrow{\$} \mathcal{M}$
$c^* \leftarrow \mathsf{Encr}(\mathsf{pk}, m^*)$
$m' \leftarrow \mathcal{A}^H(\mathsf{pk}, c^*)$
$\textit{return } [m^* = m']$
```

The advantage of $\mathcal{A}$ *winning the above experiment is defined as:*

$$\mathsf{Adv}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A}) := \Pr[1 \leftarrow \mathsf{Expt}_{\mathsf{P}}^{\mathsf{OW\text{-}CPA}}(\mathcal{A})].$$

Definition A.4 (IND-CPA Advantage). *Let* $\mathsf{P} = (\mathsf{KeyGen}, \mathsf{Encr}, \mathsf{Decr})$ *be a* dPKE *or an* rPKE*, and* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be an adversary executing an indistiguishability chosen-plaintext attack as specified by the following experiment:*

$\underline{\mathsf{Expt}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}}$

```
$H \xleftarrow{\$} \mathcal{H}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(st, m_0, m_1) \leftarrow \mathcal{A}_1^H(\mathsf{pk})$
$b \xleftarrow{\$} \{0, 1\}$
$c^* \leftarrow \mathsf{Encr}(\mathsf{pk}, m_b^*)$
$b' \leftarrow \mathcal{A}_2^H(\mathsf{pk}, m_0, m_1, c^*, st)$
return $[b = b']$
```

The advantage of $\mathcal{A}$ *winning the above experiment is defined as:*

$$\mathsf{Adv}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) := 2 \left| \Pr[1 \leftarrow \mathsf{Expt}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}(\mathcal{A})] - \frac{1}{2} \right| .$$

We finally recall the definition of a Key Encapsulation Mechanism (KEM) and the corresponding security property called indistinguishability against chosen-ciphertext-attacks.

Definition A.5 (Key Encapsulation Mechanism). *Given a message space* $\mathcal{M}$*, a public-key space* $\mathcal{PK}$*, a secret-key space* $\mathcal{SK}$ *and a key space* $\mathcal{K}$*, a* KEM K *consists of three algorithms* $\mathsf{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$*:*

- $\mathsf{KeyGen}(1^\lambda)$*: On input a security parameter* λ*, this randomized algorithm outputs a secret key* $\mathsf{sk} \in \mathcal{SK}$ *and a public key* $\mathsf{pk} \in \mathcal{PK}$*.*
- $\mathsf{Encaps}(\mathsf{pk})$*: On input a public key* pk*, this randomized algorithm outputs a ciphertext* $c \in \mathcal{C}$ *and a key* $\mathsf{k} \in \mathcal{K}$*.*
- $\mathsf{Decaps}(\mathsf{sk}, c)$*: On input a secret key* $\mathsf{sk} \in \mathcal{SK}$ *and a ciphertext* $c \in \mathcal{C}$*, this deterministic algorithm outputs either a key* $\mathsf{k} \in \mathcal{K}$ *or a failure* $\perp \notin \mathcal{K}$*.*

Definition A.6 (IND-CCA Advantage). *Let* $\mathsf{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ *be a* KEM *and* $\mathcal{A}$ *be an adversary attacking* K *and given access to a random oracle* H*. Adversary* $\mathcal{A}$ *executes a indistinguishability chosen-ciphertext attack as specified by the following experiment:*

```
$\mathsf{Expt}_{\mathsf{P}}^{\mathsf{IND\text{-}CPA}}$

$H \xleftarrow{\$} \mathcal{H}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
$(c^*, \mathsf{k}_0^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$
$\mathsf{k}_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0, 1\}$
$b' \leftarrow \mathcal{A}^{H, \mathcal{O}^D}(\mathsf{pk}, c^*, \mathsf{k}_b^*)$
$return\ [b = b']$
```

Adversary $\mathcal{A}$ is given (classical) access to the decapsulation oracle $\mathcal{O}^D$ working as follows: on input of a query c, it checks if $c = c^*$; if so, it outputs $\perp$; otherwise, it outputs $\mathsf{Decaps}(\mathsf{sk}, c)$*.*

The advantage of $\mathcal{A}$ winning the above experiment is defined as:

$$\mathsf{Adv}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) := 2 \left| \Pr[1 \leftarrow \mathsf{Expt}_{\mathsf{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})] - \frac{1}{2} \right| .$$

Definition A.7 (PRF Advantage). *Let $\mathcal{K}_{\mathsf{F}}, X, Y$ be finite sets and $\mathsf{F} : \mathcal{K}_{\mathsf{F}} \times X \to Y$ be a pseudorandom function* (PRF)*. Then the* PRF*-advantage of an adversary $\mathcal{A}$ is defined as*

$$\mathsf{Adv}_{\mathsf{F}}^{\mathsf{PRF}}(\mathcal{A}) = \left| \Pr_{\mathsf{k} \xleftarrow{\$} \mathcal{K}} [\mathcal{A}^{\mathsf{F}(\mathsf{k}, \cdot)} = 1] - \Pr_{\mathsf{R} \xleftarrow{\$} Y^X} [\mathcal{A}^{\mathsf{R}(\cdot)} = 1] \right| ,$$

where Y^X denotes all the functions from X to Y.

B Security Loss Computation Details

For completeness, we provide the detailed computations leading our security loss bounds in Table 2.

We recall from Section 1 that we defined the security loss L of the security reduction for an FO CCA scheme with bit security S_{cca} as $L := S_{weak} - \lambda$, where S_{weak} is the smallest bit security for the FO-transformed 'weak' scheme so that the CCA security reduction implies $S_{cca} \geq \lambda$ for a security parameter λ.

For the first row of Table 2, the proved CCA advantage bound is $\varepsilon_{cca} \leq q^{3/2} \varepsilon_{cpa}^{1/4}$, where q is the number of attacker random-oracle queries and ε_{cpa} is the CPA advantage against the weak scheme, where both these advantages are for attack run-time T, which is (approx.) the same for both CPA and CCA attacks. By definition of bit-securities in the CCA and

CPA cases, they are $S_{cca} := T/\varepsilon_{cca}^2$ and $S_{cpa} := T/\varepsilon_{cpa}^2$, respectively. We re-write the CCA advantage bound as $T/\varepsilon_{cca} \geq T/(q^{3/2}\varepsilon_{cpa}^{1/4})$. To get ε_{cpa}^2 in the denominator on the right, we raise both sides to the 8th power. The left-hand side becomes $(T/\varepsilon_{cca}^2)^4 \cdot T^4 = S_{cca}^4 \cdot T^4$ while the right-hand side becomes $(T/\varepsilon_{cpa}^2) \cdot T^7/q^{12} = S_{cpa} \cdot T^7/q^{12}$. Rearranging gives $S_{cca} \geq (S_{cpa} \cdot T^3/q^{12})^{1/4}$. To guarantee $S_{cca} \geq 2^\lambda$ through this inequality, we need $S_{cpa} \geq q^{12}/T^3 \cdot 2^{4\lambda}$. As $q \leq T$, the worst-case is $q = T$ which gives $S_{cpa} \geq q^9 \cdot 2^{4\lambda}$, so the overhead is $S_{cpa}/2^\lambda \geq q^9 \cdot 2^{3\lambda}$, i.e. $L = 3\lambda + 9\log q$.

For the two middle rows of Table 2, the CCA advantage bound is $\varepsilon_{cca} \leq d^{1/2}\varepsilon_{cpa}^{1/2}$. Rearranging and raising to the 4th power leads to the inequality $(T/\varepsilon_{cca}^2)^2 \cdot T^2 = S_{cca}^2 \cdot T^2 \geq (T/\varepsilon_{cpa}^2) \cdot T^3/d^2$ and therefore, $S_{cca}^2 \geq S_{cpa} \cdot T/d^2$. To guarantee $S_{cca} \geq 2^\lambda$ through this inequality, we need $S_{cpa} \geq d^2/T \cdot 2^{2\lambda}$. As $T \geq d$, we get $S_{cpa} \geq d \cdot 2^{2\lambda}$, hence a loss of $L = \lambda + \log d$.

For the last row of Table 2, the CCA advantage bound is $\varepsilon_{cca} \leq d^2\varepsilon_{cpa}$. Rearranging gives $T/\varepsilon_{cca}^2 \geq (T/\varepsilon_{cpa}^2) \cdot 1/d^4$. To guarantee $S_{cca} \geq 2^\lambda$ via this inequality, we need $S_{cpa} \geq d^4 \cdot 2^\lambda$, hence a loss of $L = 4\log d$.