cryptography

MDPI

Article

Tightly Secure PKE Combiner in the Quantum Random Oracle Model

Brian Goncalves *,† **and Atefeh Mashatan** †

Cybersecurity Research Lab, Ryerson University, Toronto, ON M5B 2K3, Canada; amashatan@ryerson.ca
* Correspondence: bgoncalves@ryerson.ca
† These authors contributed equally to this work.

Abstract: The development of increasingly sophisticated quantum computers poses a long-term threat to current cryptographic infrastructure. This has spurred research into both quantum-resistant algorithms and how to safely transition real-world implementations and protocols to quantum-resistant replacements. This transition is likely to be a gradual process due to both the complexity and cost associated with transitioning. One method to ease the transition is the use of classical–quantum hybrid schemes, which provide security against both classical and quantum adversaries. We present a new combiner for creating hybrid encryption schemes directly from traditional encryption schemes. Our construction is the only existing proposal in the literature with IND-CCA-security in the classical and quantum random oracle models, respectively.

Keywords: public-key cryptography; hybrid encryption; quantum-resistance; combiners; PKEs

check for updates

Citation: Goncalves, B.; Mashatan, A. Tightly Secure PKE Combiner in the Quantum Random Oracle Model. *Cryptography* **2022**, *6*, 15. https://doi.org/10.3390/cryptography6020015

Academic Editors: Emmanouil Magkos and Christoforos Ntantogian

Received: 8 March 2022
Accepted: 26 March 2022
Published: 29 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

As the age of scalable quantum computing comes closer to becoming a reality, research into cryptographic algorithms that are secure against quantum adversaries becomes increasingly important. The transition to new algorithms for real-world applications and protocols has historically been slow. It took nearly two decades [1] for current public-key cryptographic infrastructure to be fully deployed; hence, new post-quantum infrastructure can be expected to take a similar amount of time. Additionally, maintaining legacy systems and addressing concerns of potential downgrade attacks should be considered when the transition begins. Finally, while quantum-resistant schemes have been increasingly studied and analyzed, the underlying hardness assumptions remain relatively novel, in which future cryptanalysis may show that they are vulnerable to quantum attacks or, even worse, classical attacks. As the release of standards is still years away and full deployment of quantum-resistant algorithms is a long-term effort, there is still a need to secure today's data from quantum adversaries.

Hybrid Cryptography and Combiners. So-called classical–quantum hybrid cryptography, which combines classically secure and quantum-resistant algorithms to produce a new secure scheme, represents a stopgap solution to the dilemma of transitioning from classical to quantum-resistant cryptographic infrastructure and the need to secure data and communications versus the cost, and the time, to fully transition. Harnik et al. [2] formalized this idea of combining algorithms as a (k, n)-robust combiner, where n represents the number of inputs and k represents the threshold of secure inputs required to achieve security. Hybrid combiners have been the subject of previous works for various primitives, such as Bindel et al. [3] on hybrid signatures and Bindel et al. [4] on hybrid key encapsulation mechanisms (KEMs) and hybrid authenticated key exchange.

Traditional combiners for public-key encryption (PKE), the focus of this work, have also been studied by Asmuth and Blakely [5], Herzberg [6], and Zhang et al. [7]. However, these previous results either failed to achieve IND-CCA-security [6], required every input to be secure [8], or lacked a proof against quantum adversaries [9].

In the place of PKE combiners, there has been successful work in the field of KEM combiners. Giacon et al. presented a swath of different construction of robust IND-CCA combiners for KEMs using different *core function* constructions to produce the final key [10]. However, the results are limited to only KEMs with perfect correctness, which many quantum-resistant schemes do not have. Bindel et al. additionally provided three more constructions of robust KEM combiners against adversaries with varying levels of quantum capabilities over time [4].

Currently, there is interest in both academia and industry for hybrid cryptography. As mentioned, both Bindel et al. [3], and Bindel et al. [4] constructed hybrid primitives, digital signatures, and KEMs, respectively, with the latter also presenting a design for a hybrid authenticated key exchange built from hybrid KEMs. Mirroring the interest in hybrid cryptography, ISARA and Cisco investigated the viability of hybrid X.509 digital certificates [11] in 2018. Meanwhile, in 2016, Google experimented with combining elliptic curve Diffie–Hellman (ECDH) and ring-learning with errors (R-LWE) key exchange in the transport layer security (TLS) stack in a test version of its Chrome web browser [12,13].

Our Contributions. In this work, we present a (mostly) generic construction of a $(1,2)$-robust combiner for PKEs that preserves IND-CCA-security in the random oracle model and in the quantum random oracle model, which we call the Quantum Augmented KEM-DEM (or QuAKe) combiner. Furthermore, the security reduction of our construction is tight in both the classical and, more importantly, quantum random oracle models. We achieve this with the use of two PKEs, one of which we require to be built from the KEM-DEM paradigm put forth by Cramer and Shoup [14], and a pair of random oracles, $\mathsf{Hash}_1, \mathsf{Hash}_2$. Our construction relies on preventing an adversary from obtaining both the symmetrically encrypted message, along with a random seed, and the asymmetrically encrypted key by encrypting the symmetric ciphertext under the second PKE. By doing this, an adversary can only obtain the encrypted message and key if they were already able to break the security of both the KEM-DEM and the second PKE. Additionally, the random seed is used to guard against re-encryption and re-encapsulation attacks, as well as mix-and-match attacks. We then prove the security reduction of QuAKe is tight in both the classical and quantum random oracle models.

The paper is organized as follows. In Section 2, we define our notation lemmas required for our proof in the quantum random oracle mode and give some preliminary cryptographic definitions. In Section 3, we present our construction and its security. In Section 4, we review previous results on PKE and KEM combiners, and then provide a comparison with previous results. We then conclude our paper in Section 5.

2. Preliminaries

In this section, we cover the preliminaries used in this work. We begin with the notation used, then an introduction to the random oracle model, the security notions and definitions used, and finally define (k,n)-robust combiners.

2.1. Notation

By $y \leftarrow \mathcal{A}(x)$ we denote an algorithm, $\mathcal{A}$ (either classical or quantum), that runs on (classical) input x, and (classical) outputs y. When $\mathcal{A}$ has access to an oracle, $\mathcal{B}$, we write this as $\mathcal{A}(x)^{\mathcal{B}(\cdot)}$. If $\mathcal{A}$ is an algorithm that uses some randomness in its execution on input x and we wish to specify what the randomness is, say r, we denote it as $\mathcal{A}(x;r)$. We refer to specific subroutines within $\mathcal{A}(\cdot)$ as $\mathcal{A}.\mathsf{Subroutine}$. We consider all adversaries as algorithms (either classical or quantum) that are probabilistic polynomial time on their input length.

We write $x \xleftarrow{\$} S$ to denote that x was outputted by S probabilistically, where if S is some algorithm, then x was selected according to some internal distribution, and if S is some space, such as $\{0,1\}^l$, then we implicitly mean for x to be sampled uniformly at random.

We say a function g mapping non-negative integers to non-negative reals is called *negligible*, if for all positive numbers c, there exists an integer $\lambda_0(c) \geq 0$ such that for

all $\lambda > \lambda_0(c)$ we have $g(\lambda) < \frac{1}{\lambda^c}$. A family of functions F is called *q-wise independent* if for any distinct $x_1, \ldots, x_q$ and for $f \leftarrow\!\!\$\, F$ the values $f(x_1), \ldots, f(x_q)$ are independently uniformly distributed.

2.2. Random Oracle Model and Quantum Random Oracle Model

In this section, we briefly describe the random oracle model (ROM) and quantum random oracle model (QROM) and state the lemmas that will be used in the main result when proving the $\mathsf{Q^cQ}$-IND-CCA-security of QuAKe. We provide a brief introduction to quantum computing in Appendix A and refer to Nielsen and Chuang [15] as a standard text for a more complete explanation.

2.2.1. Random Oracles

In the classical random oracle model (ROM), we assume the existence of a truly random function H and give all parties access to query this function [16]. When implemented, a suitable hash function, Hash, is used in place of H. Any queries to the random oracle, made by any party, is then replaced with an evaluation of Hash on the query.

As in the ROM, in the quantum random oracle model (QROM), all parties are given access to a random function H, which, when implemented, is replaced by a suitable hash function Hash. However, in quantum settings, since a quantum algorithm can evaluate Hash in a superposition of inputs, an adversary must also be allowed to query H in superposition. This is called the quantum random oracle model [17]. To denote when superposition access to H is available, we will adopt the notation $|H\rangle$.

2.2.2. QROM Lemmas

We next state two useful lemmas regarding quantum random oracles. First is the one-way-to-hiding (O2H) Lemma, proven by Unruh [18]. The O2H lemma, informally, states that if a quantum adversary makes at most q queries to a quantum random oracle, $|H\rangle$, and is able to distinguish $(x, H(x))$ from (x, y), where y was sampled uniformly at random, then a simulator can recover x by measuring one of the queries to $|H\rangle$. In other words, this lemma reduces distinguishing $H(x)$ from y to guessing x. We present the updated statement of the O2H Lemma as formulated by Ambainis et al. [19].

Lemma 1 (**One-way-to-hiding, probabilities**). *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S, G(x) = H(x)$. Let z be a random bit string (S, G, H, z may have arbitrary joint distribution). Let $\mathcal{A}$ be a quantum oracle algorithm with query depth q (not necessarily unitary). Let $\mathcal{B}^H$ be an oracle algorithm that on input z does the following: pick $i \leftarrow\!\!\$\, \{1, \cdots, q\}$ run $\mathcal{A}^H(z)$ until (just before) the ith query, measure all query input registers in the computational basis, output the set T of measurement outcomes. Let*

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

$$P_{guess} := \Pr\left[S \cap T \neq \emptyset : T \leftarrow \mathcal{B}^H(z)\right]. \tag{2}$$

Then,

$$\left|P_{left} - P_{right}\right| \leq 2q\sqrt{P_{guess}}, \tag{3}$$

and

$$\left|\sqrt{P_{left}} - \sqrt{P_{right}}\right| \leq 2q\sqrt{P_{guess}}. \tag{4}$$

We refer to the algorithm $\mathcal{B}$ as the O2H simulator. As noted by Ambainis et al., the original statement of the O2H Lemma by Unruh is a consequence of the updated version [19].

The next lemma concerns simulating quantum random oracles. In the original QROM introduced by Boneh et al. [17], reduction or simulator algorithms are not allowed access

to a random oracle, and therefore, must somehow simulate the oracle itself. Zhandry [20] demonstrated that a simulator can use a $2q$-wise independent function to successfully simulate a random oracle, where q denotes the number of queries made by the adversary. Moreover, they proved the simulation of the random oracle is perfect, and no adversary can distinguish between a true random oracle and a $2q$-wise independent function.

Lemma 2. *Let H be an oracle drawn from the set of $2q$-wise independent functions uniformly at random. Then, the advantage any quantum algorithm making at most q queries to H has in distinguishing H from a truly random function is identically 0.*

For the security reduction of the main result in the QROM, we implicitly assume the use of $2q$-wise independent functions by any simulators to answer the adversary's oracle queries.

2.3. Public-Key Encryption

In this section, we provide an overview of public-key encryption algorithms and the relevant security definitions. We begin by defining a public-key encryption (PKE) algorithm. We define the *correctness* of PKEs in Appendix B.

Definition 1 (**Public-Key Encryption Scheme**). *We say a triple of algorithms $\Pi^{asym} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ form a public-key encryption (PKE) scheme, if:*

- KeyGen*: the key generation algorithm is a probabilistic algorithm which on input 1^n $(n \in \mathbb{N})$ outputs a related pair, (pk, sk), of public and secret keys.*
- Enc*: the encryption algorithm is a probabilistic algorithm that takes two inputs, a public-key pk, and a plaintext m, from a designated message space, $\mathcal{M}_{\Pi^{asym}}$, and outputs a ciphertext c.*
- Dec*: the decryption algorithm is a deterministic algorithm that takes as input a secret key sk, and ciphertext c, and returns the plaintext m, or a special designated rejection symbol $\perp$.*

We now define the IND-CCA-security (also referred to as IND-CCA2-security in the literature) and security experiment for PKEs in the random oracle model.

Definition 2 (IND-CCA**-Security for PKEs in the ROM**). *We say that a PKE, Π^{asym}, is* IND-CCA*-secure in the random oracle model if, for all adversaries $\mathcal{A}$, and a random oracle H, we have that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A}) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A}) - \frac{1}{2} \right] \right| \tag{5}$$

is a negligible function in $n \in \mathbb{N}$, where $\mathrm{Expt}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A})$ is defined in Algorithm 1.

Algorithm 1 The IND-CCA-security experiments for PKEs in the ROM, $\mathrm{Expt}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A})$.

1: $(pk, sk) \leftarrow_{\$} \Pi^{asym}.\mathsf{KeyGen}(1^n)$
2: $m_0, m_1, \mathsf{st} \leftarrow \mathcal{A}^{\Pi^{asym}.\mathsf{Dec}(sk,\cdot),H(\cdot)}(pk)$
3: $b \leftarrow_{\$} \{0,1\}$
4: $c^* \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, m_b)$
5: $b' \leftarrow \mathcal{A}^{\Pi^{asym}.\mathsf{Dec}(sk,\cdot \neq c^*),H(\cdot)}(pk, \mathsf{st}, c^*)$
6: Return $[b = b']$

In this work, when we discuss security against quantum adversaries, we are considering the case of an adversary with quantum computational powers, including the ability to query the random oracle in superposition but who only has classical access to decryption oracles. We adopt the $\mathsf{X}^{\mathsf{y}}\mathsf{Z}$ notation, introduced in Bindel et al. [3], to denote security against such adversaries, where X denotes whether the adversary is classical or quantum while they have decryption oracle access, y denotes whether the adversary can perform classical or superposition decryption queries, and Z denotes whether the adversary is

classical or quantum after they lose access to decryption oracles. Thus, IND-CCA-security against a quantum adversary with only classical decryption queries in the quantum random oracle model is denoted as $\mathsf{Q^cQ}$-IND-CCA. Security against an $\mathsf{Q^cQ}$-IND-CCA-adversary is defined analogously to Algorithm 1, and we include the full definition in Appendix B.

2.4. Key/Data Encapsulation Mechanisms

In this section, we define both key encapsulation mechanisms (KEMs), data encapsulation mechanisms (DEMs), and the security notions necessary for this work. We then describe the KEM-DEM paradigm to build a public-key encryption algorithm and the necessary conditions to attain IND-CCA-security using the paradigm. We define the *correctness* for both KEMs and DEMs in Appendix B.

First, we define a key encapsulation algorithm [21].

Definition 3 (**Key Encapsulation Mechanism**). *We say the triple of algorithms* $\mathcal{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ *form a key encapsulation mechanism (KEM), if:*

- KeyGen*: the key generation algorithm is a probabilistic algorithm which on input* 1^n $(n \in \mathbb{N})$*, outputs a related pair,* (ek, dk)*, of a public encapsulation and secret decapsulation keys.*
- Encaps*: the encapsulation algorithm is a probabilistic algorithm that takes one input, a public encapsulation key* ek*, and produces a pair of related outputs, a ciphertext* c*, and an ephemeral key* k*, from a designated key space* $\mathcal{KS}_\mathcal{K}$*.*
- Decaps*: the decapsulation algorithm is a deterministic algorithm that takes as input a secret decapsulation key* dk*, and ciphertext* c*, and returns the related ephemeral key* k*, or a specially designated rejection symbol* $\perp$*.*

Note that a KEM can be built from a public-key encryption algorithm by simply sampling a message uniformly at random, to be used as the key, then encrypting it and sending the ciphertext.

The next definition is for IND-CCA-security of KEMs. We use the standard model in the following definition, but the definition lifts to the random oracle model in a natural way.

Definition 4 (IND-CCA**-Security for KEMs**). *We say that a KEM,* $\mathcal{K}$*, is* IND-CCA *if, for all adversaries* $\mathcal{A}$*, we have that*

$$\mathsf{Adv}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) = \left| \Pr\left[\mathsf{Expt}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) - \frac{1}{2} \right] \right| \tag{6}$$

is a negligible function in $n \in \mathbb{N}$*, where* $\mathrm{Expt}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$ *is defined in Algorithm 2.*

Algorithm 2 The IND-CCA-security experiments for KEMs, $\mathrm{Expt}_{\mathcal{K}}^{\mathsf{IND\text{-}CCA}}(\mathcal{A})$.

1: $(ek, dk) \leftarrow_\$ \mathcal{K}.\mathsf{KeyGen}(1^n)$
2: $\mathsf{st} \leftarrow \mathcal{A}^{\mathcal{K}.\mathsf{Decaps}(dk,\cdot)}(ek)$
3: $(c^*, k_0) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek)$
4: $k_1 \leftarrow_\$ \mathcal{KS}_\mathcal{K}$
5: $b \leftarrow_\$ \{0,1\}$
6: $b' \leftarrow \mathcal{A}^{\mathcal{K}.\mathsf{Decaps}(dk,\cdot \neq c^*)}(ek, \mathsf{st}, c^*, k_b)$
7: Return $[b = b']$

We now informally define IND-CCA-security for KEMs against quantum adversaries in the QROM. As in the PKE case, this security is an extension of classical CCA-security to the QROM with a quantum adversary that is restricted to only classical decryption queries. Once again, we adopt the $\mathsf{X^yZ}$ notation and refer to this security notion as $\mathsf{Q^cQ}$-IND-CCA-security for KEMs, or simply $\mathsf{Q^cQ}$-IND-CCA when the context is clear.

We now define the symmetric primitive and data encapsulation mechanisms (DEMs) [21].

Definition 5 (**Data Encapsulation Mechanism/Symmetric Encryption Scheme**). *We say a triple of algorithms* $\Pi^{sym} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *form a (stateless) data encapsulation mechanism (DEM), or symmetric encryption scheme, if:*

- KeyGen*: the key generation algorithm is a probabilistic algorithm that on input* 1^n $(n \in \mathbb{N})$ *outputs a secret key* k*.*
- Enc*: the encryption algorithm is a deterministic algorithm that takes two inputs, a secret key* k *and a plaintext* m*, from a designated message space* $\mathcal{M}_{\Pi^{sym}}$*, and outputs a ciphertext* c*.*
- Dec*: the decryption algorithm is a deterministic algorithm that takes as input a secret key* k *and ciphertext* c*, and returns the plaintext* m*, or a special designated rejection symbol* $\perp$*.*

In this work, referring to such an algorithm as a DEM rather than symmetric encryption emphasizes the relation between KEMs and DEMs covered in the next section.

Next, we define the primary security definition for DEMs this work is concerned with, the IND-OT-CCA-security introduced by Cramer and Shoup [14]. Informally, IND-OT-CCA is similar to the IND-CCA-security experiment, except that the adversary cannot encrypt any messages of their choice and only after receiving the challenge ciphertext is given access to a decryption oracle. When we wish to consider security of a DEM against a quantum capable adversary, we again make use of the $\mathsf{X}^{\mathsf{y}}\mathsf{Z}$ notation and denote OT-CCA-security for a DEM against a quantum adversary with only classical decryption queries as $\mathsf{Q}^{\mathsf{c}}\mathsf{Q}$-IND-OT-CCA-security.

Definition 6 (IND-OT-CCA**-Security for DEMs**). *We say that a DEM* $\mathcal{K}$ *is* IND-OT-CCA *if, for all adversaries* $\mathcal{A}$*, we have that:*

$$\mathsf{Adv}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{\Pi^{sym}}(\mathcal{A}) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{\Pi^{sym}}(\mathcal{A}) - \frac{1}{2} \right] \right| \tag{7}$$

is a negligible function in $n \in \mathbb{N}$*, where* $\mathrm{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{\Pi^{sym}}(\mathcal{A})$ *is defined in Algorithm 3.*

Algorithm 3 The IND-OT-CCA-security experiments for DEMs, $\mathsf{Expt}^{\mathsf{IND\text{-}OT\text{-}CCA}}_{\Pi^{sym}}(\mathcal{A})$.

1: $(ek, dk) \leftarrow_{\$} \mathcal{K}.\mathsf{KeyGen}(1^n)$
2: $k \leftarrow_{\$} \Pi^{sym}.\mathsf{KeyGen}(1^n)$
3: $m_0, m_1, \mathsf{st} \leftarrow \mathcal{A}(1^n)$
4: $b \leftarrow_{\$} \{0, 1\}$
5: $c^* \leftarrow \Pi^{sym}.\mathsf{Enc}(k, m_b)$
6: $b' \leftarrow \mathcal{A}^{\Pi^{sym}.\mathsf{Dec}(sk, \cdot \neq c^*)}(\mathsf{st}, c^*)$
7: Return $[b = b']$

KEM-DEM Paradigm

We go on to describe how a public-key encryption system can be built from a KEM and DEM pair.

Intuitively, a KEM can be used to generate a key from the key space of the DEM, which, in turn, is used to encrypt the message. The KEM-DEM paradigm of constructing a PKE was first proposed by Cramer and Shoup [14]. The PKE is built by using the KEM keys as the public and private key, generating a ciphertext–key pair from the KEM, then using the key as the symmetric key to encrypt the message and sending the asymmetric and symmetric ciphertexts. Decryption is defined in the natural way. Traditionally, the KEM-DEM is referred to in the literature as a hybrid scheme, referencing public-key–symmetric-key hybrid. To avoid confusion, we elect to use the term KEM-DEM paradigm instead and use hybrid schemes to refer to classical–quantum hybrid schemes.

In addition to proposing the KEM-DEM paradigm, Cramer and Shoup outlined the necessary conditions on the KEM, $\mathcal{K}$, and DEM, Π^{sym}, for the resulting PKE, $(\mathcal{K}, \Pi^{sym})$, to achieve IND-CCA-security. The following theorem outlines these conditions.

Theorem 1. *If $\mathcal{K}$, a KEM, is* IND-CCA*-secure and Π^{sym}, a DEM, is* IND-OT-CCA*-secure, then $(\mathcal{K}, \Pi^{sym})$, described in Algorithms 4–6 is an* IND-CCA*-secure PKE.*

Algorithm 4 $(\mathcal{K}, \Pi^{sym}).\mathsf{KeyGen}(1^n)$.

1: $(ek, dk) \leftarrow_{\$} \mathcal{K}.\mathsf{KeyGen}(1^n)$
2: Return (ek, dk)

Algorithm 5 $(\mathcal{K}, \Pi^{sym}).\mathsf{Enc}(ek, m)$.

1: $(c_1, k) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek)$
2: $c_2 \leftarrow \Pi^{sym}.\mathsf{Enc}(k, n)$
3: Return (c_1, c_2)

Algorithm 6 $(\mathcal{K}, \Pi^{sym}).\mathsf{Dec}(dk, (c_1, c_2))$.

1: $k \leftarrow \mathcal{K}.\mathsf{Decaps}(dk, c_1)$
2: $m \leftarrow \Pi^{sym}.\mathsf{Dec}(k, c_2)$

We include an informal outline of the proof of Theorem 1.

Since $\mathcal{K}$ is IND-CCA-secure, the key used to encrypt the challenge message k^* is indistinguishable from a bit string of equal length sampled uniformly at random, say r. The challenger then generates (c^*_{KEM}, k^*), but instead of using k^* to encrypt the message, r is used. From here, the adversary is then given $(c^*_{KEM}, c^*_{\Pi^{sym}})$ and is effectively in the IND-OT-CCA-security experiment for Π^{sym}.

We make the observation that the proof of this theorem is also directly applicable in the quantum setting, and as a corollary state the quantum version of Theorem 1.

Corollary 1. *If $\mathcal{K}$, a KEM, is* $\mathsf{Q^cQ}$*-IND-CCA-secure and Π^{sym}, a DEM, is* $\mathsf{Q^cQ}$*-IND-OT-CCA-secure, then $(\mathcal{K}, \Pi^{sym})$, as described in Algorithms 4–6, is an* $\mathsf{Q^cQ}$*-IND-CCA-secure PKE.*

2.5. Combiners

In this section, we introduce and formally define *robust combiners*. We provide a review of previous PKE and KEM combiners in Section 4.

First, we define the notion of a (k, n)-robust combiner. Informally, a (k, n)-robust combiner is an algorithm that accepts inputs of n of the same type of cryptographic schemes to produce a new cryptographic scheme, and so long as at least k of the inputs satisfies the same security notion the output is also equally secure. The formal definition of a robust combiner was first proposed by Harnik et al. [2] to formalize such combinations.

Definition 7 ((k, n)-Robust Combiner). *Let $\mathbb{P}$ be a set of cryptographic primitives. A (k, n)-robust combiner is an algorithm that gets n candidate schemes from $\mathbb{P}$ as inputs, and whose output is a single algorithm that is secure to some security specification s, if the following holds:*

1. *If at least k candidates securely implement the security specification s, then the result of the combiner also securely implements s.*
2. *The running time of the result of the combiner is polynomial in the security parameter m, in n, and in the lengths of the inputs to $\mathbb{P}$.*

3. QuAKe

In this section, we propose the Quantum Augmented KEM-DEM (QuAKe) combiner and then prove it is a $(1, 2)$-robust combiner for both IND-CCA- and $\mathsf{Q^cQ}$-IND-CCA-security in the classical and quantum random oracle models, respectively.

3.1. Construction

We first outline the QuAKe combiner. Informally, QuAKe encrypts the message with the KEM-DEM scheme, $(\mathcal{K}, \Pi^{sym})$, then encrypts the DEM component of the ciphertext using the second PKE, Π^{asym}. Intuitively, we prevent an adversary who can break the IND-CCA-security of $(\mathcal{K}, \Pi^{sym})$ from having the $(\mathcal{K}, \Pi^{sym})$ ciphertexts and prevent an adversary who can break the IND-CCA-security of Π^{asym} from directly seeing the message encrypted within. Importantly, along with the message, we encrypt a secret random value, δ. Two hashes of δ are used as the randomness for the encapsulation of the KEM, $\mathcal{K}$, and the encryption under Π^{asym}, respectively. This δ value is then later used during the decryption process to provide a method of ensuring that the ciphertext was well-formed. This ensures that the adversary cannot perform any re-encryption/encapsulation attacks on a ciphertext. We formally present QuAKe in Algorithms 7–9.

Algorithm 7 $\Pi.\mathsf{KeyGen}(1^n)$.

$(pk, sk) \leftarrow_\$ \Pi^{asym}.\mathsf{KeyGen}(1^n)$
$(ek, dk) \leftarrow_\$ \mathcal{K}.\mathsf{KeyGen}(1^n)$
$pk' \leftarrow (pk, ek)$
$sk' \leftarrow (sk, dk)$
Return (pk', sk')

Algorithm 8 $(\mathcal{K}, \Pi^{sym}).\mathsf{Enc}(ek, m)$.

$\delta \leftarrow_\$ \{0,1\}^l$
$(c_{KEM}, k) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek; \mathsf{Hash}_1(\delta))$
$c_{DEM} \leftarrow \Pi^{sym}.\mathsf{Enc}(k, m \| \delta)$
$c_{PKE} \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, c_{DEM}; \mathsf{Hash}_2(\delta))$
return (c_{KEM}, c_{PKE})

Algorithm 9 $\Pi.\mathsf{Dec}((sk, dk), ((c_{KEM}, c_{PKE})))$.

$c_{DEM} \leftarrow \Pi^{asym}.\mathsf{Dec}(sk, c_{PKE})$
$k \leftarrow \mathcal{K}.\mathsf{Decaps}(dk, c_{KEM})$
$m \| \delta \leftarrow \Pi^{sym}.\mathsf{Dec}(k, c_{DEM})$
if $m \| \delta = \perp$ **then**
    Return $\perp$
else
    if $(c_{KEM}, -) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek; \mathsf{Hash}_1(\delta)) \wedge c_{PKE} \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, c_{DEM}; \mathsf{Hash}_2(\delta))$ **then**
        Return m
    else
        Return $\perp$
    end if
end if

We first provide some insight into the construction of QuAKe. As mentioned, the use of δ is to deterministically randomize the encryption (encapsulation) of Π^{asym} ($\mathcal{K}$) and perform a check. This check is intended to guard against adversaries capable of total manipulation of either Π^{asym} or $\mathcal{K}$. For example, if an adversary is able to recover the key from any ciphertext of $\mathcal{K}$, as well as encapsulate any key they wish, the adversary may attempt to recover the key used in the challenge ciphertext, re-encapsulate to a different ciphertext, and then submit it to the decryption oracle. Without a check, the decryption oracle would then return the challenge message directly to the adversary. The case of Π^{asym} being completely broken by the adversary is analogous to the situation described above. Thus, without any check, there is no hope for security if one of Π^{asym} or $\mathcal{K}$ is broken by the adversary.

3.2. Security of QuAKe

In this section, we prove that QuAKe is a $(1,2)$-robust combiner with a tight security reduction for both IND-CCA and $\mathsf{Q^cQ}$-IND-CCA-security in the classical and quantum random oracle models, respectively.

3.2.1. IND-CCA-Security of QuAKe.

We begin with the IND-CCA-security.

Theorem 2. *Let* $\mathsf{Hash}_1 : \{0,1\}^l \to \{0,1\}^{n_1}$ *and* $\mathsf{Hash}_2 : \{0,1\}^l \to \{0,1\}^{n_2}$ *be random oracles. Let* $(\mathcal{K}, \Pi^{sym})$ *be an* ϵ_1*-correct KEM-DEM-based PKE and* Π^{asym} *be an* ϵ_2*-correct PKE. If either* $(\mathcal{K}, \Pi^{sym})$ *is* IND-CCA*-secure or* Π^{asym} *is* IND-CCA*-secure PKE, then* Π*, as described in Algorithms 7–9, is a* $(1,2)$*-robust combiner for* IND-CCA*-security in the random oracle model and is* ϵ*-correct, where* $\epsilon = \max\{\epsilon_1, \epsilon_2\}$*. More precisely, for any efficient classical adversary* $\mathcal{A}$ *that breaks the* IND-CCA*-security of* Π*, there exist efficient adversaries* $\mathcal{B}_1$*, and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}_{\Pi}^{\mathsf{IND\text{-}CCA}} \leq \frac{q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2}}{2^l} + \epsilon \cdot q_{Dec} + \min\{\mathsf{Adv}_{(\mathcal{K},\Pi^{sym})}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}_1), \mathsf{Adv}_{\Pi^{asym}}^{\mathsf{IND\text{-}CCA}}(\mathcal{B}_2)\}, \quad (8)$$

where the run times of all $\mathcal{B}_i$ *are approximately equal to that of* $\mathcal{A}$*,* q_{Dec} *is the number of decryption queries made by* $\mathcal{A}$*, and* q_{Hash_i} *is the number of queries made to the random oracle* Hash_i *by* $\mathcal{A}$*.*

We begin with the correctness of Π. It is straightforward to see that Π is perfectly correct if both $(\mathcal{K}, \Pi^{sym})$ and Π^{asym} are perfectly correct. If $(\mathcal{K}, \Pi^{sym})$ is perfectly correct, then the real encrypted message $m\|\delta$ in (c_{KEM}, c_{DEM}) is always recovered when running the $(\mathcal{K}, \Pi^{sym})$ part of the decryption. If Π^{asym} is perfectly correct, then c_{PKE} always correctly decrypts to c_{DEM}. Finally, the deterministic re-encryption check is done; thus, an honestly generated ciphertext δ would have been used to generate the randomness for both c_{KEM} and c_{PKE} and so would pass the re-encryption check. Therefore, we have that Π is perfectly correct.

We now suppose that $(\mathcal{K}, \Pi^{sym})$ and Π^{asym} are both not perfectly correct, but are ϵ_1- and ϵ_2-correct, respectively. Whenever a decryption query is made, both $\mathcal{K}$.Decaps and Π^{asym}.Dec are performed. In the case of a decapsulation error occurring, a different symmetric key is recovered and used to decrypt some c'_{DEM} and obtain some $m'\|\delta'$. The case where $m'\|\delta'$ is equal to the actual message and randomness used is overwhelmingly unlikely. Therefore, either $\perp$ is returned if either check fails or a different message is returned if the checks are passed. As decapsulation errors happen with probability ϵ_1, the probability of returning a different output instead of the true message is bounded above by probability ϵ_1.

The argument is analogous when considering the case of a decryption error occurring, and the probability of an output that is not the true message being returned is bounded above by ϵ_2. We thus conclude that Π is ϵ-correct, where $\epsilon = \max\{\epsilon_1, \epsilon_2\}$.

Proof. We now prove the security of Π. First, we assume $(\mathcal{K}, \Pi^{sym})$ is an IND-CCA-secure KEM-DEM encryption scheme.

G_0: *Game 0*, described in Algorithm 10, is the IND-CCA-security experiment for Π so,

$$\mathsf{Adv}_{\Pi}^{\mathsf{IND\text{-}CCA}} = \left|\Pr[G_0 \to 1] - \frac{1}{2}\right|. \quad (9)$$

Algorithm 10 Game 0 for the proof of Theorem 2, G_0.

$((pk, ek), (sk, dk)) \leftarrow_{\$} \Pi.\mathsf{KeyGen}(1^n)$
$m_0, m_1, st \leftarrow \mathcal{A}^{\mathsf{Dec}((sk,dk),\cdot),\mathsf{Hash}_1(\cdot),\mathsf{Hash}_2(\cdot)}(pk, ek)$
$b \leftarrow_{\$} \{0,1\}$
$\delta^* \leftarrow_{\$} \{0,1\}^l$
$(c^*_{KEM}, k^*) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek; \mathsf{Hash}_1(\delta^*))$
$c^*_{DEM} \leftarrow \Pi^{sym}.\mathsf{Enc}(k^*, m_b \| \delta^*)$
$c^*_{PKE} \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, c^*_{DEM}; \mathsf{Hash}_2(\delta^*))$
$b' \leftarrow \mathcal{A}^{\mathsf{Dec}((sk,dk),\cdot \neq (c^*_{KEM}, c^*_{PKE})),\mathsf{Hash}_1(\cdot),\mathsf{Hash}_2(\cdot)}(pk, ek, st, (c^*_{KEM}, c^*_{PKE}))$
Return $[b' = b]$

G_1: For *Game 1*, outlined in Algorithm 11, we replace both $\mathsf{Hash}_1(\delta^*)$ and $\mathsf{Hash}_2(\delta^*)$ with h' and h'', respectively, chosen uniformly at random from the ranges of Hash_1 and Hash_2, so that if Hash_i is called on δ^*, either h' or h'' is returned instead of $\mathsf{Hash}_1(\delta^*)$, or $\mathsf{Hash}_2(\delta^*)$, respectively. This replacement is done consistently with the decryption algorithm so that if Hash_i is evaluated on δ^* during decryption, h' and h'' are used to perform the checks. The probability of distinguishing between the first two games is upper bounded by the probability of $\mathcal{A}$ guessing δ^* and querying it. As δ^* was sampled uniformly at random, and as $\mathcal{A}$ has no access to δ^*, unless they are able to completely invert both Π^{asym} and $(\mathcal{K}, \Pi^{sym})$ since the decryption oracle never returns δ^*, we have that

$$|\Pr[G_0 \to 1] - \Pr[G_1 \to 1]| \leq \frac{q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2}}{2^l}, \tag{10}$$

where q_{Hash_i} is the number of queries $\mathcal{A}$ makes to Hash_i.

Algorithm 11 Game 1 for the proof of Theorem 2, G_1.

$((pk, ek), (sk, dk)) \leftarrow_{\$} \Pi.\mathsf{KeyGen}(1^n)$
$m_0, m_1, st \leftarrow \mathcal{A}^{\mathsf{Dec}((sk,dk),\cdot),\mathsf{Hash}_1(\cdot),\mathsf{Hash}_2(\cdot)}(pk, ek)$
$b \leftarrow_{\$} \{0,1\}$
$\delta^* \leftarrow_{\$} \{0,1\}^l$
$h' \leftarrow_{\$} \mathsf{Hash}_1.Range$ //Using random coins in place of $\mathsf{Hash}_1(\delta^*)$
$h'' \leftarrow_{\$} \mathsf{Hash}_2.Range$ //Using random coins in place of $\mathsf{Hash}_2(\delta^*)$
$(c^*_{KEM}, k^*) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek; h')$
$c^*_{DEM} \leftarrow \Pi^{sym}.\mathsf{Enc}(k^*, m_b \| \delta^*)$
$c^*_{PKE} \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, c^*_{DEM}; h'')$
$b' \leftarrow \mathcal{A}^{\mathsf{Dec}((sk,dk),\cdot \neq (c^*_{KEM}, c^*_{PKE})),\mathsf{Hash}_1(\cdot),\mathsf{Hash}_2(\cdot)}(pk, ek, st, (c^*_{KEM}, c^*_{PKE}))$
Return $[b' = b]$

We can now demonstrate that if $\mathcal{A}$ can win *Game 1*, then $\mathcal{A}$ can be used as an oracle algorithm to win the IND-CCA experiment of $(\mathcal{K}, \Pi^{sym})$.
Claim: [1]

$$\Pr[G_1 \to 1] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{(\mathcal{K}, \Pi^{sym})}(\mathcal{B}_1) + \epsilon \cdot q_{Dec}. \tag{11}$$

Proof. In the IND-CCA experiment of $(\mathcal{K}, \Pi^{sym})$, $\mathcal{B}_1$ is first given the encapsulation key ek of $\mathcal{K}$, then it runs Π^{asym} and generates (pk, sk), and picks two random oracles Hash_1 and Hash_2, then forwards (pk, ek) to $\mathcal{A}$. Here, $\mathcal{B}_1$ is able to perfectly simulate the oracles $\mathcal{A}$ would have at this point, until eventually $\mathcal{A}$ terminates and outputs its two challenge messages m_0 and m_1. Then, $\mathcal{B}_1$ picks δ^* uniformly at random and submits $m_0 \| \delta^*$ and $m_1 \| \delta^*$ as its challenge messages. It then receives (c^*_{KEM}, c^*_{DEM}) before picking h'' uniformly at random and computing $c^*_{PKE} \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, c^*_{DEM}; h'')$ and giving (c^*_{KEM}, c^*_{PKE}) to $\mathcal{A}$.

To answer decryption queries for $\mathcal{A}$, when $\mathcal{B}_1$ receives (c_{KEM}, c_{PKE}) it first decrypts c_{PKE} itself by using sk, then uses its decryption oracle (c_{KEM}, c_{PKE}), and finally performs

the checks after receiving the decryption $m'\|\delta'$. Note that if $(c_{KEM}, c_{DEM}) = (c^*_{KEM}, c^*_{DEM})$, then $\mathcal{B}_1$ cannot query its decryption oracle, but it does not need to do so, as in the real experiment such a query, provided a decryption error does not occur, would fail the re-encryption check. Thus, $\mathcal{B}_1$ will just perform a hash query to each of Hash_1 and Hash_2 before returning $\perp$. As a result, the simulation is then perfect unless an error occurs in *Game 1* during decryption and results in a message being returned. Eventually, $\mathcal{A}$ will submit a guess and $\mathcal{B}_1$ will submit the same guess.

Let E denote the event that $\mathcal{A}$ performs a query of the form, (c^*_{KEM}, c_{PKE}) and $\mathsf{Dec}(sk, c_{PKE}) = c^*_{DEM}$, that does not return $\perp$ in *Game 1* but does return $\perp$ when simulated by $\mathcal{B}_1$.

We have that

$$\Pr[G_1 \to 1] = \Pr[G_1 \to 1|E] \cdot \Pr[E] + \Pr[G_1 \to 1|\neg E] \cdot \Pr[\neg E] \tag{12}$$

$$\leq \Pr[E] + \Pr[G_1 \to 1|\neg E] \tag{13}$$

First, note that $\neg E$ corresponds to $\mathcal{B}_1$'s simulation of $\mathcal{A}$'s decryption oracle being perfect, that is $\mathcal{B}_1$'s answer to queries of the form (c^*_{KEM}, c_{PKE}) and $\mathsf{Dec}(sk, c_{PKE}) = c^*_{DEM}$ agree with $\mathcal{A}$'s decryption oracle. Thus, we have

$$\Pr[G_1 \to 1|\neg E] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_1). \tag{14}$$

We now bound from above $\Pr[E]$. As described, if decryption was performed correctly, the oracle would return $\perp$ as the query would fail the re-encryption check; thus, E could only have occurred if a decryption error had taken place. Since QuAKe is ϵ-correct we apply a uniform bound across all decryption queries and conclude that $\Pr[E] \leq \epsilon \cdot q_{Dec}$.

Thus , we have

$$\Pr[G_1 \to 1] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{(\mathcal{K}, \Pi^{sym})}(\mathcal{B}_1) + \epsilon \cdot q_{Dec}.$$

□

As a result, by combing the inequalities above, we have

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{(\mathcal{K}, \Pi^{sym})}(\mathcal{B}_1) + \epsilon \cdot q_{Dec} + \frac{q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2}}{2^l}. \tag{15}$$

□

We now consider the case where Π^{asym} is an IND-CCA PKE.

Proof. G_0: *Game 0*, as described in Algorithm 10, is the IND-CCA-security experiment for Π so,

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{kdp} = \left|\Pr[G_0 \to 1] - \frac{1}{2}\right|. \tag{16}$$

G_1: For *Game 1*, we perform the same game hop as before in Algorithm 11, and replace both $\mathsf{Hash}_1(\delta^*)$ and $\mathsf{Hash}_2(\delta^*)$ with h' and h'', chosen independently and uniformly at random from the ranges of Hash_1 and Hash_2. This change is done consistently with the random oracles and the decryption oracle so that if δ^* is queried during decryption, h' and h'' are used for the check. By the same argument as above we have that the probability that $\mathcal{A}$ distinguishes between these games is

$$|\Pr[G_0 \to 1] - \Pr[G_1 \to 1]| \leq \frac{q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2}}{2^l}, \tag{17}$$

where q_{Hash_i} is the number of queries $\mathcal{A}$ makes to Hash_i.

We are now able to show that if $\mathcal{A}$ can win *Game 1*, then $\mathcal{A}$ can be used as an oracle algorithm to win the IND-CCA experiment of Π^{asym}.

Claim: [2]

$$\Pr[G_1 \to 1] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2) + \epsilon \cdot q_{Dec}. \tag{18}$$

Proof. Suppose that $\mathcal{A}$ can win *Game 1* in the classical setting with non-negligible probability, then it can be used as an oracle algorithm to break the IND-CCA-security of Π^{asym} as follows. Once $\mathcal{B}_2$ is given pk by the IND-CCA challenger it runs $(\mathcal{K}, \Pi^{sym})$ to generate the key pair (ek, dk), then it selects Hash_1 and Hash_2 uniformly at random, and finally forwards the (pk, ek) to $\mathcal{A}$. At this stage, $\mathcal{B}_2$ can perfectly act as a random oracle to $\mathcal{A}$ and uses its decryption oracle for Π^{asym} to act as a decryption oracle for Π. Eventually, $\mathcal{A}$ terminates and outputs challenge messages m_0 and m_1 to $\mathcal{B}_2$ who then selects δ^* and h' uniformly at random and performs the following computations:

$$(c^*_{KEM}, k^*) \leftarrow \mathcal{K}.\mathsf{Encaps}(ek; h'), \tag{19}$$

$$c^*_{DEM,0} \leftarrow \Pi^{sym}.\mathsf{Enc}(k, m_0 \| \delta^*), \tag{20}$$

$$c^*_{DEM,1} \leftarrow \Pi^{sym}.\mathsf{Enc}(k, m_1 \| \delta^*). \tag{21}$$

Then, $\mathcal{B}_2$ submits $c^*_{DEM,0}, c^*_{DEM,1}$ as its challenge messages. Next, $\mathcal{B}_2$ then receives c^*_{PKE} and forwards (c^*_{KEM}, c^*_{PKE}) to $\mathcal{A}$ and acts as the decryption oracle. We note that there are special cases of decryption queries that $\mathcal{B}_2$ will not fully perform, but can answer.

1. (c, c^*_{PKE}): $\mathcal{B}_2$ cannot query c^*_{PKE} to its own decryption oracle. However, $\mathcal{B}_2$ answers the query as follows: use dk to decapsulate c; if the result is k^*, query Hash_1 and Hash_2 on r, a uniform random value, then return $\perp$; otherwise, simply return $\perp$ without querying the random oracles. In the first case, the real IND-CCA experiment would reject, as the re-encapsulation check would fail. While in the second case, the symmetric decryption algorithm would reject as $c^*_{DEM,b}$ was encrypted under k^* and the key given was different.
2. $(c^*_{KEM}, c_{PKE}) \wedge \mathsf{Dec}(sk, c_{PKE}) = c^*_{DEM,b}$: $\mathcal{B}_2$ will always return $\perp$ and query Hash_1 and Hash_2 on r, a uniform random value. In a real experiment, such a query would be rejected as it would fail the re-encryption check.

For all other queries, $\mathcal{B}_2$ is able to perform decryption queries perfectly matching $\mathcal{A}$'s oracles.

Eventually, $\mathcal{A}$ terminates and outputs a guess, which $\mathcal{B}_2$ matches.

Let F denote the event that $\mathcal{A}$ performs a special case decryption query that does not return $\perp$ in *Game 1* but returns $\perp$ when simulated by $\mathcal{B}_2$.

We have that

$$\Pr[G_1 \to 1] = \Pr[G_1 \to 1|F] \cdot \Pr[F] + \Pr[G_1 \to 1|\neg E] \cdot \Pr[\neg F] \tag{22}$$

$$\leq \Pr[F] + \Pr[G_1 \to 1|\neg F]. \tag{23}$$

First, note that $\neg F$ corresponds to $\mathcal{B}_2$'s simulation of $\mathcal{A}$'s decryption oracle being perfect, as it was able to correctly answer all special case decryption queries. Thus, we have

$$\Pr[G_1 \to 1|\neg F] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2). \tag{24}$$

We now bound from above $\Pr[F]$. As outlined above in how $\mathcal{B}_2$ answers special decryption cases, if decryption was performed correctly, both cases would return $\perp$, thus F could only have occurred if a decryption error had taken place. Since QuAKe is ϵ-correct we apply a uniform bound across all decryption queries and conclude that $\Pr[F] \leq \epsilon \cdot q_{Dec}$.

Thus, we have

$$\Pr[G_1 \to 1] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2) + \epsilon \cdot q_{Dec}.$$

□

Finally, we can conclude if Π^{asym} is in an IND-CCA PKE, then

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi} \leq \mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2) + \epsilon \cdot q_{Dec} + \frac{q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2}}{2^l}. \tag{25}$$

□

3.2.2. Q$^{\mathsf{c}}$Q-IND-CCA-Security of QuAKe

Next, we prove the Q$^{\mathsf{c}}$Q-IND-CCA-security of QuAKe.

Theorem 3. *Let* $\mathsf{Hash}_1 : \{0,1\}^l \to \{0,1\}^{n_1}$ *and* $\mathsf{Hash}_2 : \{0,1\}^l \to \{0,1\}^{n_2}$ *be quantum random oracles. Let* $(\mathcal{K}, \Pi^{sym})$ *be an* ϵ_1*-correct KEM-DEM based PKE and* Π^{asym} *be an* ϵ_2*-correct PKE. If either* $(\mathcal{K}, \Pi^{sym})$ *is* Q$^{\mathsf{c}}$Q-IND-CCA*-secure or* Π^{asym} *is* Q$^{\mathsf{c}}$Q-IND-CCA*-secure, then* Π*, as described in Algorithms 7–9, is a* $(1,2)$*-robust combiner for* Q$^{\mathsf{c}}$Q-IND-CCA*-security in the quantum random oracle model and is* ϵ*-correct, where* $\epsilon = \max\{\epsilon_1, \epsilon_2\}$*. More precisely, for any efficient quantum adversary* $\mathcal{A}$ *that breaks the* Q$^{\mathsf{c}}$Q-IND-CCA*-security of* Π*, there exists efficient adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi} \leq \frac{2(q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2} + q_{\mathsf{Dec}})}{\sqrt{2^l}} + \epsilon \cdot q_{Dec} + \min\begin{cases} \mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{(\mathcal{K},\Pi^{sym})}(\mathcal{B}_1) \\ \mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2) \end{cases}. \tag{26}$$

where the run times of all $\mathcal{B}_i$ *are approximately equal to that of* $\mathcal{A}$*, and the hash queries performed by each* $\mathcal{B}_i$ *are equal to* $\mathcal{A}$*.*

Proof. We will outline the proof of Q$^{\mathsf{c}}$Q-IND-CCA-security of QuAKe. It is virtually identical to that of Theorem 2, except Hash_1 and Hash_2 are replaced with quantum random oracles, $|\mathsf{Hash}_1\rangle$ and $|\mathsf{Hash}_2\rangle$, the simulator algorithms use $2q'$-wise independent functions, where q' is the number of queries the adversary makes in the reduction, and the adversaries are quantum. The primary difference is the first game hop to G_1, as described in Algorithm 11 in both halves of the proof. In the classical setting, we employ a simple replacement of $\mathsf{Hash}_i(\delta)$ with h' or h''. However, such a replacement does not work in the quantum random oracle model. This is due to the inability of the challenger to read the oracle queries and perform the replacement of $\mathsf{Hash}_1(\delta)$ with h' and $\mathsf{Hash}_2(\delta)$ with h'' in the quantum random oracle model as a result of superposition access to the oracles. Instead, we invoke the O2H Lemma 1 to replace the hashes with random values and otherwise continue in the same manner. We provide a description of the O2H simulator $\mathcal{B}$ below in Algorithm 12, where $\mathcal{A}$ is the Q$^{\mathsf{c}}$Q-IND-CCA adversary from the security experiment. After this replacement, the proof continues in the same fashion as in Theorem 2.

In *Game 1*, we reprogram both Hash_1 and Hash_2 on δ^*; thus, to invoke the O2H Lemma we set $S := \{\delta\}$, $H := |\mathsf{Hash}_1 \times \mathsf{Hash}_2\rangle$, where $\mathsf{Hash}_1 \times \mathsf{Hash}_2(a) := (\mathsf{Hash}_1(a), \mathsf{Hash}_2(a))$, G is H except reprogrammed such that when queried on δ, (h', h'') is returned, and $z = (\delta^*, (\mathsf{Hash}_1(\delta^*), \mathsf{Hash}_2(\delta^*))$.

Algorithm 12 O2H Simulator algorithm, $\mathcal{B}^{|\mathsf{Hash}_1 \times \mathsf{Hash}_2\rangle}(\delta^*)$

1: $i \leftarrow_{\$} \{1, ..., q'\}$
2: run $\mathcal{A}^{|\mathsf{Hash}_1 \times \mathsf{Hash}_2\rangle}(\delta^*, (\mathsf{Hash}_1(\delta^*), \mathsf{Hash}_2(\delta^*))$ until the ith query.
3: **if** $i >$ the number of queries made to $|\mathsf{Hash}_1 \times \mathsf{Hash}_2\rangle$ **then**
4: Return $\perp$
5: **else**Measure the query $\hat{\delta}$
6: return $[[\hat{\delta} = \delta^*]]$
7: **end if**

We then have that P_{left}, as in the O2H Lemma, describes *Game 0* for $\mathcal{A}$ as it is running with the original, untampered oracle H. While P_{right} describes $\mathcal{A}$ running with the repro-

grammed oracle G, which returns (h', h'') when queried on δ^*. In both cases, $\mathcal{A}$ is running on input $z = (\delta^*, (\mathsf{Hash}_1(\delta^*), \mathsf{Hash}_2(\delta^*))$.

Finally, P_{guess} is then described in Algorithm 12. Note, as δ^* was chosen uniformly at random from $\{0,1\}^l$, we have $P_{guess} = \frac{1}{2^l}$. We also note that $\mathcal{B}$ makes at most $q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2} + q_{\mathsf{Dec}}$ queries to its oracle $|\mathsf{Hash}_1 \times \mathsf{Hash}_2\rangle$ to answer $\mathcal{A}$'s queries, and so we have that $q' = q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2} + q_{\mathsf{Dec}}$. Thus, by the O2H Lemma, we have that

$$|\Pr[G_0 \to 1] - \Pr[G_1 \to 1]| \leq \frac{2(q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2} + q_{\mathsf{Dec}})}{\sqrt{2^l}}. \tag{27}$$

As mentioned before, from here, the proof proceeds in the same fashion as in Theorem 2, and we can then conclude

$$\mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi} \leq \frac{2(q_{\mathsf{Hash}_1} + q_{\mathsf{Hash}_2} + q_{\mathsf{Dec}})}{\sqrt{2^l}} + \epsilon \cdot q_{Dec} + \min\begin{cases} \mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{(\mathcal{K},\Pi^{sym})}(\mathcal{B}_1) \\ \mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{B}_2) \end{cases}.$$

□

4. Comparisons

We next include a brief discussion on the previous results on combiners, beginning with PKE combiners and then KEM combiners.

4.1. PKE Combiners

Asmuth and Blakely [5] proposed what would retroactively become one of the first PKE combiner with the so-called "XOR-Input" combiner, defined as

$$c = (\Pi_1.\mathsf{Enc}(pk_1, r), \Pi_2.\mathsf{Enc}(pk_2, m \oplus r)), \tag{28}$$

where r is a uniformly random value. However, Harnik et al. would later prove that such a combiner is not robust for IND-CCA-security, but rather is only robust for weaker notions, such as IND-CPA and IND-CCA1-security [6]. Additionally, Harnik et al. proved that the nested, or *cascade*, encryption combiner is also not robust for IND-CCA-security.

Hohenberger et al. [8] proposed a combiner which does attain IND-CCA-security but requires three encryption schemes, each of which are required to be secure according to three *different* security notions, meaning it is, in essence, a $(3,3)$-robust combiner. This undermines the appeal of using combiners in that the resulting algorithm is secure without full trust in the security of all inputs. Furthermore, there does not exist any result on the necessary and, more importantly, sufficient conditions to reach $\mathsf{Q^cQ}$-IND-CCA-security. Thus, the viability of this combiner for classical and quantum schemes is unclear.

4.2. KEM Combiners

Giacon et al. provided several IND-CCA-robust KEM combiners with different constructions assuming either the standard, random oracle, or ideal cipher models [10]. In this work, various key-mixing functions, or *core functions*, as Giacon et al. termed them, are used to combine the different keys and ciphertexts to produce a new secure key. Moreover, the authors establish a sufficient condition for the core function, such that the respective combiners retain IND-CCA-security from any one of the input KEMs. This condition is a novel security notion called *split-key pseudorandomness*. Informally, split-key pseudorandomness states that so long as any of the input keys are uniformly distributed, then the resulting evaluation of the core function will also be uniformly distributed. The specific core functions used in this work include: evaluating a pseudorandom function (PRF) on an individual key and all ciphertexts before XORing all evaluations together, using a random oracle on the ciphertexts and the keys or in a nested series of keyed permutations, and finally evaluating a PRF on the ciphertexts and a nested series of ideal cipher evaluations.

Bindel et al. provided further constructions of KEM combiners with the further consideration of adversaries with different levels of quantum capabilities, so-called X^yZ adversaries [4]. The results of this approach were three KEM combiners in the same vein as Giacon et al.'s core functions which combine the different keys to produce the new key while also being secure against both classical and quantum adversaries. Specifically, the combiners constructed are the so-called "XOR-then MAC" combiners where the keys are split to produce an ephemeral key and MAC key to be computed over the ciphertexts and two combiners using a dual PRF as well as traditional PRFs.

Ultimately, the most successful previous works are those which present constructions of hybrid secure robust KEM combiners, that hybridize PKEs according to Theorem 1 and Corollary 1. However, this is done utilizing a similar approach, namely generating multiple keys and using core functions to sufficiently randomize them and creating a single symmetric key.

In comparison, our construction relies on fewer and relatively traditional assumptions of cryptographic primitives than the previous constructions. However, our construction does have the drawback of worse performance during decryption due to the re-encryption and re-encapsulation checks performed. Finally, most of the previous constructions rely on the secure KEM component having perfect correctness. This limits which quantum-resistant algorithms can be used as inputs due to the correctness error present in many lattice-based schemes. Our construction does not have this limitation.

5. Conclusions

Hybrid cryptographic algorithms, which combine both classical and quantum-resistant primitives, offer an appealing solution to the upcoming problems associated with the transitioning of the cryptographic infrastructure. Such algorithms provide quantum-resistance without the risk of losing current-day security assurances and satisfy existing standards. While there have been earlier work on the theory of hybrid cryptography, there have not been any results on direct constructions of hybrid PKEs or hybrid combiners for PKEs. The more common approach has been to develop KEM combiners and then convert the resulting KEM into a PKE.

In this work, we furthered the theory of hybrid cryptography by proposing a new provably secure $(1,2)$-robust combiner, QuAKe, with tight reduction for both IND-CCA-security in the random oracle model, and Q^cQ-IND-CCA-security in the quantum random oracle model. Our combiner built from the KEM-DEM paradigm first proposed by Cramer and Shoup [14] and augmented the paradigm to directly construct a new, fully secure PKE. Moreover, our construction was built with fewer components and security assumptions when compared with previous results, as well as not being limited to schemes with perfect decryption correctness.

Author Contributions: Conceptualization, B.G. and A.M.; methodology, B.G. and A.M.; formal analysis, B.G. and A.M.; resources, A.M.; writing—original draft preparation, B.G.; writing—review and editing, B.G. and A.M.; visualization, B.G.; supervision, A.M.; funding acquisition, A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Grant CRDPJ 536635, and NXM Labs Inc.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funding agencies had no role in the design of the study; in the analyses, in the writing of the manuscript, or in the decision to publish the results.

Appendix A

We now provide a brief introduction to quantum computation knowledge used for this work. Nielsen and Chuang [15] provide a standard text with a more complete explanation of the subject.

Let $\mathcal{H}$ be a finite-dimensional complex Hilbert space with an inner product. Vectors in $\mathcal{H}$ are denoted with "bra–ket" notation, with $|x\rangle$ being a vector in $\mathcal{H}$, and $\langle x|$ denoting the complex conjugate transpose of $|x\rangle$. The inner product on $|x\rangle, |y\rangle$ is then given by $\langle x, y|x, y\rangle$. A quantum state is defined as a vector in $\mathcal{H}$ with norm 1. Let $\{|x\rangle\}_x$ be a basis for $\mathcal{H}$, then any quantum state $|y\rangle$ in $\mathcal{H}$ can be represented in superposition as,

$$|y\rangle = \sum \phi_x |x\rangle, \tag{A1}$$

where ϕ_x are complex numbers such that $|y\rangle$ has norm 1.

For two quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the joint quantum system is given by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$; for two states $|x\rangle$ in $\mathcal{H}_1$ and $|y\rangle$ in $\mathcal{H}_2$, the joint quantum state is represented as $|x\rangle|y\rangle$, or $|x, y\rangle$. Quantum operations on $\mathcal{H}$ are represented by the unitary transformations **U**. Consequently, these quantum operations are, in fact, reversible prior to measurement, as during quantum computation they are unitary matrices. This is notable as it imposes some constraints with the quantization of classical operations such as decryption or decapsulation. In particular, let A be a classical algorithm with input x in $\{0,1\}^a$ and output y in $\{0,1\}^b$ and

$$\{0,1\}^a \times \{0,1\}^b \to \{0,1\}^a \times \{0,1\}^b : (x, t) \mapsto (x, t \oplus A(x)) \tag{A2}$$

be a classical reversible mapping. Then, the corresponding unitary transformation **A** acting linearly on quantum states is given by

$$\mathbf{A} : \sum_{x,t} \psi_{x,t} |x, t\rangle \mapsto \sum_{x,t} \psi_{x,t} |x, t \oplus A(x)\rangle.$$

For full generality, an additional workspace register may be included with the input and output registers. Thus, the general quantization of the classical algorithm is

$$\mathbf{A} : \sum_{x,t,z} \psi_{x,t,z} |x, t, z\rangle \mapsto \sum_{x,t,z} \psi_{x,t,z} |x, t \oplus A(x), z\rangle. \tag{A3}$$

Appendix B

We now include the omitted definitions from Section 2 of the main paper.

Definition A1 (Correctness of PKEs). *We say that a public-key encryption scheme Π^{asym} is ϵ-correct, if for all messages m in the message space $\mathcal{M}_{\Pi^{asym}}$:*

$$\Pr[\mathsf{Dec}(sk, c) \neq m | (pk, sk) \leftarrow \mathsf{KeyGen}, c \leftarrow \mathsf{Enc}(pk, m)] \leq \epsilon. \tag{A4}$$

We say that a PKE Π^{asym} is perfectly correct, if $\epsilon = 0$.

Definition A2 (IND-CCA**-Security for PKEs in the QROM).** *We say that a PKE Π^{asym} is* IND-CCA*-secure in the QROM (*$\mathsf{Q^cQ}$-IND-CCA*-secure) in the quantum random oracle model, if, for all quantum adversaries $\mathcal{A}_\mathsf{Q}$ and a quantum random oracle $|H\rangle$, we have that*

$$\mathsf{Adv}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A}_\mathsf{Q}) = \left| \Pr\left[\mathsf{Expt}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A}_\mathsf{Q}) - \frac{1}{2} \right] \right| \tag{A5}$$

is a negligible function in $n \in \mathbb{N}$, where $\mathrm{Expt}^{\mathsf{Q^cQ\text{-}IND\text{-}CCA}}_{\Pi^{asym}}(\mathcal{A}_\mathsf{Q})$ is defined in Algorithm A1.

Algorithm A1 The Q^{c}Q-IND-CCA-security experiments for PKEs in the QROM, $\mathrm{Expt}^{\mathsf{Q}^{\mathsf{c}}\text{Q-IND-CCA}}_{\Pi^{asym}}(\mathcal{A}_{\mathsf{Q}})$.

1: $(pk, sk) \leftarrow_{\$} \Pi^{asym}.\mathsf{KeyGen}(1^n)$
2: $m_0, m_1, \mathsf{st} \leftarrow \mathcal{A}_{\mathsf{Q}}^{\Pi^{asym}.\mathsf{Dec}(sk,\cdot),|H(\cdot)\rangle}(pk)$
3: $b \leftarrow_{\$} \{0,1\}$
4: $c^* \leftarrow \Pi^{asym}.\mathsf{Enc}(pk, m_b)$
5: $b' \leftarrow \mathcal{A}_{\mathsf{Q}}^{\Pi^{asym}.\mathsf{Dec}(sk,\cdot\neq c^*),|H(\cdot)\rangle}(pk, \mathsf{st}, c^*)$
6: return $[b = b']$

Definition A3 (Correctness of KEMs). *We say that a KEM $\mathcal{K}$ is ϵ-correct, if:*

$$\Pr[\mathsf{Decaps}(dk, c) \neq k | (ek, dk) \leftarrow \mathsf{KeyGen}, (c, k) \leftarrow \mathsf{Encaps}(ek)] \leq \epsilon. \tag{A6}$$

We say a KEM $\mathcal{K}$ is perfectly correct, if $\epsilon = 0$.

Definition A4 (Correctness of DEMs). *We say that a DEM Π^{sym} is ϵ-correct, if for all messages m in the message space $\mathcal{M}_{\Pi^{sym}}$:*

$$\Pr[\mathsf{Dec}(k, c) \neq m | K \leftarrow \mathsf{KeyGen}, c \leftarrow \mathsf{Enc}(k, m)] \leq \tag{A7}$$

We say that a DEM Π^{sym} is perfectly correct, if $\epsilon = 0$.

References

1. Post-Quantum Cryptography. 2017. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography (accessed on 7 March 2022).
2. Harnik, D.; Kilian, J.; Naor, M.; Reingold, O.; Rosen, A. On Robust Combiners for Oblivious Transfer and Other Primitives. In *Advances in Cryptology—EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 96–113.
3. Bindel, N.; Herath, U.; McKague, M.; Stebila, D. Transitioning to a Quantum-Resistant Public Key Infrastructure. In *Post-Quantum Cryptography*; Lange, T., Takagi, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 384–405.
4. Bindel, N.; Brendel, J.; Fischlin, M.; Goncalves, B.; Stebila, D. Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange. In *Post-Quantum Cryptography*; Ding, J., Steinwandt, R., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 206–226.
5. Asmuth, C.; Blakley, G. An Efficient Algorithm for Constructing a Cryptosystem Which is Harder to Break than Two Other Cryptosystems. *Comput. Math. Appl.* **1981**, *7*, 447–450. [CrossRef]
6. Herzberg, A. Folklore, Practice and Theory of Robust Combiners. *J. Comput. Secur.* **2009**, *17*, 159–189. [CrossRef]
7. Zhang, C.; Cash, D.; Wang, X.; Yu, X.; Chow, S.S.M. Combiners for Chosen-Ciphertext Security. In *Computing and Combinatorics*; Dinh, T.N., Thai, M.T., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 257–268.
8. Hohenberger, S.; Lewko, A.; Waters, B. Detecting Dangerous Queries: A New Approach for Chosen Ciphertext Security. In *Advances in Cryptology—EUROCRYPT 2012*; Pointcheval, D., Johansson, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 663–681.
9. Dodis, Y.; Katz, J. Chosen-Ciphertext Security of Multiple Encryption. In *Theory of Cryptography*; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 188–209.
10. Giacon, F.; Heuer, F.; Poettering, B. KEM Combiners. In *Public-Key Cryptography—PKC 2018*; Abdalla, M., Dahab, R., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 190–218.
11. Kampanakis, P.; Panburana, P.; Daw, E.; Van Geest, D. The Viability of Post-Quantum X.509 Certificates. In *IACR Cryptology ePrint Archive, Report 2018/063*; 2018. Available online: http://www.eprint.mirror.cypherpunks.ru/2018/063.pdf (accessed on 4 March 2022).
12. Braithwaite, M. Google Security Blog: Experimenting with Post-Quantum Cryptography. 2016. Available online: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html (accessed on 12 October 2021).
13. Langley, A. Intent to Implement and Ship: CECPQ1 for TLS, 2016. Google Group. Available online: https://groups.google.com/a/chromium.org/forum/#!topic/security-dev/DS9pp2U0SAc (accessed on 12 October 2021).
14. Cramer, R.; Shoup, V. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. *SIAM J. Comput.* **2004**, *33*, 167–226. [CrossRef]
15. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed.; Cambridge University Press: Cambridge, UK, 2011.

16. Bellare, M.; Rogaway, P. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security—CCS '93, Fairfax, VA, USA, 3–5 November 1993; ACM: New York, NY, USA, 1993; pp. 62–73. [CrossRef]
17. Boneh, D.; Dagdelen, Ö.; Fischlin, M.; Lehmann, A.; Schaffner, C.; Zhandry, M. Random Oracles in a Quantum World. In *Advances in Cryptology—ASIACRYPT 2011*; Lee, D.H., Wang, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 41–69.
18. Unruh, D. Revocable Quantum Timed-Release Encryption. *J. ACM* **2015**, *62*, 49:1–49:76. [CrossRef]
19. Ambainis, A.; Hamburg, M.; Unruh, D. Quantum Security Proofs Using Semi-classical Oracles. In *Advances in Cryptology—CRYPTO 2019*; Boldyreva, A., Micciancio, D., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 269–295.
20. Zhandry, M. Secure Identity-Based Encryption in the Quantum Random Oracle Model. In *Advances in Cryptology—CRYPTO 2012*; Safavi-Naini, R., Canetti, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 758–775.
21. Herranz, J.; Hofheinz, D.; Kiltz, E. Some (in)sufficient Conditions forSecure Hybrid Encryption. *Inf. Comput.* **2010**, *208*, 1243–1257. [CrossRef]