Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange

Nina Bindel[1] Jacqueline Brendel[1] Marc Fischlin[1] Brian Goncalves[2] Douglas Stebila[3]

[1] *Technische Universität Darmstadt, Darmstadt, Germany*
[2] *McMaster University, Hamilton, Ontario, Canada*
[3] *University of Waterloo, Waterloo, Ontario, Canada*

nbindel@cdc.informatik.tu-darmstadt.de, {jacqueline.brendel, marc.fischlin}@cryptoplexity.de
goncalb@mcmaster.ca, douglas@stebila.ca

September 24, 2018

Abstract

Concerns about the impact of quantum computers on currently deployed public key cryptography have instigated research into not only quantum-resistant cryptographic primitives but also how to transition applications from classical to quantum-resistant solutions. One approach to mitigate the risk of quantum attacks and to preserve common security guarantees are *hybrid* schemes, which combine classically secure and quantum-resistant schemes. Various academic and industry experiments and draft standards related to the Transport Layer Security (TLS) protocol already use some form of hybrid key exchange; however sound theoretical approaches to substantiate the design and security of such hybrid key exchange protocols are missing so far.

We initiate the modeling of hybrid authenticated key exchange protocols. We consider security against adversaries with varying levels of quantum power over time, such as adversaries who may become quantum in the future or are quantum in the present. We reach our goal using a three-step approach: First, we introduce security notions for key encapsulation mechanisms (KEMs) that enable a fine-grained distinction between different quantum scenarios. Second, we propose several combiners for constructing hybrid KEMs that correspond closely to recently proposed Internet-Drafts for hybrid key exchange in TLS 1.3. Finally, we present a provably sound design for hybrid key exchange using KEMs as building blocks.

Contents

1 Introduction

Research into cryptographic algorithms that could resist attacks by quantum computers is a significant field of current research. Even after new algorithms have been agreed upon, history shows that transitioning applications and protocols to use new algorithms can be a long and difficult process. Backwards compatibility has to be maintained without introducing the risk of downgrade attacks, and the adoption rate of new versions is very slow. An additional obstacle for the post-quantum transition is the uncertainty about the hardness of post-quantum assumptions due to their relative novelty. Parameter choices for post-quantum schemes are not yet reliable[1] and evolving cryptanalysis may yet show them to be vulnerable even against classical attacks. So we find ourselves in a predicament: on the one hand, the demand to protect today's communication from the potential threat posed by quantum computers and the expected lengthy time frame to complete widespread deployment of new algorithms, call for beginning the transition sooner rather than later; on the other hand we are not sufficiently confident in the concrete security of post-quantum schemes for immediate deployment.

Hybrid schemes and robust combiners. So-called hybrid schemes offer a solution for this dilemma: they combine two or more algorithms of the same kind such that the combined scheme is secure as long as one of the two components remains secure. The study of such schemes in the symmetric setting dates back to work by Even and Goldreich [26]. In the public key setting, work by Zhang et al. [48] and Dodis and Katz [25] examined the security of using multiple IND-CCA-secure public key encryption schemes. Harnik et al. [30] defined the term *robust combiner* to formalize such combinations, and the case of combiners for oblivious transfer, with a sketch of a combiner for key agreement. Combiners for other primitives have since followed, including Bindel et al. [12] on hybrid digital signatures. Most relevant to our setting of key exchange and KEMs is the recent work by Giacon et al. [29] which considers various KEM combiners. While the work by Giacon et al. [29] on KEM combiners is an important first step towards constructing hybrid KEMs, their solutions focus solely on classical adversaries. Since the advent of quantum computing and thus the introduction of more powerful adversaries is an important motivation for investigating hybrid key exchange (and thus, implicitly, also KEM combiners), quantum security analyses of hybrid schemes is not to be neglected; in particular because most of the constructions of [29] use idealized assumptions such as random oracles that might not immediately transfer to the quantum setting [13]. Moreover, the (quantum) security of hybrid authenticated key exchange remains unresolved in [29]. An alternative recent approach to model security of protocols in which a component fails is the breakdown-resilience model of Brendel, Fischlin, and Günther [20].

There is appetite in academia [17, 16] and industry for hybrid key exchange in particular. In 2016 Google temporarily tested a hybrid key exchange ciphersuite "CECPQ1" combining elliptic curve Diffie–Hellman (ECDH) and ring-LWE key exchange (specifically, NewHope [3]) in the Transport Layer Security (TLS) stack of an experimental build of its Chrome browser [19, 36]. Microsoft Research [23] and Cloudflare [44] have also expressed interest in hybrid key exchange, and several Internet-Drafts have been submitted to the IETF's TLS working group on hybrid key exchange [42, 46].

Quantum security. Designing quantum-resistant cryptographic schemes requires not only quantum-hard mathematical assumptions, but also appropriate security definitions and proofs. Boneh et al. [13] initiated the study of security of classical public key primitives in the quantum random oracle model, where the locally quantum adversary can access the random oracle in superposition. A line of subsequent work by Boneh, Zhandry, and others [47, 15, 14] extends security definitions of various cryptographic primitives to the case of fully quantum adversaries, i.e., where the adversary's interaction with any other oracles (e.g.,

[1] For example, Albrecht et al. [2] summarize and compare different hardness estimations of instances of the LWE and NTRU problems used in Round 1 submissions to the NIST Post-Quantum Cryptography Standardization [38]. Their results show that depending on the estimation method the differences of bit hardness are up to several hundred bits.

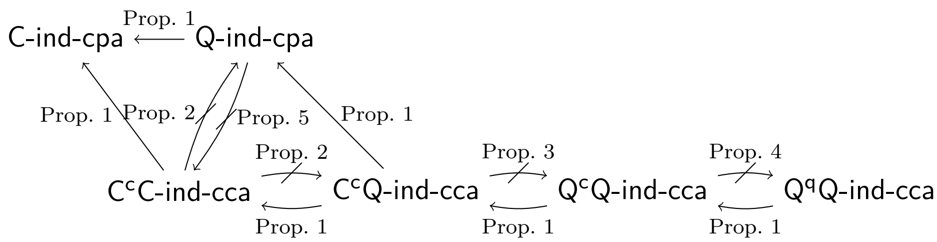

Figure 1: Implications ($\rightarrow$) and separations ($\not\rightarrow$) between indistinguishability-based security notions for KEMs wrt. two-stage adversaries.

the decryption oracle for indistinguishability under chosen-ciphertext-attacks of public key encryption, the signing oracle for unforgeability of digital signatures) can also be in superposition. A recent work by Bindel et al. [12] gives a hierarchy of intermediate security notions, where the adversary may be classical during some portions of the security experiment, and quantum in others, to capture the transition from fully classical to fully quantum security.

Our Contributions. We observe that, despite the strong interest by industry in hybrid key exchange, there has been little academic investigation of the design and security of such schemes. Since early prototypes often become de facto standards, it is important to develop solid theoretical foundations for hybrid key exchange and KEMs at an early stage, especially considering the presence of quantum adversaries. Our work bridges the gap for quantum-resistant hybrid KEMs and extends the foundations to treat hybrid authenticated key exchange protocols: We give new security models both for KEMs and authenticated key exchange protocols that account for adversaries with different levels of quantum capabilities in the security experiment. Furthermore, we examine several combiners for KEMs and prove their robustness. These include a new combiner, called XOR-then-MAC combiner, which is based on minimal assumptions and is—to the best of our knowledge—the first KEM combiner construction which is provably secure against fully quantum adversaries. We also discuss a nested dual PRF combiner closely related to the key schedule used in TLS 1.3 [40]. We then proceed to show how hybrid KEMs can be used to construct hybrid authenticated key exchange protocols. In more detail our contributions are as follows.

Hierarchy of KEM security definitions. We define a family of new security notions for KEMs. Following the approach of Bindel et al. [12] for signature schemes, we adapt the security experiment for indistinguishability under chosen-ciphertext attack (IND-CCA) to distinguish between classical and quantum adversarial capabilities at several key points: the adversary's local computational power during interaction with the decapsulation oracle; whether or not an adversary can make decapsulation queries in superposition; and the adversary's local computational power later, after it can no longer interact with the decapsulation oracle. We represent the three choices as X, y, and Z respectively, and abbreviate a combination as $\mathsf{X^yZ}$-ind-cca. This leads to four different levels: fully classical adversaries (denoted $\mathsf{X^yZ} = \mathsf{C^cC}$); "future-quantum" ($\mathsf{C^cQ}$), where the adversary is classical today but gains quantum power later; "post-quantum" ($\mathsf{Q^cQ}$) where the locally quantum adversary still interacts classically with the decapsulation oracle; and "fully quantum" ($\mathsf{Q^qQ}$), where all computation and interaction can be quantum. As summarized in Figure 1, we show that these different security notions form a strict hierarchy. Unless stated otherwise, the following constructions in the paper focus on providing security against $\mathsf{Q^cQ}$ adversaries, excluding the fully-quantum scenario. This "restriction" is natural as hybrid solutions are intended to secure the transition to the post-quantum setting.

KEM combiners. We present three KEM combiners and show their robustness for $\mathsf{C^cC}$, $\mathsf{C^cQ}$, and $\mathsf{Q^cQ}$ adversaries; all these proofs are in the standard model and do not rely on classical or quantum random oracles.

- XtM: The XOR-then-MAC combiner XtM computes the session key k of the resulting hybrid KEM as the first half of $k_1 \oplus k_2$, where k_1 and k_2 are the session keys of the two input KEMs. Additionally, the XOR-then-MAC combiner augments the ciphertexts with a MAC tag $\mathsf{MAC}_{K_{\mathsf{mac}}}(c_1 \| c_2)$, where K_{mac} is the second half of $k_1 \oplus k_2$, and c_1 and c_2 are the ciphertexts of the two input KEMs. XtM uses the lowest number of cryptographic assumptions, as it relies solely on the security of one of the two combined KEMs and the (one-time) existential unforgeability of the message authentication scheme; such a MAC can be built unconditionally and efficiently using universal hash functions.
  We also discuss that the XtM combiner achieves full quantum resistance ($\mathsf{Q}^{\mathsf{q}}\mathsf{Q}$) if one of the input KEMs has this property and the MAC is $\mathsf{Q}^{\mathsf{c}}\mathsf{Q}$ secure, where the MAC can again be built unconditionally. To the best of our knowledge this is the first security proof for a KEM combiner in this setting.
- $\mathsf{dualPRF}$: The dual PRF combiner computes k as $\mathsf{PRF}(\mathsf{dPRF}(k_1, k_2), c_1 \| c_2)$. In a dual PRF dPRF, the partial functions $\mathsf{dPRF}(k_1, \cdot)$ and $\mathsf{dPRF}(\cdot, k_2)$ are both assumed to be PRFs (and thus be indistinguishable from random functions).
  This combiner is motivated by the key derivation function used in TLS 1.3 [40], which acts both as an extractor (like HKDF's extraction algorithm) and as a pseudorandom function (like HMAC in HKDF), and models how Whyte et al.'s hybrid TLS 1.3 proposal derives the combined session key [46] by concatenating both session keys prior to key derivation.
- The nested combiner N, computes k as $\mathsf{PRF}(\mathsf{dPRF}(F(k_1), k_2), c_1 \| c_2)$. It is motivated by Schanck and Stebila's hybrid TLS 1.3 proposal which derives the combined session key by feeding each component into an extended TLS 1.3 key schedule [42].

Hybrid key exchange. Our third contribution is to show how to build hybrid authenticated key exchange from hybrid KEMs. Our construction relies on Krawczyk's SigMA-compiler [35] using signatures and MACs to authenticate and lift the protocol to one that is secure against active adversaries. The intriguing question here is which security properties the involved primitives need to have in order to achieve resistance against the different levels of adversarial quantum power. Intuitively, the "weakest primitive" determines the overall security of the compiled protocol. However, as we will show in Section 4, this intuition is not entirely correct for partially quantum adversaries.

2 Key Encapsulation Mechanisms and Their Security

In this section, we adjust the basic definitions for key encapsulation mechanisms and their indistinguishability-based security notions to the partially and fully quantum adversary setting. Furthermore we establish the relations between these different notions of security.

A *key encapsulation mechanism* is a triple of algorithms $\mathcal{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ and a corresponding key space K.

1. The probabilistic *key generation* algorithm $\mathsf{KeyGen}()$ returns a public/secret-key pair (pk, sk).
2. The probabilistic *encapsulation* algorithm $\mathsf{Encaps}(pk)$ takes as input a public key pk and outputs a ciphertext c as well as a key $k \in K$.
3. The deterministic *decapsulation* algorithm $\mathsf{Decaps}(sk, c)$ takes as input a secret key sk and a ciphertext c and returns a key $k \in K$ or $\perp$, denoting failure.

A KEM $\mathcal{K}$ is ϵ*-correct* if for all $(sk, pk) \leftarrow \mathsf{KeyGen}()$ and $(c, k) \leftarrow \mathsf{Encaps}(pk)$, it holds that $\Pr[\mathsf{Decaps}(sk, c) \neq k] \leq \epsilon$. We say it is *correct* if $\epsilon = 0$.

The security of KEMs is defined in terms of the indistinguishability of the session key against chosen-plaintext (IND-CPA) and chosen-ciphertext (IND-CCA) adversaries. In the traditional IND-CPA experiment of KEMs, the challenger $\mathcal{C}$ generates keys $(sk, pk) \leftarrow \mathsf{KeyGen}()$, computes $(c^*, \kappa_0^*) \leftarrow \mathsf{Encaps}(pk)$, and samples

$\text{Expt}_{\mathcal{K}}^{\text{Z-ind-cpa}}(\mathcal{A})$:

$H \leftarrow_{\$} \mathcal{H}_{\mathcal{K}}$
$q_H \leftarrow 0$
$(sk, pk) \leftarrow \mathsf{KeyGen}()$
$(c^*, \kappa_0^*) \leftarrow \mathsf{Encaps}(pk)$
$\kappa_1^* \leftarrow_{\$} K$
$b \leftarrow_{\$} \{0, 1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_H^{\mathsf{Z}}(\cdot)}(pk, c^*, \kappa_b^*)$
return $[\![b = b']\!]$

$\mathcal{O}_H^{\mathsf{C}}(x)$:

$q_H \leftarrow q_H + 1$
Return $H(x)$

$\mathcal{O}_H^{\mathsf{Q}}(\sum_{x,t,z} \psi_{x,t,z} |x, t, z\rangle)$:

$q_H \leftarrow q_H + 1$
Return state $\sum_{x,t,z} \psi_{x,t,z} |x, t \oplus H(x), z\rangle$

Figure 2: Security experiment for indistinguishability of a KEM $\mathcal{K}$ under chosen-plaintext attack against a classical ($\mathsf{Z} = \mathsf{C}$) or quantum ($\mathsf{Z} = \mathsf{Q}$) adversary $\mathcal{A}$ in the classical or quantum random oracle model.

κ_1^* uniformly at random from the key space K and a random bit b. The adversary $\mathcal{A}$ is given c^*, κ_b^*, and pk, and is asked to output a bit b', indicating whether it believes it received the key corresponding to c^* or a random value. The adversary wins if it outputs the correct bit b', i.e., if $b' = b$.

In the traditional IND-CCA experiment the adversary $\mathcal{A}$ additionally has access to a decapsulation oracle, which returns the decapsulation of any ciphertext not equal to the challenge ciphertext c^*.

2.1 Indistinguishability Security of KEMs Against Partially or Fully Quantum Adversaries

We adapt the traditional definitions of IND-CPA and IND-CCA security of KEMs for quantum adversaries. We give a brief introduction to the quantum computation knowledge used in this paper in the appendix in Section A.

2.1.1 IND-CPA security against quantum adversaries.

In the standard model, the IND-CPA adversary $\mathcal{A}$ is simply treated as a quantum algorithm. For IND-CPA in the random oracle model, one can choose whether the adversary should have classical or quantum access to the random oracle [13] or not. While both options lead to valid definitions, giving the adversary quantum access to the random oracle is clearly the stronger option; moreover, it seems sensible to allow the adversary quantum access to the random oracle since the random oracle is meant to capture idealized public hash functions that can be implemented by an adversary in practice. Depending on the adversary's power this implementation can be classical or quantum.

In Figure 2, we give a unified definition of classical and quantum IND-CPA, denoted Z-ind-cpa, where Z is either C (for classical) or Q (for quantum). We define the corresponding advantage $\mathsf{Adv}_{\mathcal{K}}^{\text{Z-ind-cpa}}(\mathcal{A}) = \left|\Pr\left[\text{Expt}_{\mathcal{K}}^{\text{Z-ind-cpa}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|$.

For consistency with the IND-CCA case, where we need to distinguish the cases when the adversary has quantum power and how it interacts with the decapsulation oracle, we occasionally also use the notation $\mathsf{X}^{\mathsf{y}}\mathsf{Z}$-ind-cpa instead of Z-ind-cpa in the IND-CPA case. In such cases we sometimes refer to both as $\mathsf{X}^{\mathsf{y}}\mathsf{Z}$-ind-atk with $\mathsf{atk} \in \{\mathsf{cpa}, \mathsf{cca}\}$. We stress, however, that X and y in the cpa case are irrelevant, and are there solely for notational uniformity.

2.1.2 IND-CCA security against partially or fully quantum adversaries.

Previous works on security of KEMs against quantum adversaries, such as that of Hofheinz, Hövelmanns, and Kiltz [31], consider a quantum adversary that has local quantum power and can query the random oracle in superposition. Bindel et al. [12] consider partially quantum adversaries to model the security of signature schemes against quantum adversaries. We consider that approach in the context of IND-CCA security of KEMs to enable more distinctions when modeling chosen-ciphertext attacks for KEMs. In particular, our model allows to distinguish between adversaries with evolving quantum capabilities over time.

For example, one may believe that no adversary today has a sufficiently powerful quantum computer to break any cryptographic assumption, and that it may still be some decades before a full-fledged quantum computer is built. In that scenario, one would want to protect today's communications against attacks in which the (currently classical) attacker records encrypted communications today and, once a quantum computer is available, attempts to extract the encryption keys from the corresponding collected key exchange transcripts.

Alternatively, one might not feel comfortable excluding the possibility that powerful quantum computers exist already today or in the nearer future. In this case, stronger security guarantees are needed to protect the communications, as a locally quantum adversary may already interfere with the deployed protocols and could for example break classical signatures used for authentication.

To capture these cases for hybrid signatures, Bindel et al. [12] introduced a two-stage security experiment for unforgeability of signature schemes; we transfer this adversary notion to key encapsulation mechanisms.

For IND-CCA security of KEMs, we consider four ways in which the adversary could act quantumly: (i) the adversary could locally be running a classical or quantum computer during the stage in which the adversary can interact with the decapsulation oracle; (ii) the adversary's interaction with the decapsulation oracle could be classical or quantum; (iii) the adversary could locally be running a classical or quantum computer after the interaction with the decapsulation oracle has finished; and (iv) the adversary's interaction with the random oracle (if any) could be classical or quantum. As we did above for IND-CPA security, we define the adversary's interaction with the random oracle to be quantum whenever the adversary is quantum, eliminating this fourth option.

To model this, we consider a two-stage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, in which $\mathcal{A}_1$ has access to the decapsulation oracle, then terminates and passes a state to the second-stage adversary $\mathcal{A}_2$, which does not have access to the decapsulation oracle. Let $\mathsf{X}, \mathsf{Z} \in \{\mathsf{C}, \mathsf{Q}\}$ and $\mathsf{y} \in \{\mathsf{c}, \mathsf{q}\}$. We will use the terminology "$\mathsf{X}^\mathsf{y}\mathsf{Z}$ adversary" to denote that $\mathcal{A}_1$ is either classical ($\mathsf{X} = \mathsf{C}$) or quantum ($\mathsf{X} = \mathsf{Q}$), that $\mathcal{A}_1$'s access to its decapsulation oracle is either classical ($\mathsf{y} = \mathsf{c}$) or quantum ($\mathsf{y} = \mathsf{q}$), and that $\mathcal{A}_2$ is either classical ($\mathsf{Z} = \mathsf{C}$) or quantum ($\mathsf{Z} = \mathsf{Q}$). In the random oracle model, the adversary can query the random oracle in superposition, if it is quantum; this is independent of y but depends on X and Z.

Not all combinations of classical and quantum adversaries in the two-stage setting are meaningful in a real-world context. We consider the following configurations of two-stage $\mathsf{X}^\mathsf{y}\mathsf{Z}$ adversaries to be relevant:

- $\mathsf{C}^\mathsf{c}\mathsf{C}$ security corresponds to the scenario with a purely **classical** adversary with classical access to all oracles. This corresponds to the traditional IND-CCA security notion.
- $\mathsf{C}^\mathsf{c}\mathsf{Q}$ security refers to a scenario with a currently classical but potentially **future quantum** adversary. In particular, that means that the adversary is classical as long as the adversary has access to the decapsulation oracle; eventually the adversary gains local quantum computing power, but by this time the adversary relinquishes access to the decapsulation.
- $\mathsf{Q}^\mathsf{c}\mathsf{Q}$ security corresponds to the scenario with an adversary that always has local quantum computing power, but interacts with the active system (in the first stage) only using classical queries. This kind of setting is for example considered in [31] and is commonly referred to as the **post-quantum** setting.
- $\mathsf{Q}^\mathsf{q}\mathsf{Q}$ security corresponds to a scenario with a **fully quantum** adversary with quantum access to all oracles in the first stage.[2]

It is notation-wise convenient to define an order for the notions, with $\mathsf{Q} \geq \mathsf{C}$ and $\mathsf{q} \geq \mathsf{c}$, consequently

[2] Our fully quantum $\mathsf{Q}^\mathsf{q}\mathsf{Q}$ model is different from [1] since our challenge ciphertext c^* is classical, whereas [1] considers quantum challenge ciphertexts

$\underline{\mathrm{Expt}_{\mathcal{K}}^{\mathsf{X^yZ\text{-}ind\text{-}cca}}(\mathcal{A})}$:

$H \leftarrow_\$ \mathcal{H}_{\mathcal{K}}$
$q_D \leftarrow 0$, $q_H \leftarrow 0$
$(sk, pk) \leftarrow \mathsf{KeyGen}()$
$(c^*, \kappa_0^*) \leftarrow \mathsf{Encaps}(pk)$
$\kappa_1^* \leftarrow_\$ K$
$b \leftarrow_\$ \{0,1\}$
$st \leftarrow \mathcal{A}_1^{\mathcal{O}_H^{\mathsf{X}}(\cdot), \mathcal{O}_D^{\mathsf{y}}(\cdot)}(pk, c^*, \kappa_b^*)$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}_H^{\mathsf{Z}}(\cdot)}(st)$
return $[\![b = b']\!]$

$\underline{\mathsf{Decaps}^{\perp}(sk, c, c^*)}$:

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

$\underline{\mathcal{O}_D^{\mathsf{c}}(c)}$:

$q_D \leftarrow q_D + 1$
return $\mathsf{Decaps}^{\perp}(sk, c, c^*)$

$\underline{\mathcal{O}_D^{\mathsf{q}}(\sum_{c,t,z} \psi_{c,t,z} \, |c, t, z\rangle)}$:

$q_D \leftarrow q_D + 1$
return $\sum_{c,t,z} \psi_{c,t,z} \left| c, t \oplus \mathsf{Decaps}^{\perp}(sk, c, c^*), z \right\rangle$

Figure 3: Security experiment for indistinguishability of a KEM $\mathcal{K}$ under chosen-ciphertext attack against a two-stage $\mathsf{X^yZ}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the classical or quantum random oracle model. $\mathcal{O}_H^{\mathsf{c}}$ and $\mathcal{O}_H^{\mathsf{q}}$ are as in Figure 2.

implying a partial order $\mathsf{X^yZ} \geq \mathsf{U^vW}$ if $\mathsf{X} \geq \mathsf{U}$, $\mathsf{y} \geq \mathsf{v}$, and $\mathsf{Z} \geq \mathsf{W}$, i.e., $\mathsf{Q^qQ} \geq \mathsf{Q^cQ} \geq \mathsf{C^cQ} \geq \mathsf{C^cC}$. Let $\max S$ (resp., $\min S$) denote the set of maximal (resp., minimal) elements of S according to this partial order. Since we usually have a total order on S, i.e., $S \subseteq \{\mathsf{C^cC}, \mathsf{C^cQ}, \mathsf{Q^cQ}, \mathsf{Q^qQ}\}$, we often simply speak of *the* maximal element. For example, it holds that $\mathsf{C^cQ} = \max\{\mathsf{C^cC}, \mathsf{C^cQ}\}$.

Figure 3 shows the security experiment for indistinguishability of keys in a key encapsulation mechanism $\mathcal{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ under chosen-ciphertext attacks for a two-stage $\mathsf{X^yZ}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ in the classical or quantum random oracle model; the standard model notion can be obtained by omitting the hash oracles. For every notion $\mathsf{X^yZ\text{-}ind\text{-}cca}$, we define the corresponding advantage $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{X^yZ\text{-}ind\text{-}cca}}(\mathcal{A}) = \left| \Pr\left[\mathrm{Expt}_{\mathcal{K}}^{\mathsf{X^yZ\text{-}ind\text{-}cca}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2} \right|$.

2.2 Relations Between Indistinguishability Security Notions

Similarly to [12], and as described in Figure 1, the various indistinguishability notions for KEMs are related to each other through a series of implications and separations as we show in this section.

Proposition 1 (Implications)**.** *Let $\mathcal{K}$ be a key encapsulation mechanism. If $\mathcal{K}$ is* $\mathsf{Q^qQ\text{-}ind\text{-}cca}$ *secure, then $\mathcal{K}$ is also* $\mathsf{Q^cQ\text{-}ind\text{-}cca}$ *secure. If $\mathcal{K}$ is* $\mathsf{Q^cQ\text{-}ind\text{-}cca}$ *secure, then $\mathcal{K}$ is also* $\mathsf{C^cQ\text{-}ind\text{-}cca}$ *secure. If $\mathcal{K}$ is* $\mathsf{C^cQ\text{-}ind\text{-}cca}$ *secure, then $\mathcal{K}$ is also* $\mathsf{C^cC\text{-}ind\text{-}cca}$ *secure and* $\mathsf{Q\text{-}ind\text{-}cpa}$ *secure. If $\mathcal{K}$ is* $\mathsf{Q\text{-}ind\text{-}cpa}$ *secure or* $\mathsf{C^cC\text{-}ind\text{-}cca}$ *secure, then $\mathcal{K}$ is also* $\mathsf{C\text{-}ind\text{-}cpa}$ *secure.*

Proof. The proof is straightforward since every classical adversary can be seen as a quantum adversary that forgoes its additional quantum power. Furthermore a $\mathsf{Q^cQ\text{-}ind\text{-}cca}$ adversary can be seen as a $\mathsf{Q^qQ\text{-}ind\text{-}cca}$ adversary that does not use superposition queries to the oracle. It thus holds that $\mathsf{Adv}_{\mathcal{K}}^{\mathsf{Q^qQ\text{-}ind\text{-}cca}}(\mathcal{A}) \geq \mathsf{Adv}_{\mathcal{K}}^{\mathsf{Q^cQ\text{-}ind\text{-}cca}}(\mathcal{A}) \geq \mathsf{Adv}_{\mathcal{K}}^{\mathsf{C^cQ\text{-}ind\text{-}cca}}(\mathcal{A}) \geq \mathsf{Adv}_{\mathcal{K}}^{\mathsf{C^cC\text{-}ind\text{-}cca}}(\mathcal{A})$ and $\mathsf{Q\text{-}ind\text{-}cpa} \geq \mathsf{C\text{-}ind\text{-}cpa}$. Moreover, it trivially holds that $\mathsf{C^cC\text{-}ind\text{-}cca} \geq \mathsf{C\text{-}ind\text{-}cpa}$. □

In the following we show that these implications are in fact strict by showing separations between the different notions. We start with Proposition 2 which states essentially that there exist KEMs that are classically secure ($\mathsf{C^cC}$), but that become insecure once adversaries gain quantum power at a later point in time ($\mathsf{C^cQ}$).

Proposition 2 ($\mathsf{C^cC\text{-}ind\text{-}cca} \not\Longrightarrow \mathsf{Q\text{-}ind\text{-}cpa}, \mathsf{C^cQ\text{-}ind\text{-}cca}$)**.** *In the classical random oracle model, assuming RSA is a one-way function (for classical algorithms), there exists a* $\mathsf{C^cC\text{-}ind\text{-}cca}$ *secure KEM in the random oracle model that is neither* $\mathsf{Q\text{-}ind\text{-}cpa}$ *secure nor* $\mathsf{C^cQ\text{-}ind\text{-}cca}$ *secure.*

Proposition 2 follows immediately from the fact that the KEM based on RSA-OAEP is $\mathsf{C^cC}$-ind-cca secure in the random oracle model [10]. However, a $\mathsf{C^cQ}$ adversary with local access to quantum computing power in the second stage can run Shor's algorithm to factor the RSA modulus and recover the decapsulation key to win the Q-ind-cpa or $\mathsf{C^cQ}$-ind-cca experiments.

Next, we show that there exist KEMs that are secure as long as only classical adversaries interact with the decapsulation oracle ($\mathsf{C^cQ}$) but that become insecure in the post-quantum setting ($\mathsf{Q^cQ}$).

Proposition 3 ($\mathsf{C^cQ}$-ind-cca $\not\Longrightarrow$ $\mathsf{Q^cQ}$-ind-cca)**.** *Let $\mathcal{K}$ be an* $\mathsf{C^cQ}$-ind-cca *secure KEM and $\mathcal{K}_{\mathcal{BD}}$ be a* C-ind-cpa *secure KEM for which there is an efficient quantum algorithm that recovers the session key (i.e., it is not* Q-ow-cpa*). Then there exists a key encapsulation mechanism $\mathcal{K}'$ that is* $\mathsf{C^cQ}$-ind-cca *secure but not* $\mathsf{Q^cQ}$-ind-cca *secure.*

Proof. In the following, we construct the separating key encapsulation mechanism $\mathcal{K}' = (\mathsf{KeyGen}', \mathsf{Encaps}', \mathsf{Decaps}')$, which is $\mathsf{C^cQ}$-ind-cca secure, but not $\mathsf{Q^cQ}$-ind-cca secure. The idea is to include a backdoor which is only available if the first-stage adversary has access to local quantum computing power. The KEM $\mathcal{K}'$ is defined as described in Figure 4, where $\mathcal{K}_{\mathcal{BD}} = (\mathsf{KeyGen}_{\mathcal{BD}}, \mathsf{Encaps}_{\mathcal{BD}}, \mathsf{Decaps}_{\mathcal{BD}})$ is a C-ow-cpa secure KEM which can be broken with local quantum power. An example is again the RSA-OAEP based KEM.

$\mathsf{KeyGen}'()$:
1 $(pk, sk) \leftarrow \mathsf{KeyGen}()$
2 $(pk_{\mathcal{BD}}, sk_{\mathcal{BD}}) \leftarrow \mathsf{KeyGen}_{\mathcal{BD}}()$
3 $(c_{\mathcal{BD}}, k_{\mathcal{BD}}) \leftarrow \mathsf{Encaps}_{\mathcal{BD}}(pk_{\mathcal{BD}})$
4 return $(pk', sk') = ((pk, pk_{\mathcal{BD}}, c_{\mathcal{BD}}), (sk, k_{\mathcal{BD}}))$

$\mathsf{Encaps}'(pk, pk_{\mathcal{BD}}, c_{\mathcal{BD}})$:
1 $(c, k) \leftarrow \mathsf{Encaps}(pk)$
2 return (c, k)

$\mathsf{Decaps}'(sk, k_{\mathcal{BD}}, c)$:
1 If $c = k_{\mathcal{BD}}$ then return sk
2 Else return $\mathsf{Decaps}(sk, c)$

Figure 4: Description of separating KEM $\mathcal{K}'$ which is $\mathsf{C^cQ}$-ind-cca secure, but not $\mathsf{Q^cQ}$-ind-cca secure

We start by showing that $\mathcal{K}'$ is $\mathsf{C^cQ}$-ind-cca secure. Assume it is not, i.e., there exists an efficient $\mathsf{C^cQ}$ adversary $\mathcal{A}$ that can break the IND-CCA security of $\mathcal{K}'$. Then there exist an adversary $\mathcal{B}$ that can break the $\mathsf{C^cQ}$-ind-cca security of $\mathcal{K}$. The adversary $\mathcal{B}$ receives its challenge, say, (pk, c^*, κ_b^*). It runs Steps 2-3 of KeyGen' by itself, and sends $((pk, pk_{\mathcal{BD}}, c_{\mathcal{BD}}), c^*, \kappa_b^*)$ as input to $\mathcal{A}$. Whenever $\mathcal{A}$ (in its first stage) queries the decapsulation oracle $\mathcal{O}_{D'}(\cdot)$ on some ciphertext $c \neq k_{\mathcal{BD}}$, algorithm $\mathcal{B}$ forwards the query to its own decapsulation oracle $\mathcal{O}_D(\cdot)$. If the adversary queries the oracle on $k_{\mathcal{B}}$, then $\mathcal{B}$ returns $\perp$. Since the KEM $\mathcal{K}_{\mathcal{BD}}$ is ow-cpa and we are still in the first phase, any query of $\mathcal{A}$ about $k_{\mathcal{BD}}$ would immediately refute the one-wayness via a black-box reduction. Hence, $\mathcal{B}$'s simulation is correct, except if $\mathcal{A}$ breaks one-wayness of $\mathcal{K}_{\mathcal{BD}}$.

Next we show that $\mathcal{K}'$ is not $\mathsf{Q^cQ}$-ind-cca secure. The first-stage adversary has access to local quantum computing power. With this it can break the classically secure KEM $\mathcal{K}_{\mathcal{BD}}$ to obtain $k_{\mathcal{BD}}$ from $c_{\mathcal{BD}}$ attached to the public key. By construction of $\mathcal{K}'$, the decapsulation oracle queried on $k_{\mathcal{BD}}$ returns the secret key sk of $\mathcal{K}$, allowing it to recover the encapsulated key. □

We now show that post-quantum secure KEMs ($\mathsf{Q^cQ}$) are not necessarily secure in the fully quantum setting where the adversary has quantum access to the decapsulation oracle ($\mathsf{Q^qQ}$).

Proposition 4 ($\mathsf{Q^cQ}$-ind-cca $\not\Longrightarrow$ $\mathsf{Q^qQ}$-ind-cca)**.** *Let λ be the security parameter. Assume that there exists a quantum secure family of pseudorandom permutations. Furthermore, assume there exists a* $\mathsf{Q^cQ}$-ind-cca *secure KEM $\mathcal{K}$ whose ciphertexts are at least 3λ bits long. Then there exists a KEM $\mathcal{K}'$ that is* $\mathsf{Q^cQ}$-ind-cca *secure but not* $\mathsf{Q^qQ}$-ind-cca *secure.*

Finally we note that IND-CPA security in the quantum setting is not necessarily enough to show classical IND-CCA security:

Proposition 5 (Q-ind-cpa $\not\Rightarrow$ $\mathsf{C^cC}$-ind-cca)**.** *Assume there exists a* Q-ind-cpa *secure KEM* $\mathcal{K}$*. Then there exists a KEM* $\mathcal{K}'$ *that is* Q-ind-cpa *secure but not* $\mathsf{C^cC}$-ind-cca *secure.*

Proof. Let $\mathcal{K} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ be a Q-ind-cpa secure KEM, then we can construct a KEM $\mathcal{K}'$ that is Q-ind-cpa but not $\mathsf{C^cC}$-ind-cca secure. The KEM $\mathcal{K}' = (\mathsf{KeyGen}', \mathsf{Encaps}', \mathsf{Decaps}')$ is defined as described in Figure 5.

$\mathsf{KeyGen}'()$:	$\mathsf{Encaps}'(pk)$:	$\mathsf{Decaps}'(sk, c)$:
$(pk, sk) \leftarrow \mathsf{KeyGen}()$	$(c, k) \leftarrow \mathsf{Encaps}(pk)$	If $c = pk$ return sk
return (pk, sk)	return (c, k)	Else return $k \leftarrow \mathsf{Decaps}(sk, c)$

Figure 5: Description of KEM $\mathcal{K}'$.

Clearly $\mathcal{K}'$ is not $\mathsf{C^cC}$-ind-cca secure since it is broken as soon as the public key is asked as a ciphertext to the decapsulation oracle of $\mathcal{K}'$. However, as long as no queries are allowed to the decapsulation oracle, an adversary cannot distinguish $\mathcal{K}$ and $\mathcal{K}'$. Hence, $\mathcal{K}'$ is Q-ind-cpa secure. □

2.3 The Fujisaki–Okamoto Transform

The Fujisaki–Okamoto (FO) transform [27, 28, 24] constructs an IND-CCA secure PKE or KEM from an IND-CPA secure (or one-way-CPA secure) PKE in the random oracle model; analogues that are secure in the quantum random oracle model have been given by Targhi and Unruh [43] and in a modular framework by Hofheinz, Hövelmanns, and Kiltz [31]. These results can also be applied in our two-stage adversary setting. The FO transform converts a C-ow-cpa secure PKE into a $\mathsf{C^cC}$-ind-cca secure PKE (or KEM), in the (classical) random oracle model. The transforms presented in [43, 31] convert a Q-ow-cpa secure PKE into a $\mathsf{Q^cQ}$-ind-cca secure PKE or KEM. It remains an open question how to transform a Q-ind-cpa secure KEM into a $\mathsf{Q^qQ}$-ind-cca secure KEM. We give the formal definition of Z-ow-cpa in the supplementary material in Section B.

3 Practical Combiners for Hybrid Key Encapsulation

In this section, we discuss the use of robust combiners to construct hybrid key encapsulation mechanisms. We propose three combiners motivated by practical applications of hybrid KEMs.

The first combiner, the XOR-then-MAC combiner XtM, uses a simple exclusive-or of the two keys k_1, k_2 of the KEMs but adds a message authentication over the ciphertexts (with a key derived from the encapsulated the keys). Hence, this solution relies solely on the additional assumption of a secure one-time message authentication code which, in turn, can be instantiated unconditionally. The second combiner, dualPRF, relies on the existence of dual pseudorandom functions [5, 4, 8] which provide security if either the key material or the label carries entropy. The HKDF key derivation function is, for example, based on this dual principle. The third combiner, N, is a nested variant of the dual-PRF combiner inspired by the key derivation procedure in TLS 1.3 and the proposal how to augment it for hybrid schemes in [42].

Throughout this section we let $\mathcal{K}_1 = (\mathsf{KeyGen}_1, \mathsf{Encaps}_1, \mathsf{Decaps}_1)$ and $\mathcal{K}_2 = (\mathsf{KeyGen}_2, \mathsf{Encaps}_2, \mathsf{Decaps}_2)$ be two KEMs. Furthermore, we write $\mathcal{C}[\mathcal{K}_1, \mathcal{K}_2] = (\mathsf{KeyGen}_{\mathcal{C}}, \mathsf{Encaps}_{\mathcal{C}}, \mathsf{Decaps}_{\mathcal{C}})$ for the hybrid KEM constructed by one of the three proposals $\mathcal{C} \in \{\mathsf{XtM}, \mathsf{dualPRF}, \mathsf{N}\}$. In all our schemes, $\mathsf{KeyGen}_{\mathcal{C}}$ simply returns the concatenation of the two public keys ($pk \leftarrow (pk_1, pk_2)$) and the two secret keys ($sk \leftarrow (sk_1, sk_2)$).

In this and the following, we focus on proving security against at most post-quantum $\mathsf{Q^cQ}$ adversaries, namely adversaries with classical access to the decapsulation oracle only, omitting $\mathsf{Q^qQ}$-ind-cca security. This is due to the fact that hybrid KEMs and KE solutions are designed to secure the transitional phase until quantum computers become first available. The eventually following widespread deployment of

$\mathsf{Encaps}_{\mathsf{XtM}}(pk_1, pk_2)$:

$(c_1, k_1 \| k_{\mathsf{mac},1}) \leftarrow \mathsf{Encaps}_1(pk_1)$
$(c_2, k_2 \| k_{\mathsf{mac},2}) \leftarrow \mathsf{Encaps}_2(pk_2)$
$k_{\mathsf{kem}} \leftarrow k_1 \oplus k_2$
$k_{\mathsf{mac}} \leftarrow (k_{\mathsf{mac},1}, k_{\mathsf{mac},2})$
$c \leftarrow (c_1, c_2)$
$\tau \leftarrow \mathsf{MAC}_{k_{\mathsf{mac}}}(c)$
return $((c, \tau), k_{\mathsf{kem}})$

$\mathsf{Decaps}_{\mathsf{XtM}}((sk_1, sk_2), ((c_1, c_2), \tau))$:

$k_1' \| k_{\mathsf{mac},1}' \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$,
$k_2' \| k_{\mathsf{mac},2}' \leftarrow \mathsf{Decaps}_2(sk_2, c_2)$
$k_{\mathsf{kem}}' \leftarrow k_1' \oplus k_2'$
$k_{\mathsf{mac}}' \leftarrow (k_{\mathsf{mac},1}', k_{\mathsf{mac},2}')$
if $\mathsf{MVf}_{k_{\mathsf{mac}}'}((c_1, c_2), \tau) = 0$: return $\perp$
else: return k_{kem}'

Figure 6: KEM constructed by the XOR-then-MAC combiner $\mathsf{XtM}[\mathcal{K}_1, \mathcal{K}_2, \mathcal{M}]$ with MAC $\mathcal{M} = (\mathsf{MKG}, \mathsf{MAC}, \mathsf{MVf})$.

quantum computers and cryptography, and thus security against $\mathsf{Q^qQ}$ adversaries, is outside the scope of the post-quantum setting.

3.1 XtM: XOR-then-MAC Combiner

Giacon et al. [29] demonstrate that the plain XOR-combiner, which concatenates the ciphertexts and XORs the individual keys, preserves ind-cpa security. They show that, in general, it does not preserve ind-cca security, e.g., the combiner may become insecure if one of the KEMs is insecure. We note that it is easy to see that this is even true if *both* KEMs are ind-cca secure: Given a challenge ciphertext (c_1^*, c_2^*) the adversary can make two decapsulation requests for (c_1^*, c_2) and (c_1, c_2^*) with fresh ciphertexts $c_1 \neq c_1^*$, $c_2 \neq c_2^*$ for which it knows the encapsulated keys. This allows the adversary to easily recover the challenge key from the answers.

3.1.1 The XOR-then-MAC combiner.

Our approach is to prevent the adversary from mix-and-match attacks by computing a message authentication code over the ciphertexts and attaching it to the encapsulation. For this we require a strongly robust MAC combiner which takes two keys $k_{\mathsf{mac},1}, k_{\mathsf{mac},2}$ as input and provides one-time unforgeability, even if one of the keys is chosen adversarially. We discuss the construction of such MACs later. The combined KEM key is derived as an exclusive or of the leading parts of the two encapsulated keys, $k \leftarrow k_1 \oplus k_2$, and the MAC key $k_{\mathsf{mac}} = (k_{\mathsf{mac},1}, k_{\mathsf{mac},2})$ consisting of the remaining parts of both encapsulated keys. If necessary, the encapsulated keys can be stretched pseudorandomly by the underlying encapsulation schemes first to achieve the desired output length. We depict the resulting hybrid KEM in Figure 6.

Using the XOR-then-MAC combiner in protocols. It may seem that it would be preferable to protect against mix-and-match attacks as above by protecting the derived key directly (by making key derivation depend on both keys and both ciphertexts), as opposed to the approach in the XOR-then-MAC combiner, which appends a MAC to the ciphertext to protect the ciphertext from modification. However, in many practical protocols, the parties often compute a MAC over the transcript to provide integrity and authenticity; an example of this can be found in the `Finished` message in the Transport Layer Security (TLS) protocol. The key for the MAC is usually derived from the session key, and the transcript includes the data for establishing the key, such as the KEM ciphertexts. In these cases, it may be possible to apply the XOR-then-MAC approach without needing to add an extra MAC over the ciphertext, instead relying on the one already present.

$\underline{\mathrm{Expt}^{\mathsf{X^yZ\text{-}OT\text{-}sEUF}}_{\mathcal{M}}(\mathcal{A})}$:

$q_V \leftarrow 0$
$k = (k_1, k_2) \leftarrow \mathsf{MKG}()$
$(m^*, b, k_b^* st) \leftarrow \mathcal{A}_1()$
if $b = 1$ then $k^* \leftarrow (k_1^*, k_2)$ else $k^* \leftarrow (k_1, k_2^*)$
$\tau^* \leftarrow \mathsf{MAC}_{k^*}(m^*)$
$st \leftarrow \mathcal{A}_1^{\mathcal{O}_V^{\mathsf{y}}(\cdot)}(\tau^*)$
$(m', \tau', k_b') \leftarrow \mathcal{A}_2(st)$
if $b = 1$ then $k' \leftarrow (k_1', k_2)$ else $k' \leftarrow (k_1, k_2')$
if $[\mathsf{MVf}_{k'}(m', \tau') = 1]$
$\wedge [(m', \tau') \neq (m^*, \tau^*)]$:
  return 1
else: return 0

$\underline{\mathcal{O}_V^{\mathsf{c}}(m, \tau, k_b')}$:

$q_V \leftarrow q_V + 1$
if $b = 1$ then $k' \leftarrow (k_1', k_2)$ else $k' \leftarrow (k_1, k_2')$
return $\mathsf{MVf}_{k'}(m, \tau)$

$\underline{\mathcal{O}_V^{\mathsf{q}}(\sum_{m,\tau,b,k_b',t,z} \psi_{m,\tau,t,z} |m, \tau, t, z\rangle)}$:

$q_V \leftarrow q_V + 1$
if $b = 1$ then $k' \leftarrow (k_1', k_2)$ else $k' \leftarrow (k_1, k_2')$
return $\sum_{m,\tau,b,k_b',t,z} \psi_{m,\tau,t,z} |m, \tau, b, k_b', t \oplus \mathsf{MVf}_{k'}(m, \tau), z\rangle$

Figure 7: Security experiment for one-time strong existential unforgeability (with multiple verifications) of a two-key MAC $\mathcal{M} = (\mathsf{MKG}, \mathsf{MAC}, \mathsf{MVf})$ against an $\mathsf{X^yZ}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.

3.1.2 Security of MACs.

It suffices to use one-time MACs with multiple verification queries. This means that the adversary can initially choose a message, receives the MAC, and can then make multiple verification attempts for other messages. We require strong unforgeability, meaning the adversary wins if it creates any new valid message-tag pair, even for the same initial message. We use a two-stage version of the definition with an $\mathsf{X^yZ}$ adversary who is of type X while it has y access to the verification oracle and receives the challenge ciphertext. The adversary is of type Z after it no longer has access to the verification oracle. As explained earlier in this paper, the meaningful notions are $\mathsf{X^yZ} \in \{\mathsf{C^cC}, \mathsf{C^cQ}, \mathsf{Q^cQ}, \mathsf{Q^qQ}\}$.

Formally, a MAC is a tuple of algorithms $\mathcal{M} = (\mathsf{MKG}, \mathsf{MAC}, \mathsf{MVf})$ for key generation, MAC tag generation, and MAC tag verification. To capture the strong combiner property of MACs, where the adversary may try to win for a key $k_{\mathsf{mac}} = (k_{\mathsf{mac},1}, k_{\mathsf{mac},2})$ where either $k_{\mathsf{mac},1}$ or $k_{\mathsf{mac},2}$ is chosen by the adversary, we allow the adversary to specify one of the two keys for computing the challenge and for each verification query and in the forgery attempt. The security experiment for $\mathsf{X^yZ\text{-}OT\text{-}sEUF}$ security of such two-keys $\mathcal{M}$ is given in Figure 7.

We discuss possible instantiations for the MAC later, but already note that such MACs can be built without relying on cryptographic assumptions.

3.1.3 Security of the XOR-then-MAC combiner.

We can now show that the XOR-then-MAC combiner is a robust KEM combiner, in the sense that the resulting KEM is as secure as the strongest of the two input KEMs (assuming the MAC is also equally secure). In particular, we show in Theorem 1 that $\mathsf{XtM}[\mathcal{K}_1, \mathcal{K}_2, \mathcal{M}]$ is IND-CCA secure in the post-quantum setting ($\mathsf{Q^cQ}$) if the MAC $\mathcal{M}$ and at least one of the two KEMs is post-quantum IND-CCA secure. In fact, the security offered by the MAC is only required in case of IND-CCA attacks, yielding an even better bound for the IND-CPA case.

Theorem 1 (XOR-then-MAC is robust)**.** *Let* $\mathcal{K}_1$ *be a* $\mathsf{X^cZ\text{-}ind\text{-}atk}$ *secure KEM,* $\mathcal{K}_2$ *a* $\mathsf{U^cW\text{-}ind\text{-}atk}$ *secure KEM, and* $\mathcal{M}$ *is an* $\mathsf{R^cT\text{-}OT\text{-}sEUF}$ *secure MAC, where* $\mathsf{R^cT} = \max\{\mathsf{X^cZ}, \mathsf{U^cW}\}$*. Then* $\mathsf{XtM}[\mathcal{K}_1, \mathcal{K}_2, \mathcal{M}]$ *as defined in Figure 6 is also* $\mathsf{R^cT\text{-}ind\text{-}atk}$ *secure.*

More precisely, for any efficient adversary $\mathcal{A}$ *of type* $\mathsf{R^cT}$ *against the combined KEM* $\mathcal{K}' = \mathsf{XtM}[\mathcal{K}_1, \mathcal{K}_2, \mathcal{M}]$*,* *there exist efficient adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) \quad &\leq \quad 2\cdot\min\left\{\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_1}(\mathcal{B}_1), \mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_2}(\mathcal{B}_2)\right\} \\ &\quad + \ \mathsf{Adv}^{\mathsf{R^cT\text{-}OT\text{-}sEUF}}_{\mathcal{M}}(\mathcal{B}_3).\end{aligned}$$

Moreover, the run times of $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ are approximately the same as that of $\mathcal{A}$, and $\mathcal{B}_3$ makes at most as many verification queries as $\mathcal{A}$ makes decapsulation queries.

Proof. Assume there exists an adversary $\mathcal{A}$ that breaks the $\mathsf{R^cT\text{-}ind\text{-}atk}$ security of $\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]$. We show that this yields an adversary that then breaks either the $\mathsf{R^cT\text{-}ind\text{-}atk}$ security of $\mathcal{K}_1$ or of $\mathcal{K}_2$, or the $\mathsf{R^cT\text{-}OT\text{-}sEUF}$ security of $\mathcal{M}$. Because of symmetry it suffices to consider the case of $\mathcal{K}_1$. The following proof holds analogously for $\mathcal{K}_2$ being $\mathsf{R^cT\text{-}ind\text{-}atk}$ secure. We also focus on the $\mathsf{ind\text{-}cca}$ case here; the $\mathsf{ind\text{-}cpa}$ case follows easily from this.

We prove the theorem by applying the common technique of game hopping, bounding the adversary's advantage introduced with each game hop until the adversary cannot win beyond the guessing probability.

Game 0. This is the original $\mathsf{R^cT\text{-}ind\text{-}atk}$ game against $\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]$.

Game 1. We now replace the key $k_1^*\|k^*_{\mathsf{mac},1}$ of $\mathcal{K}_1$ returned with the challenge ciphertext part c_1^* with a uniformly random and independent value $r_1^*\|r^*_{\mathsf{mac},1}$ from the same keyspace K. This means that we first create $(c_1^*, k_1^*\|k^*_{\mathsf{mac},1})$ and then use $(c_1^*, r_1^*\|r^*_{\mathsf{mac},1})$ immediately from then on. This is done consistently in the challenge value for deriving the challenge key portion and the MAC, as well as in all decapsulation requests involving the ciphertext portion c_1^*. More precisely, we replace the step "$k_1\|k_{\mathsf{mac},1} \leftarrow\!\!\$\ \mathsf{Decaps}_1(sk_1, c_1)$" in the decapsulation procedure with the step "if $c_1 = c_1^*$ then $k_1\|k_{\mathsf{mac},1} \leftarrow r_1^*\|r^*_{\mathsf{mac},1}$ else $k_1\|k_{\mathsf{mac},1} \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$".
We show that if $\mathcal{A}$ can efficiently distinguish Game 1 from Game 0, then there exists an adversary $\mathcal{B}_1$ against the $\mathsf{R^cT\text{-}ind\text{-}atk}$ security of $\mathcal{K}_1$. Algorithm $\mathcal{B}_1$ receives as input a public key pk_1 and a challenge ciphertext c_1^* of $\mathcal{K}_1$, as well as the challenge key $k_1^*\|k^*_{\mathsf{mac},1}$. This challenge key is either the actual key or random. Algorithm $\mathcal{B}_1$ simulates the environment for $\mathcal{A}$ as follows. First, $\mathcal{B}_1$ generates the key pair (pk_2, sk_2) for $\mathcal{K}_2\|k_{\mathsf{mac},2}$ and sets $pk \leftarrow (pk_1, pk_2)$. Furthermore, $\mathcal{B}_1$ chooses the second challenge ciphertext portion c_2^* and key share $k_2^*\|k^*_{\mathsf{mac},2}$ itself. It computes $k^*_{\mathsf{kem}} \leftarrow k_1^* \oplus k_2^*$ and $k^*_{\mathsf{mac}} \leftarrow (k^*_{\mathsf{mac},1}, k^*_{\mathsf{mac},2})$, and assembles the challenge ciphertext (c_1^*, c_2^*, τ^*) where $\tau^* \leftarrow \mathsf{MAC}_{k^*_{\mathsf{mac}}}((c_1^*, c_2^*))$. It also picks a challenge bit and replaces k^*_{kem} by a random value if this bit is 1. Adversary $\mathcal{B}_1$ then runs $\mathcal{A}$ on input $(pk, (c_1^*, c_2^*, \tau^*), k^*_{\mathsf{kem}})$.

If $\mathcal{A}$ is an active adversary, mounting an $\mathsf{ind\text{-}cca}$ attack, decapsulation queries for ciphertexts $c = (c_1, c_2)$ with $c_1 \neq c_1^*$ and some MAC tag τ are answered as follows: c_1 is decapsulated using $\mathcal{B}_1$'s decapsulation oracle for $\mathcal{K}_1$, c_2 is decapsulated using sk_2, and the response k_{kem} is then computed as the appropriately truncated XOR of these decapsulations after verifying the MAC tag. If $c = (c_1^*, c_2)$ then $\mathcal{B}_1$ uses $k_1^*\|k^*_{\mathsf{mac},1}$ as the decapsulation of c_1^*, and then continues as in the previous case. For passive $\mathsf{ind\text{-}cpa}$ adversaries $\mathcal{A}$, algorithm $\mathcal{B}_1$ does not need to provide any simulation of decapsulation queries. At some point, the distinguisher $\mathcal{A}$ terminates and outputs a guess bit b'. Adversary $\mathcal{B}_1$ outputs the same bit b'.

Clearly, $\mathcal{B}_1$ perfectly simulates the environments for $\mathcal{A}$ corresponding to Game 0 if the challenge key $k_1^*\|k^*_{\mathsf{mac},1}$ is the actual key, and perfectly simulates Game 1 if $k_1^*\|k^*_{\mathsf{mac},1}$ is random. Furthermore, $\mathcal{B}_1$ is of the same type as $\mathcal{A}$. Hence we have

$$\mathsf{Adv}^{G_0}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) \leq \mathsf{Adv}^{G_1}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) + 2\cdot\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_1}(\mathcal{B}_1),$$

where the factor 2 is owned to the transition from the prediction-based $\mathsf{R^cT\text{-}ind\text{-}atk}$ attack with a random challenge bit b to an indistinguishability-based comparison between fixed games here.

Game 2. In a syntactical change we replace the now random value r_1^* by $r_1^* \oplus k_2^*$ where k_2^* is the encapsulated key in $\mathcal{K}_2$ in the challenge ciphertext. This is done consistently in the challenge value and MAC, as well as in all decapsulation requests involving the ciphertext portion c_1^*. We leave $r_{\mathsf{mac},1}^*$ unaltered.
Effectively, the modification means that the encapsulated key in the challenge ciphertext is now $r_1^* = (r_1^* \oplus k_2^*) \oplus k_2^*$. Since r_1^* and k_2^* are independent, the distributions of r_1^* and $r_1^* \oplus k_2^*$ are identical, so the adversary's advantage does not change: $\mathsf{Adv}^{G_1}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) = \mathsf{Adv}^{G_2}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A})$.

In Game 2 the adversary now receives a random value as the challenge key and the MAC is also computed over a random key part $r_{\mathsf{mac},1}^*$, independently of the challenge bit b. To complete the argument we only need to show that the adversary in an $\mathsf{ind\text{-}cca}$ attack does not gain any advantage via the decapsulation oracle (in which $r_1^* \oplus k_2^*$ from the challenge key is used for inputs of the form $(c_1^*, *, *)$). We next argue that the difference is negligible, though, because in the actual attack this can only happen if the adversary forges a MAC.

Game 3. In this game we change the decapsulation oracle in that we let it immediately reject with output $\perp$ if it is queried on a ciphertext of the form $(c_1^*, *, *)$ for the challenge ciphertext c_1^*.
An adversary is only able to notice the difference between Games 2 and 3 if it queries about a fresh ciphertext $(c_1^*, c_2, \tau) \neq (c_1^*, c_2^*, \tau^*)$ with c_2 being a $\mathcal{K}_2$ ciphertext of the adversary's choice, and τ being a valid MAC tag. If τ is not a valid MAC tag or the adversary queries exactly the challenge ciphertext, then our decapsulation oracle would also return $\perp$.

We show that if an adversary $\mathcal{A}$ distinguishes between the games, we can build an adversary $\mathcal{B}_3$ against the MAC. Adversary $\mathcal{B}_3$ runs $\mathcal{A}$ according to Game 2, choosing all components (sk_1, pk_1) and (sk_2, pk_2) and c_1^*, c_2^* of $\mathcal{K}_1$ and $\mathcal{K}_2$ itself. To create the challenge ciphertext, adversary $\mathcal{B}_3$ makes its one-time MAC request with message (c_1^*, c_2^*) and receives τ^*. It runs $\mathcal{A}$ on (c_1^*, c_2^*, τ^*) and a random string k_{kem}^*. For an $\mathsf{ind\text{-}cca}$ attack, if $\mathcal{A}$ makes a decapsulation query about (c_1, c_2, τ) for $c_1 \neq c_1^*$, then $\mathcal{B}_3$ uses knowledge of its decapsulation keys to compute the answer. For $c_1 = c_1^*$ adversary $\mathcal{B}_3$ calls its verification oracle with $(c_1, c_2, \tau, 2, k_2)$, where $k_2 \| k_{\mathsf{mac},2} \leftarrow \mathsf{Decaps}_1(sk_2, c_2)$, and returns $\perp$ to $\mathcal{A}$ to continue the simulation.

For the analysis note that Game 2 uses as the challenge key either an independent random string r_1^* (if $b = 0$) or a random key (if $b = 1$). In both cases, the KEM part of the key is a uniform key independent of bit b—as is $\mathcal{B}_3$'s choice k_{kem}^*— and the MAC key part is also independent and uniform. The latter holds in $\mathcal{B}_3$'s simulation as well, since the $\mathsf{OT\text{-}sEUF}$ game chooses a random MAC key. In other words, the simulation is perfect up to the step where, potentially, $\mathcal{A}$ makes a query for a fresh ciphertext with a valid MAC which would yield a reply different from $\perp$. But then $\mathcal{B}_3$ would find a forgery against the MAC in one of its multiple verification attempts.

Since $\mathcal{B}_3$ is of the same type $\mathsf{R^cT}$ as $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{G_2}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) \leq \mathsf{Adv}^{G_3}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{R^cT\text{-}OT\text{-}sEUF}}_{\mathcal{M}}(\mathcal{B}_3).$$

The claim now follows, noting that in the final game the secret bit b is perfectly hidden from $\mathcal{A}$. The challenge key is an independent string in either case $b = 0$ or $b = 1$, and the decapsulation queries are now also independent of the bit b, since the change in the oracle's answers only depends on the public value c_1^*. Hence, $\mathcal{A}$'s output is independent of the secret bit, and thus $\mathsf{Adv}^{G_3}_{\mathsf{XtM}[\mathcal{K}_1,\mathcal{K}_2,\mathcal{M}]}(\mathcal{A}) \leq 0$. □

Instantiating the MAC. We use a strong form of combiner for MACs where the adversary can choose one of the two MAC keys. It is easy to build secure MAC combiners of this type by concatenating two MACs, each computed under one of the keys. For specific constructions carious improvements may apply. For instance, for deterministic MACs in which verification is performed via re-computation, one may aggregate the two MACs via exclusive-or [33] to reduce the communication overhead.

MACs satisfying the $\mathsf{Q^cQ\text{-}OT\text{-}sEUF}$ notion can be constructed based on the Carter-Wegman paradigm using universal hash functions [45], without relying on cryptographic assumptions. Our construction of course uses that the input, consisting of the ciphertexts holding the keys, is larger then the keys, such that we need to extend the domain of the universal hash function. For a pairwise-independent hash function with bound ϵ, it is clear that an adversary cannot win with a single verification query after seeing one MAC, except with probability at most ϵ. Since verification is deterministic and consists of re-computing the tag, it follows that the adversary cannot win with probability more than $q\epsilon$ with q verification queries [7].

The Carter-Wegman paradigm allows for another potential improvement. Suppose one uses hashing of the form $am + b$ over some finite field $\mathbf{F}$ with addition $+$ and multiplication $\cdot$, where m is the message and $k_{\mathsf{mac}} = (a, b)$ is the MAC key. Then, instead of computing one MAC for each key part $k_{\mathsf{mac},1}$ and $k_{\mathsf{mac},2}$, one can compute a single MAC over the key $k_{\mathsf{mac}} = k_{\mathsf{mac},1} + k_{\mathsf{mac},2} = (a_1 + a_2, b_1 + b_2)$. This combiner provides strong unforgeability as required above, since for known keys $k_{\mathsf{mac},2} = (a_2, b_2)$ and $k'_{\mathsf{mac},2} = (a'_2, b'_2)$ one can transform a MAC for message m under unknown key $k_{\mathsf{mac},1} + k_{\mathsf{mac},2}$ into one for $k_{\mathsf{mac},1} + k'_{\mathsf{mac},2}$, simply by adding $(a'_2 - a_2) \cdot m + (b'_2 - b_2)$ to the tag. By symmetry this holds analogously for known keys $k_{\mathsf{mac},1}$ and $k'_{\mathsf{mac},1}$.

Alternatively to Carter-Wegman MACs, one could use HMAC for instantiating the MAC directly, or rely on the HKDF paradigm of using HMAC as an extractor. Namely, one applies the extraction step of HKDF, $\mathsf{HKDF.Ext}$, with the ciphertexts acting as the salt and the MAC key as the keying material. This approach is based on the idea that HMAC is a good extractor. We discuss such issues in more detail next, when looking at the TLS-like combiner.

3.1.4 Resistance against full quantum attacks.

We have shown that the combiner $\mathsf{XtM}[\mathcal{K}_1, \mathcal{K}_2, \mathcal{M}]$ inherits security of the underlying KEMs if the MAC is secure, for classical queries to the decapsulation oracle (which is the setting we also consider for key exchange). We outline here that the result can be easily extended to fully quantum adversaries with superposition queries to the decapsulation oracle. This only assumes that one of the individual KEMs achieves this level of security. Interestingly, the MAC $\mathcal{M}$ only needs to be $\mathsf{Q^cQ\text{-}OT\text{-}sEUF}$ secure for a single classical verification query. The reason is that the MAC in the challenge is still computed classically, and in the security reduction we will measure a potential forgery in a decapsulation superposition query and output this classical MAC.

The approach for showing security is very similar to the proof in the post-quantum case. The only difference lies in the final game hop, where we cannot simply read off a potential MAC forgery from a decapsulation query of the form $(c_1^*, *, *)$ for the value c_1^* in the challenge, because the query is in superposition. But we can adapt the "measure-and-modify" technique of Boneh et al. [13] for proving the quantum-resistance of Bellare-Rogaway style encryptions. In our case, if the amplitudes of entries (c_1^*, c_2, τ) with a valid MAC and fresh $(c_2, \tau) \neq (c_2^*, \tau^*)$ in the quantum decapsulation queries would be non-negligible, then we could measure for a randomly chosen query among the polynomial many decapsulation queries to get a (classical) MAC forgery with non-negligible probability. This would contradict the $\mathsf{Q^cQ\text{-}OT\text{-}sEUF}$ security of $\mathcal{M}$. If, on the other hand, the query probability of such forgeries is negligible, then we can change the function $\mathsf{Decaps}^{\perp}$ into $\mathsf{Decaps}^{\perp\perp}$ which now also outputs $\perp$ for any query of the form $(c_1^*, *, *)$. Following the line of reasoning as in [13], based on the results of Bennett et al. [11], this cannot change the adversary's output behavior significantly. Again, since we can instantiate the MAC for classical queries information-theoretically, we get a secure KEM combiner in the fully quantum case, without requiring an extra assumption beyond full quantum resistance of one of the KEMs.

3.2 $\mathsf{dualPRF}$: Dual-PRF Combiner

Our second combiner is based on dual PRFs [5, 4, 8]. The definitions of (dual) PRF security can be found in the supplementary material in Section B. Informally, a dual PRF $\mathsf{dPRF}(k, x)$ is a PRF when either the

$\mathsf{Encaps}_{\mathsf{dualPRF}}(pk_1, pk_2)$:

$(c_1, k_1) \leftarrow \mathsf{Encaps}_1(pk_1)$
$(c_2, k_2) \leftarrow \mathsf{Encaps}_2(pk_2)$
$c \leftarrow (c_1, c_2)$
$k_d \leftarrow \mathsf{dPRF}(k_1, k_2)$
$k \leftarrow \mathsf{PRF}(k_d, c)$
return (c, k)

$\mathsf{Decaps}_{\mathsf{dualPRF}}(sk_1, sk_2, c_1, c_2)$:

$k_1' \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$
$k_2' \leftarrow \mathsf{Decaps}_2(sk_2, c_2)$
$k_d' \leftarrow \mathsf{dPRF}(k_1', k_2')$
return $\mathsf{PRF}(k_d', (c_1, c_2))$

Figure 8: KEM constructed by the dual PRF combiner $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$.

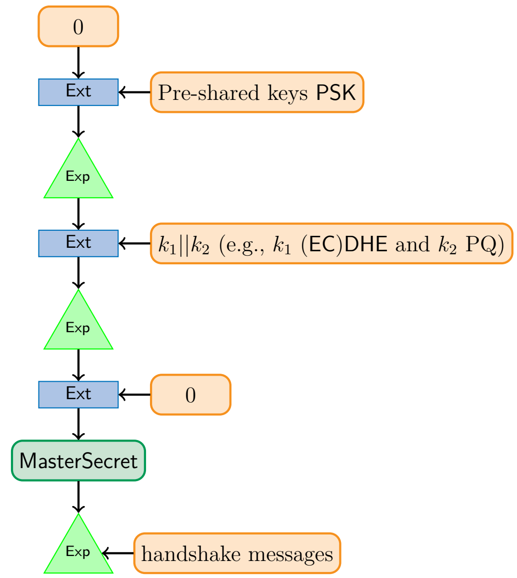

Figure 9: Excerpt from altered TLS 1.3 key schedule as proposed in [46] to incorporate an additional secret k_2, effectively enabling a hybrid mode.

key material k is random (i.e., $\mathsf{dPRF}(k, \cdot)$ is a PRF), or alternatively when the input x is random (i.e., $\mathsf{dPRF}(\cdot, x)$ is a PRF). HMAC has been shown to be a secure MAC under the assumption it is a dual PRF, and Bellare and Lysyanskaya have given a generic validation of the dual PRF assumption for HMAC [8] and therefore HKDF.

To construct a hybrid KEM from a dual PRF, the naive approach of directly using a dual PRF to compute the session key of the combined KEM as $\mathsf{dPRF}(k_1, k_2)$ is not sufficient. If, say, $\mathcal{K}_1$ is secure and $\mathcal{K}_2$ is completely broken, then an adversary might be able to transform the challenge ciphertext (c_1^*, c_2^*) into (c_1^*, c_2), where $c_2 \neq c_2^*$ but encapsulates the same key k_2 as c_2^*. With a single decapsulation query the adversary would be able to recover the key $\mathsf{dPRF}(k_1, k_2)$ and distinguish it from random. Our approach, shown in Figure 8, is to apply another pseudorandom function with the output of the dual PRF as the PRF key and the ciphertexts as the input label: $\mathsf{PRF}(\mathsf{dPRF}(k_1, k_2), (c_1, c_2))$.

Our $\mathsf{dualPRF}$ combiner is inspired by the key derivation in TLS 1.3 [40] and models Whyte et al.'s proposal for supporting hybrid key exchange in TLS 1.3 [46]. In TLS 1.3, HKDF's extract function is applied to the raw ECDH shared secret; the result is then fed through HKDF's expand function with the (hash of the) transcript as (part of) the label. In Whyte et al.'s hybrid proposal, the session keys from multiple KEMs are concatenated as a single shared secret input to HKDF extract. The $\mathsf{dualPRF}$ combiner models this by taking dPRF as HKDF extract and PRF as HKDF expand.

3.2.1 Security of the dual-PRF combiner.

We can now show that the dual-PRF combiner is a robust KEM combiner, in the sense that the resulting KEM has the security of the strongest of the two input KEMs (assuming the PRF and dual PRF are also sufficiently secure). In particular, we show that $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$ is IND-CCA secure in the post-quantum setting ($\mathsf{Q^cQ}$) if dPRF is a post-quantum secure dual PRF, PRF is a post-quantum secure PRF, and at least one of the two KEMs is post-quantum IND-CCA secure.

Theorem 2 (Dual-PRF is robust)**.** *Let $\mathcal{K}_1$ be an $\mathsf{X^cZ\text{-}ind\text{-}atk}$ secure KEM, $\mathcal{K}_2$ be a $\mathsf{U^cW\text{-}ind\text{-}atk}$ secure KEM, and $\mathsf{R^cT} = \max\{\mathsf{X^cZ}, \mathsf{U^cW}\}$. Moreover, let $\mathsf{dPRF} : K_1 \times K_2 \rightarrow K'$ be a $\mathsf{R^cT}$ secure dual PRF, and*

$\mathsf{PRF} : K' \times \{0,1\}^* \to K_{\mathsf{dualPRF}}$ *be an* $\mathsf{R^cT}$ *secure PRF. Then* $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$ *as defined in Figure 8 is* $\mathsf{R^cT}$-ind-atk *secure.*

More precisely, for any ind-atk *adversary* $\mathcal{A}$ *of type* $\mathsf{R^cT}$ *against the combiner* $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$, *we derive efficient adversaries* $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, *and* $\mathcal{B}_4$ *such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) \quad \leq \quad & \min\left\{\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_1}(\mathcal{B}_1), \mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_2}(\mathcal{B}_2)\right\} \\ & +2 \cdot \mathsf{Adv}^{\mathsf{R^cT\text{-}dprf\text{-}sec}}_{\mathsf{dPRF}}(\mathcal{B}_3) + 2 \cdot \mathsf{Adv}^{\mathsf{R^cT\text{-}prf\text{-}sec}}_{\mathsf{PRF}}(\mathcal{B}_4).\end{aligned}$$

The theorem relies on two-stage security notions for PRFs and dual PRFs, which are the natural adaptation of PRF and dual-PRF security: a two-stage $\mathsf{X^yZ}$ adversary for PRF is classical or quantum (X) while it has access to the PRF oracle (which it accesses classically or in superposition depending on y). After this, in the second stage, it runs classically or quantumly (Z) without oracle access, before outputting a guess as to whether its oracle was real or random. We give the formal definitions of two-stage security notions for PRFs and dual PRFs in the supplementary material in Section B.

Proof. As before, we prove the theorem by considering a sequence of game hops. We focus on the case that $\mathcal{K}_1$ is secure, and mention the necessary modification in the proof for $\mathcal{K}_2$ being secure as we progress through the games.

Game 0. The original $\mathsf{R^cT}$-ind-atk game for $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$.

Game 1. We replace the value k_1^* computed in the challenge ciphertext by a uniformly random value r_1^* of equal length and compute the final key in the challenge value as $\mathsf{PRF}(\mathsf{dPRF}(r_1^*, k_2^*), (c_1^*, c_2^*))$; note that for $b = 1$ this value is eventually replaced with a random value. Decapsulation requests (c_1, c_2) are also answered by using r_1^* instead of k_1^* in case $c_1^* = c_1$. That is, instead of computing $k_1 \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$ we compute "if $c_1 = c_1^*$ then $k_1 \leftarrow r_1^*$ else $k_1 \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$".
An adversary distinguishing Game 0 from Game 1 would immediately yield an efficient adversary $\mathcal{B}_1$ against the indistinguishability of $\mathcal{K}_1$. Adversary $\mathcal{B}_1$ receives as input pk_1 and a challenge (c_1^*, k_1^*), and simulates the environment for $\mathcal{A}$ as follows: First, $\mathcal{B}_1$ generates the key pair (pk_2, sk_2) for $\mathcal{K}_2$ and sets $pk \leftarrow (pk_1, pk_2)$. Furthermore, $\mathcal{B}_1$ generates the second challenge ciphertext portion c_2^* and key share k_2^* itself. It assembles the challenge ciphertext as (c_1^*, c_2^*) and computes $k^* = \mathsf{PRF}(\mathsf{dPRF}(k_1^*, k_2^*), (c_1^*, c_2^*))$. It also picks a challenge bit and replaces k^* by a random value if this bit is 1. Adversary $\mathcal{B}_1$ then runs $\mathcal{A}$ on input $(pk, (c_1^*, c_2^*), k^*)$.

If $\mathcal{A}$ is an active adversary in the ind-cca case, decapsulation queries for ciphertexts $c = (c_1, c_2)$ with $c_1 \neq c_1^*$ are answered by relaying c_1 to the corresponding decapsulation oracle for $\mathcal{K}_1$ and decapsulating c_2 with the help of sk_2. The final response is computed according to the protocol description. If $c = (c_1^*, c_2)$ then $\mathcal{B}_1$ simply substitutes the evaluation and response of $\mathcal{K}_1$'s decapsulation oracle with k_1^* and computes the answer accordingly. For passive adversaries $\mathcal{A}$ in an ind-cpa attack, algorithm $\mathcal{B}_1$ does not need to provide any simulation of decapsulation queries. At some point, the distinguisher $\mathcal{A}$ terminates and outputs a guess bit b'. Adversary $\mathcal{B}_1$ outputs the same bit b'.

For the analysis note that the difference between the two games lies exactly in the distinction between the actual key k_1^* and a random value. Hence, we have

$$\mathsf{Adv}^{G_0}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) \leq \mathsf{Adv}^{G_1}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]} + 2 \cdot \mathsf{Adv}^{\mathsf{X^cZ\text{-}ind\text{-}atk}}_{\mathcal{K}_1}(\mathcal{B}_1).$$

For $\mathcal{K}_2$ being secure the proof applies analogously, yielding an adversary $\mathcal{B}_2$.

Game 2. Next, we replace the value $\mathsf{dPRF}(r_1^*, k_2^*)$ by a uniformly random value r^* in the computation of the challenge ciphertext. We make the following additional modification to the current decapsulation procedure: We use r^* in decapsulation requests (instead of $\mathsf{dPRF}(r_1^*, k_2^*)$) for any request of the form (c_1^*, c_2)

for which $\mathsf{Decaps}_2(sk_2, c_2) = k_2^*$. (Note that we still use r_1^* and $\mathsf{dPRF}(r_1^*, k_2)$ for queries of the form $c_1 = c_1^*$ but $k_2 = \mathsf{Decaps}_2(sk_2, c_2) \neq k_2^*$.)
Distinguishing Game 1 from Game 2 would immediately yield an efficient adversary $\mathcal{B}_3$ against the PRF-security of dPRF. The reduction is straightforward. Algorithm $\mathcal{B}_3$, in its first query, can ask about some input value and either receives the PRF value or a random reply, and from then on $\mathcal{B}_3$ can ask about other inputs to learn further PRF values.

Algorithm $\mathcal{B}_3$ creates keys (pk_1, sk_1), (pk_2, sk_2), as well as c_1^*, c_2^* (with encapsulated keys k_1^*, k_2^*). It then queries its PRF oracle about k_2^* to receive a value r^*. It starts a simulation of $\mathcal{A}$ on input $(pk, (c_1^*, c_2^*), \mathsf{PRF}(r^*, (c_1^*, c_2^*)))$. Each decapsulation query for (c_1, c_2) with $c_1 \neq c_1^*$ is answered with the help of sk_1, sk_2. Each query with $c_1 = c_1^*$ is answered by decapuslating k_2 from c_2 with sk_2. If now $k_2 = k_2^*$ then we use r^* to compute the final answer, else we query the pseudorandom function oracle about k_2 and use the reply to complete the computation. Note that this is admissible since k_2 is different from the input k_2^* in $\mathcal{B}_3$'s first query. Output the final answer of adversary $\mathcal{A}$.

If $\mathcal{B}_3$ receives a pseudorandom value r^* then we perfectly simulate Game 1. If, on the other hand, r^* is random, then we perfectly simulate Game 2. Thus we have:

$$\mathsf{Adv}^{G_1}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) \leq \mathsf{Adv}^{G_2}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) + 2 \cdot \mathsf{Adv}^{\mathsf{R^cT\text{-}dprf\text{-}sec}}_{\mathsf{dPRF}}(\mathcal{B}_3).$$

Here the factor 2 in the distinguishing advantage against dPRF comes from the fact that we use the variant of having a real-or-random challenge and then communicating with the actual function on different inputs.

For $\mathcal{K}_2$ being secure the same line of reasoning applies, because dPRF is a dual PRF, such that $\mathsf{dPRF}(\cdot, r_2^*)$ is also pseudorandom.

Game 3. Finally, we replace the value $\mathsf{PRF}(r^*, (c_1^*, c_2^*))$ by a uniformly random value R^* in the computation of the challenge ciphertext. The decapsulation procedure remains unchanged.
We show security by a reduction to the security of the PRF. Algorithm $\mathcal{B}_4$ again creates keys (pk_1, sk_1), (pk_2, sk_2) and the challenge ciphertext (c_1^*, c_2^*). It queries the PRF oracle about (c_1^*, c_2^*) to obtain a value R^* and runs $\mathcal{A}$ on $(pk, (c_1^*, c_2^*), R^*)$. Each decapsulation query (c_1, c_2) is answered as follows: If $c_1 \neq c_1^*$ we answer with the help of the decapsulation keys sk_1, sk_2, computing the same reply as the original decapsulation oracle. In case $c_1 = c_1^*$ (and consequently $c_2 \neq c_2^*$) and where $k_2 = k_2^*$ we call the external oracle about (c_1^*, c_2) to derive the answer. For $c_1 = c_1^*$ but $k_2 \neq k_2^*$ we use $\mathsf{dPRF}(r_1^*, k_2)$ to compute the answer.

We again have that if $\mathcal{B}_4$ receives a pseudorandom value R^* then we perfectly simulate Game 2. If R^* is random, then we perfectly simulate Game 3. Hence,

$$\mathsf{Adv}^{G_2}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) \leq \mathsf{Adv}^{G_3}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) + 2 \cdot \mathsf{Adv}^{\mathsf{R^cT\text{-}prf\text{-}sec}}_{\mathsf{PRF}}(\mathcal{B}_4).$$

The analogous applies when $\mathcal{K}_2$ is secure.

In the final game neither the challenge ciphertext nor the decapsulation oracle carries any information about the secret challenge bit b. That is, the oracle does not depend on R^* in case $b = 0$, so this case is indistinguishable from the $b = 1$ case. We thus arrive at the final bound: $\mathsf{Adv}^{G_3}_{\mathsf{dualPRF}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF}]}(\mathcal{A}) = 0$.

This concludes the proof that $\mathsf{dualPRF}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}]$ is a hybrid KEM with $\mathsf{R^cT\text{-}ind\text{-}atk}$ security. □

3.3 N: Nested Dual-PRF Combiner

We augment the dualPRF combiner in the previous section by an extra preprocessing step for the key k_1: $k_e \leftarrow \mathsf{Ext}(0, k_1)$, where Ext is another PRF. This is the nested dual-PRF combiner N shown in Figure 10.

Our nested dual-PRF combiner N models Schanck and Stebila's proposal for hybrid key exchange in TLS 1.3 [42]. In their proposal, as depicted in Figure 11, one stage of the TLS 1.3 key schedule is applied

$\mathsf{Encaps}_{\mathsf{N}}(pk_1, pk_2)$:

$(c_1, k_1) \leftarrow \mathsf{Encaps}_1(pk_1)$
$(c_2, k_2) \leftarrow \mathsf{Encaps}_2(pk_2)$
$c = (c_1, c_2)$
$k_e = \mathsf{Ext}(0, k_1)$
$k_d = \mathsf{dPRF}(k_e, k_2)$
$k = \mathsf{PRF}(k_d, c)$
return (c, k)

$\mathsf{Decaps}_{\mathsf{N}}(sk_1, sk_2, c_1, c_2)$:

$k_1' \leftarrow \mathsf{Decaps}_1(sk_1, c_1)$
$k_2' \leftarrow \mathsf{Decaps}_2(sk_2, c_2)$
$k_e' = \mathsf{Ext}(0, k_1)$
$k_d' = \mathsf{dPRF}(k_e', k_2)$
return $\mathsf{PRF}(k_d', (c_1, c_2))$

Figure 10: KEM constructed by the nested dual-PRF combiner $\mathsf{N}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF},\mathsf{Ext}\ ]$.

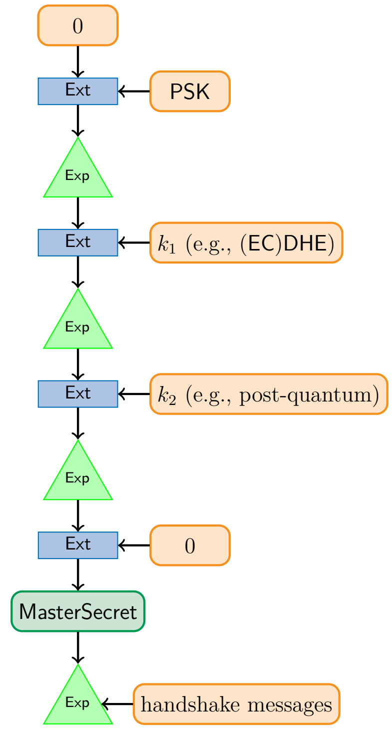

Figure 11: Excerpt from altered TLS 1.3 key schedule as proposed in [42] to incorporate an additional secret k_2, effectively enabling a hybrid mode.

for each of the constituent KEMs in the hybrid KEM construction: each stage in the key schedule applies the HKDF extract function with one input being the output from the previous stage of the key schedule and the other input being the shared secret from this stage's KEM. Finally, HKDF expand incorporates the (hash of the) transcript, including all KEMs' ciphertexts. Modeling the extraction function Ext as a PRF, our nested combiner N captures this scenario.

3.3.1 Security of the nested dual-PRF combiner.

We can now show that the nested dual-PRF combiner N is a robust KEM combiner, in the sense that the resulting KEM has the security of the strongest of the two input KEMs (assuming the PRFs are sufficiently secure). Informally, the theorem shows that $\mathsf{N}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}, \mathsf{Ext}]$ is IND-CCA secure in the post-quantum setting if dPRF is a post-quantum secure dual PRF, PRF and Ext are post-quantum secure PRFs, and at least one of the two KEMs is post-quantum IND-CCA secure.

Theorem 3 (Nested dual-PRF is robust)**.** *Let $\mathcal{K}_1$ be an $\mathsf{X^cZ}$-ind-atk secure KEM, $\mathcal{K}_2$ be a $\mathsf{U^cW}$-ind-atk secure KEM, $\mathsf{dPRF} : K' \times K_2 \to K''$ be a $\mathsf{R^cT} = \max\{\mathsf{X^cZ}, \mathsf{U^cW}\}$ secure dual PRF, $\mathsf{PRF} : K'' \times \{0,1\}^* \to K_{\mathsf{N}}$ be an $\mathsf{R^cT}$ secure PRF, and $\mathsf{Ext} : \{0,1\}^* \times K_1 \to K'$ be an $\mathsf{R^cT}$ secure PRF. Then the combiner $\mathsf{N}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}, \mathsf{Ext}]$ as defined in Figure 8 is $\mathsf{R^cT}$-ind-atk secure.*

More precisely, for any ind-atk adversary $\mathcal{A}$ of type $\mathsf{R^cT}$ against the combined KEM $\mathsf{N}[\mathcal{K}_1, \mathcal{K}_2, \mathsf{dPRF}, \mathsf{PRF}, \mathsf{Ext}]$, *we derive efficient adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$, and $\mathcal{B}_5$ such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathsf{N}[\mathcal{K}_1,\mathcal{K}_2,\mathsf{dPRF},\mathsf{PRF},\mathsf{Ext}]}(\mathcal{A}) \quad\leq\quad & \min\left\{\mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_1}(\mathcal{B}_1), \mathsf{Adv}^{\mathsf{R^cT\text{-}ind\text{-}atk}}_{\mathcal{K}_2}(\mathcal{B}_2)\right\} \\ & +2\cdot\mathsf{Adv}^{\mathsf{R^cT\text{-}dprf\text{-}sec}}_{\mathsf{dPRF}}(\mathcal{B}_3) + 2\cdot\mathsf{Adv}^{\mathsf{R^cT\text{-}prf\text{-}sec}}_{\mathsf{PRF}}(\mathcal{B}_4) \\ & +2\cdot\mathsf{Adv}^{\mathsf{R^cT\text{-}prf\text{-}sec}}_{\mathsf{Ext}}(\mathcal{B}_5).\end{aligned}$$

The proof follows easily from the proof of the $\mathsf{dualPRF}$ combiner. Only here we make one more intermediate step in which we use the pseudorandomness of Ext to argue that the output of $\mathsf{Ext}(0, k_1)$ is pseudorandom.

4 Authenticated Key Exchange from Hybrid KEMs

We now turn towards the question of how to achieve hybrid authenticated key exchange from hybrid KEMs. There exists a vast body of literature on compilers for authenticated key exchange [6, 34, 18, 32, 37, 41]. In the following we consider secure AKE protocols from key encapsulation mechanisms combined with SigMA-style authentication [35]. As for KEMs, we consider a two-stage adversary and adjust the commonly used model for authenticated key exchange by Bellare and Rogaway [9] to this setting.

4.1 Security Model

We begin by establishing the security definition for authenticated key exchange against active attackers, starting from the model of Bellare and Rogaway [9].

4.1.1 Parties and sessions.

Let KE be a key exchange protocol. We denote the set of all participants in the protocol by $\mathcal{U}$. Each participant $U \in \mathcal{U}$ is associated with a long-term key pair (pk_U, sk_U), created in advance; we assume every participant receives an authentic copy of every other party's public key through some trusted out-of-band mechanism. In a single run of the protocol (referred to as a *session*), U may act as either initiator or responder. Any participant U may execute multiple sessions in parallel or sequentially.

We denote by $\pi^j_{U,V}$ the jth session of user $U \in \mathcal{U}$ (called the session *owner*) with intended communication partner V. Associated to each session are the following per-session variables; we often write $\pi^j_{U,V}.\mathsf{var}$ to refer to the variable var of session $\pi^j_{U,V}$.

- $\mathsf{role} \in \{\mathsf{initiator}, \mathsf{responder}\}$ is the role of the session owner in this session.
- $\mathsf{st_{exec}} \in \{\mathsf{running}, \mathsf{accepted}, \mathsf{rejected}\}$ reflects the current status of execution. The initial value at session creation is $\mathsf{running}$.
- $\mathsf{sid} \in \{0,1\}^* \cup \{\bot\}$ denotes the session identifier. The initial value is $\bot$.
- $\mathsf{st_{key}} \in \{\mathsf{fresh}, \mathsf{revealed}\}$ indicates the status of the session key K. The initial value is fresh.
- $\mathsf{K} \in \mathcal{D} \cup \{\bot\}$ denotes the established session key. The initial value is $\bot$.
- $\mathsf{tested} \in \{\mathsf{true}, \mathsf{false}\}$ marks whether the session key K has been tested or not. The initial value is false.

To identify related sessions which might compute the same session key, we rely on the notion of partnering using session identifiers. Two sessions $\pi^i_{S,T}$ and $\pi^j_{U,V}$ are said to be *partnered* if $\pi^i_{S,T}.\mathsf{sid} = \pi^j_{U,V}.\mathsf{sid} \neq \bot$. We assume that if the adversary has not interfered, sessions in a protocol run between two honest participants are partnered.

4.1.2 Adversary model.

The adversary interacts with honest parties running AKE protocol instances via oracle queries, which ultimately allow the adversary to fully control all network communications (injecting messages and scheduling if and when message delivery occurs) and compromise certain secret values; the goal of the adversary is to distinguish the session key of an uncompromised session of its choice from random.

We model the adversary as a two-stage potentially quantum adversary with varying levels of quantum capabilities. As in Section 3, we only consider adversaries that interact with parties using classical oracle queries, omitting $\mathsf{Q^qQ}$ adversaries.

The following queries model the adversary's control over normal operations by honest parties:

$\mathsf{NewSession}(U, V, role)$**:** Creates a new session $\pi^j_{U,V}$ for U (with j being the next unused counter value for sessions between U and intended communication partner $V \in \mathcal{U} \cup \{\star\}$) and sets $\pi^j_{U,V}.\mathsf{role} \leftarrow role$.

$\mathsf{Send}(\pi^j_{U,V}, m)$**:** Sends the message m to the session $\pi^j_{U,V}$. If no session $\pi^j_{U,V}$ exists or does not have $\pi^j_{U,V}.\mathsf{st_{exec}} = \mathsf{running}$, return $\bot$. Otherwise, the party U executes the next step of the key agreement protocol based on its local state, updates the execution status $\pi^j_{U,V}.\mathsf{st_{exec}}$, and returns any outgoing messages. If $\mathsf{st_{exec}}$ changes to $\mathsf{accepted}$ and the intended partner V has previously been corrupted, we mark the session key as revealed: $\pi^j_{U,V}.\mathsf{st_{key}} \leftarrow \mathsf{revealed}$.

The next queries model the adversary's ability to compromise secret values:

$\mathsf{Reveal}(\pi^j_{U,V})$**:** If $\pi^j_{U,V}.\mathsf{st_{exec}} = \mathsf{accepted}$, $\mathsf{Reveal}(\pi^j_{U,V})$ returns the session key $\pi^j_{U,V}.\mathsf{K}$ and marks the session key as revealed: $\pi^j_{U,V}.\mathsf{st_{key}} \leftarrow \mathsf{revealed}$. Otherwise, it returns $\bot$.

$\mathsf{Corrupt}(U)$**:** Returns the long-term secret key sk_U of U. Set $\pi^j_{V,W}.\mathsf{st_{key}} \leftarrow \mathsf{revealed}$ in all sessions where $V = U$ or $W = U$. (If the security definition is meant to capture forward secrecy, this last operation is omitted.)

The final query is used to define the indistinguishability property of session keys:

$\mathsf{Test}(\pi^j_{U,V})$**:** At the start of the experiment, a test bit b_{test} is chosen uniformly and random and fixed through the experiment. If $\pi^j_{U,V}.\mathsf{st_{exec}} \neq \mathsf{accepted}$, the query returns $\bot$. Otherwise, sets $\pi^j_{U,V}.\mathsf{tested} \leftarrow \mathsf{true}$ and proceeds as follows. If $b_{\mathsf{test}} = 0$, a key $\mathsf{K}^* \leftarrow\!\!\$\ \mathcal{D}$ is sampled uniformly at random from the session key distribution $\mathcal{D}$. If $b_{\mathsf{test}} = 1$, K^* is set to the real session key $\pi^j_{U,V}.\mathsf{K}$. Return K^*. The Test query may be asked only once.

4.2 Security Definitions

We now provide the specific security experiment and definition for AKE security. We follow the approach of Brzuska et al. [22, 21] and divide Bellare–Rogaway-style AKE security into the sub-notions of BR-Match security and BR key secrecy.

Definition 1 (Two-Stage BR-Match Security)**.** Let λ be the security parameter. Furthermore let KE be an authenticated key exchange protocol and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two-stage $\mathsf{X^cZ}$ QPT adversary interacting with KE via the queries defined in Section 4.1.2 in the following game $G^{\mathsf{BR\text{-}Match}}_{\mathsf{KE}}(\mathcal{A})$:

Setup. The challenger generates long-term public/private-key pairs with certificates for each participant $U \in \mathcal{U}$.

Query Phase 1. The adversary $\mathcal{A}_1$ receives the generated public keys and has access to the queries NewSession, Send, Reveal, Corrupt, and Test.

Stage Change. The adversary $\mathcal{A}_1$ passes some state st to the second stage adversary $\mathcal{A}_1$ and terminates.

Query Phase 2. $\mathcal{A}_2$ may now perform local computations on state st and only has access to queries Reveal and Corrupt.

Stop. At some point, the adversary $\mathcal{A}_2$ stops with no output.

We say that $\mathcal{A}$ wins the game, denoted by $G_{\mathsf{KE}}^{\mathsf{BR\text{-}Match}}(\mathcal{A}) = 1$, if at least one of the following conditions holds:

1. There exist two distinct sessions π and π' with $\pi.\mathsf{sid} = \pi'.\mathsf{sid} \neq \bot$, and $\pi.\mathsf{st}_{\mathsf{exec}}, \pi'.\mathsf{st}_{\mathsf{exec}} \neq \mathsf{rejected}$, but $\pi.\mathsf{K} \neq \pi'.\mathsf{K}$. (Different session keys in partnered sessions.)
2. There exist two sessions $\pi := \pi_{U,V}^{k}$ and $\pi' := \pi_{V',U'}^{k'}$ such that $\pi.\mathsf{sid} = \pi'.\mathsf{sid} \neq \bot$, $\pi.\mathsf{role} = \mathsf{initiator}$, and $\pi'.\mathsf{role} = \mathsf{responder}$, but $U \neq U'$ or $V \neq V'$. (Different intended partners.)
3. There exist at least three sessions π, π', and π'' such that π, π', π'' are pairwise distinct, but $\pi.\mathsf{sid} = \pi'.\mathsf{sid}' = \pi''.\mathsf{sid} \neq \bot$. (More than two sessions share the same session identifier.)

We say KE is *two-stage* $\mathsf{BR\text{-}Match}$ *secure* if for all QPT $\mathsf{X^cZ}$ adversaries $\mathcal{A}$ the advantage function

$$\mathsf{Adv}_{\mathsf{KE}}^{\mathsf{BR\text{-}Match}}(\mathcal{A}) = \Pr\left[G_{\mathsf{KE}}^{\mathsf{BR\text{-}Match}}(\mathcal{A}) = 1\right]$$

is negligible in the security parameter λ.

Definition 2 (Two-Stage BR Key Secrecy)**.** Let KE be a key exchange protocol with key distribution $\mathcal{D}$ and let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two-stage $\mathsf{X^cZ}$ adversary interacting with KE via the queries defined in Section 4.1 within the following security experiment $\mathrm{Expt}_{\mathsf{KE},\mathcal{D}}^{\mathsf{X^cZ\text{-}BR}}(\mathcal{A})$:

Setup. The challenger generates long-term public/private-key pairs for each participant $U \in \mathcal{U}$, chooses the test bit $b_{\mathsf{test}} \leftarrow_{\$} \{0,1\}$ at random, and sets $\mathsf{lost} \leftarrow \mathsf{false}$.

Query Phase 1. Adversary $\mathcal{A}_1$ receives the generated public keys and may (classically) query $\mathsf{NewSession}$, Send, Reveal, $\mathsf{Corrupt}$, and Test.

Stage Change. At some point, $\mathcal{A}_1$ terminates and outputs some state st to be passed to the second stage adversary $\mathcal{A}_2$.

Query Phase 2. $\mathcal{A}_2$ may now perform local computations on state st, but may query only Reveal and $\mathsf{Corrupt}$.

Guess. At some point, $\mathcal{A}_2$ terminates and outputs a guess bit b_{guess}.

Finalize. The challenger sets $\mathsf{lost} \leftarrow \mathsf{true}$ if there exist two (not necessarily distinct) sessions π, π' such that $\pi.\mathsf{sid} = \pi'.\mathsf{sid}$, $\pi.\mathsf{st}_{\mathsf{key}} = \mathsf{revealed}$, and $\pi'.\mathsf{tested} = \mathsf{true}$. (That is, the adversary has tested and revealed the key in a single session or in two partnered sessions.) If $\mathsf{lost} = \mathsf{true}$, the challenger outputs a random bit; otherwise the challenger outputs $[\![b_{\mathsf{guess}} = b_{\mathsf{test}}]\!]$. Note that forward secrecy, if being modelled, is incorporated into the $\mathsf{Corrupt}$ query and need not be stated in the Finalize step.

We say that $\mathcal{A}$ wins the game if $b_{\mathsf{guess}} = b_{\mathsf{test}}$ and $\mathsf{lost} = \mathsf{false}$. We say KE provides $\mathsf{X^cZ\text{-}BR}$ *key secrecy* (with/without forward secrecy) if for all QPT $\mathsf{X^cZ}$ adversaries $\mathcal{A}$ the advantage function

$$\mathsf{Adv}_{\mathsf{KE},\mathcal{D}}^{\mathsf{X^cZ\text{-}BR}}(\mathcal{A}) = \left|\Pr\left[\mathrm{Expt}_{\mathsf{KE},\mathcal{D}}^{\mathsf{X^cZ\text{-}BR}}(\mathcal{A}) \Rightarrow 1\right] - \frac{1}{2}\right|$$

is negligible in the security parameter.

Definition 3 (Two-Stage BR Security)**.** We call a key exchange protocol KE $\mathsf{X^cZ\text{-}BR}$ *secure* (with/without forward secrecy) if KE provides $\mathsf{BR\text{-}Match}$ security (Def. 1) and $\mathsf{X^cZ\text{-}BR}$ key secrecy (with/without forward secrecy) (Def. 2).

4.2.1 Implications.

Similarly to the two-stage security notions of indistinguishability for KEMs, the following implications hold for two-stage BR security:

Theorem 4 ($\mathsf{Q^cQ\text{-}BR} \implies \mathsf{C^cQ\text{-}BR} \implies \mathsf{C^cC\text{-}BR}$)**.** *Let* KE *be an authenticated key exchange protocol. If* KE *is* $\mathsf{Q^cQ\text{-}BR}$ *secure, then* KE *is also* $\mathsf{C^cQ\text{-}BR}$ *secure. If* KE *is* $\mathsf{C^cQ\text{-}BR}$ *secure then it also is* $\mathsf{C^cC\text{-}BR}$ *secure.*

Proof. We show both implications separately and focus on the case of BR key secrecy. A similar argument shows the implications for two-stage BR-Match security and establishes the final result.

- $\mathsf{Q^cQ\text{-}BR}$ **key secrecy** $\implies$ $\mathsf{C^cQ\text{-}BR}$ **key secrecy:** This holds trivially since the adversary in the $\mathsf{C^cQ\text{-}BR}$ key secrecy experiment is a restricted version of the $\mathsf{Q^cQ\text{-}BR}$ adversary with only classical access in the first stage and the oracles for NewSession, Send, and Test removed in the second stage.
- $\mathsf{C^cQ\text{-}BR}$ **key secrecy** $\implies$ $\mathsf{C^cC\text{-}BR}$ **key secrecy:** Assume otherwise, i.e., there exist a scheme KE' that achieves key secrecy in the $\mathsf{C^cQ\text{-}BR}$ sense, but for which there exists an efficient $\mathsf{C^cC\text{-}BR}$ adversary $\mathcal{A}$. An adversary $\mathcal{B}$ can use $\mathcal{A}$ to break the $\mathsf{C^cQ\text{-}BR}$ key secrecy of KE': $\mathcal{B}$ forwards all queries made by $\mathcal{A}$ to its own oracles and responds with their answers. It does not perform a stage change and thus simulates the environment for $\mathcal{A}$ faithfully since it has the same oracles available as $\mathcal{A}$. Once $\mathcal{A}$ outputs a guess bit b_{guess}, $\mathcal{B}$ performs a stage change and outputs the same bit. The winning probability of $\mathcal{B}$ is the same as that of $\mathcal{A}$, contradicting the assumption.

□

4.3 Compilers for Hybrid Authenticated Key Exchange

In the following we present a compiler for authenticated key exchange in the two-stage adversary setting. The compiled protocol, denoted by $\mathcal{C}_{\mathsf{SigMA}}$, combines a passively secure key encapsulation mechanisms (KEMs) with SigMA-style authentication [35]. Figure 12 shows the compiled protocol between Alice and Bob. It takes as input an IND-CPA-secure KEM $\mathcal{K}$, a signature scheme $\mathcal{S}$, a message authentication scheme $\mathcal{M}$ —both existentially unforgeable under chosen-message attack— and a KDF-secure key derivation function KDF, where security is considered with respect to two-stage adversaries. To obtain hybrid authenticated key exchange, the key encapsulation mechanism $\mathcal{K}$ may then be instantiated with any hybrid KEM.

4.3.1 Security Analysis.

We now show that the compiled protocol $\mathcal{C}_{\mathsf{SigMA}}$ achieves *two-stage* BR security (cf. Definition 3). In the theorem below we assume that the key encapsulation mechanism $\mathcal{K}$ is either classically secure or quantum resistant against passive adversaries, i.e., R-ind-cpa = C-ind-cpa or R-ind-cpa = Q-ind-cpa and that the remaining primitives achieve either $\mathsf{C^cC}$, $\mathsf{C^cQ}$, or $\mathsf{Q^cQ}$ security.

One would generally assume that the "weakest primitive" determines the overall security of the compiled protocol. However, it turns out this intuition is not quite correct. Naturally, in case either the unauthenticated key agreement $\mathcal{K}$ or the key derivation function KDF are only classically secure, we cannot expect more than classical $\mathsf{C^cQ\text{-}BR}$ security of the compiled protocol. Similarly, full post-quantum $\mathsf{Q^cQ\text{-}BR}$ security can only be achieved if *all* components of the protocol provide this level of security. Interestingly though, for the compiled protocol to guarantee security against future-quantum adversaries ($\mathsf{C^cQ\text{-}BR}$ security) it suffices for the signature and MAC scheme to be classically secure when combined with Q-ind-cpa-secure key encapsulation and at least $\mathsf{C^cQ}$-secure key derivation. This is due to the fact that, in the proof of the main theorem, Theorem 5, the signatures and message authentication codes of π^* and π^*_{a} must be received while the first stage adversary, which is classical, is present. As soon as the stage change occurs, the adversary loses the power to interfere with still ongoing sessions and message transmissions via the then withdrawn Send oracle.

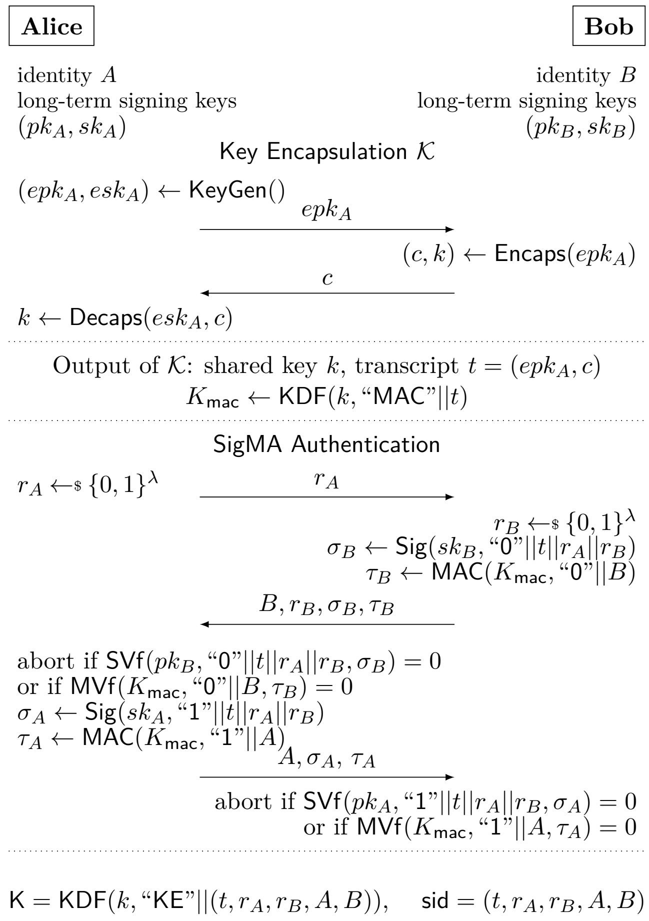

Figure 12: Compiled protocol $\mathcal{C}_{\mathsf{SigMA}}$ - AKE from signatures and MACs

Theorem 5. *Let $\mathcal{K}$ be an* R-ind-cpa *key encapsulation mechanism, $\mathcal{S}$ be an* $\mathsf{S^cT}$*-unforgeable signature scheme, $\mathcal{M}$ be a* $\mathsf{U^cV}$*-unforgeable message authentication scheme, and* KDF *be a* $\mathsf{W^cX}$*-secure key derivation function. Then the compiled protocol $\mathcal{C}_{\mathsf{SigMA}}$ is* $\mathsf{Y^cZ}$-BR *secure with forward secrecy, where*

- $\mathsf{Y^cZ} = \mathsf{C^cC}$*, if either the key encapsulation mechanism $\mathcal{K}$ or the key derivation function* KDF *are only classically secure, i.e., if either* $\mathsf{R} = \mathsf{C}$ *or* $\mathsf{W^cX} = \mathsf{C^cC}$*.*
- $\mathsf{Y^cZ} = \mathsf{Q^cQ}$*, if all components are resistant against fully quantum adversaries, i.e,* $\mathsf{S^cT} = \mathsf{U^cV} = \mathsf{W^cX} = \mathsf{Q^cQ}$ *(and* $\mathsf{R} = \mathsf{Q}$*).*
- $\mathsf{Y^cZ} = \mathsf{C^cQ}$*, if the employed signature and MAC scheme are at most future-quantum secure, i.e., if* $\mathsf{S^cT}, \mathsf{U^cV} \in \{\mathsf{C^cC}, \mathsf{C^cQ}\}$ *(and* $\mathsf{R} = \mathsf{Q}$*,* $\mathsf{W^cX} \geq \mathsf{C^cQ}$*).*

More precisely, for any efficient two-stage $\mathsf{Y^cZ}$ *adversary $\mathcal{A}$ there exist efficient adversaries $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_4$ such that*

$$\begin{aligned} & \mathsf{Adv}^{\mathsf{Y^cZ\text{-}BR}}_{\mathcal{C}_{\mathsf{SigMA}},\mathcal{D}}(\mathcal{A}) \\ \leq \quad & 2^{-|nonce|} \cdot n_s^3 \left(n_u \cdot \mathsf{Adv}^{\mathsf{S^cT\text{-}eufcma}}_{\mathcal{S}}(\mathcal{B}_1) \cdot + \left(\mathsf{Adv}^{\mathsf{M^cN\text{-}pass}}_{\mathcal{K}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{W^cX\text{-}kdf\text{-}sec}}_{\mathsf{KDF}}(\mathcal{B}_3) + \mathsf{Adv}^{\mathsf{U^cV\text{-}eufcma}}_{\mathcal{M}}(\mathcal{B}_4) \right) \right) \end{aligned}$$

where n_s denotes the maximum number of sessions, $|nonce|$ the length of the nonces, and n_u the maximum number of participants.

To prove the theorem, we show the required properties, BR-Match security and $\mathsf{Y^cZ\text{-}BR}$ key secrecy, separately.

Proof Match Security. Let t be the transcript of the key encapsulation mechanism $\mathcal{K}$ between parties A and B . Recall that the session identifier sid is set to be $\mathsf{sid} = (t, r_A, r_B, A, B)$ which consists of public information only. We show that $\mathcal{A}$ cannot achieve any of the three winning conditions defined in Def. 1 with non-negligible probability:

Ad (1) Partnered sessions agree on the session identifier sid, which fixes the transcript t and hence also the input value k to the key derivation function. Consequently, partnered sessions derive the same session keys.

Ad (2) The session identifiers contain the partner identities, thus agreement on the session identifiers implies agreement on the partner identity, excluding the possibility of different intended partners.

Ad (3) For more than two honest sessions to share a session identifier, a third honest session must share a colliding transcript t with the initial two sessions and must have a collision in either of the (randomly chosen) nonces r_A or r_B (depending on whether the third session is initiator or responder). It is easy to see that this occurs only with negligible probability.

□

Proof of Key Secrecy. For the proof we apply the common technique of game hopping. In each game hop we bound the respective difference in the adversary's advantages until the adversary cannot win anymore.

Game 0. The original two-stage BR key secrecy game $\mathrm{Expt}^{\mathsf{Y^cZ\text{-}BR}}_{\mathcal{C}_{\mathsf{SigMA}},\mathcal{D}}(\mathcal{A})$.

Game 1. We start by aborting the game if two sessions of honest parties generate the same nonce r_A or r_B. Let n_s denote the maximum number of sessions and $|nonce|$ the length of the nonces. Since there are n_s possible pairs of randomly sessions, the probability of an abort for this reason is upper-bounded by $n_s^2 \cdot 2^{-|nonce|}$. Hence we have $\mathsf{Adv}^{G_0}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq n_s^2 \cdot 2^{-|nonce|} + \mathsf{Adv}^{G_1}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A})$.

Game 2. For simplicity of the argument it is beneficial to restrict the adversary to a single Test query. This can be done via a standard hybrid argument, guessing the tested session in the beginning, and using the Reveal queries resp. random keys to answer other Test queries. Since session partnering can be checked publicly, we can also answer consistently in such simulated Test queries. Note also that the adversary always has access to the Reveal oracle in the second stage, even if the other queries are prohibited. This strategy reduces the adversary $\mathcal{A}$'s advantage by a factor of at most $\frac{1}{n_s}$: $\mathsf{Adv}^{G_1}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq n_s \cdot \mathsf{Adv}^{G_2}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A})$. From now on, the test session is known in advance and we denote it by π^*. Notice that, in order for the adversary to win, π^* has received all incoming messages and must have accepted before the stage change of $\mathcal{A}$ occurred.

Game 3. We abort the game if the test session π^* run by party $U \in \{A, B\}$ receives a signature σ_V on (“b”$||t||r_A||r_B$) that verifies correctly but has not been signed by some honest party V at this point. Note that this signature must have been received prior to any stage change of the adversary since the second stage of the adversary does not have access to Test. Furthermore, recall that the concerned participants, and thus their long-term secrets, may not be corrupted before the tested session has accepted. In case of a corruption of one of the involved parties after acceptance, forward secrecy is achieved since this does not interfere with the honest generation of the signature σ_V received by session π^* in this game hop.
The probability of an abort happening for this reason can be upper-bounded by the success probability of a reduction $\mathcal{B}_1$ against the $\mathsf{S^cT\text{-}eufcma}$ security of the signature scheme $\mathcal{S}$. The reduction $\mathcal{B}_1$ obtains some public key pk^* as its challenge and proceeds by guessing the party V under whose identity the forgery received by π^* is issued. $\mathcal{B}_1$ generates all key exchange parameters as specified, except for setting $pk_V = pk^*$.

The signing operations by V are performed by relaying the queries to the signature oracle, any other action can be carried out by $\mathcal{B}_1$ itself. If at some point the tested session π^* accepts a signature for a previously unsigned message, then $\mathcal{B}_1$ outputs this message-signature pair as a forgery.

Since the correctly validated signature has not been created by an honest party before, party V cannot have signed ("b"$||t||r_A||r_B$) in the past. At most it could have signed ("b′"$||t||r_A||r_B$) for $\mathsf{b}' = 1 - \mathsf{b}$. With probability $\frac{1}{n_u}$, where n_u is the total number of users, our reduction correctly anticipates the party V, and thus $\mathsf{Adv}^{G_2}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq \mathsf{Adv}^{G_3}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) + n_u \cdot \mathsf{Adv}^{\mathsf{S^cT\text{-}eufcma}}_{\mathcal{S}}(\mathcal{B}_1)$.

Game 4. Next, we guess the honest session π^*_{a} of party V that has issued the valid signature σ_V obtained by π^* in Game 3 and abort if our guess was wrong. Due to the previous game hop such a session must exist. Furthermore it is unique since there is no collision among the nonces due to Game 1. Note that the session π^*_{a} is not necessarily partnered with the test session π^*, since it need not have accepted yet. We therefore refer to this session as *associated.*
This game hop reduces the adversary's advantage by a factor of at most $\frac{1}{n_s}$. Thus, we have $\mathsf{Adv}^{G_3}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq n_s \cdot \mathsf{Adv}^{G_4}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A})$.

Game 5. We now replace the decapsulated value k in both the test session and its associated session by a uniformly random value $\tilde{k}$ of the same length, after the KEM phase and before the parties compute the MAC key. Both parties use the key $\tilde{k}$ instead for the subsequent computations.
If $\mathcal{A}$ were able to distinguish Game 4 and Game 5, then this implies an efficient adversary $\mathcal{B}_2$ against the $\mathsf{R\text{-}ind\text{-}cpa}$ security of $\mathcal{K}$. The reduction $\mathcal{B}_2$ simulates the environment for $\mathcal{A}$ as follows:

- Initially, $\mathcal{B}_2$ receives as a challenge a public key epk^* and a corresponding challenge ciphertext c^* along with the challenge key k^*.
- In order to initialize $\mathcal{A}$, the reduction $\mathcal{B}_2$ generates all key exchange parameters as specified.
- $\mathcal{B}_2$ can initiate any new sessions that $\mathcal{A}$ establishes via $\mathsf{NewSession}$ and can answers all $\mathsf{Corrupt}$ queries with the appropriate long-term secret key.
- $\mathcal{B}_2$ further simulates all Send queries. For Send queries on π^* and π^*_{a}, the adversary $\mathcal{B}_2$ uses (epk^*, c^*) as transcript t for the key encapsulation, creates signatures with the correct signing key of the parties, and computes tags with $K_{\mathsf{mac}} \leftarrow \mathsf{KDF}(k^*, \text{"MAC"}||t)$.
- For all but the predicted sessions π^* and π^*_{a}, algorithm $\mathcal{B}_2$ simulates any Reveal query straightforwardly by itself. For the session π^*_{a}, if it has already accepted, algorithm $\mathcal{B}_2$ derives the session key by using k^* as the allegedly decapsulated key from the KEM phase. Note that, at this point, we have not yet shown that the associated session π^*_{a} is partnered with the test session, such that the adversary could Reveal that session without violating freshness.
- Once $\mathcal{A}$ queries Test on π^*, algorithm $\mathcal{B}_2$ simulates the Test oracle by providing $\mathsf{K}^* = \mathsf{KDF}(k^*, \text{"KE"}||(t, r_U, r_V, U, V))$ to $\mathcal{A}$, where U, V are the respective owners of π^* and π^*_{a}, and r_U, r_V are the nonces exchanged in the authentication. Note that depending on the nature of k^* this then corresponds to either Game 4 or Game 5.

Once $\mathcal{A}$ has output some b_{guess}, then $\mathcal{B}_2$ outputs the same b_{guess} and thus it is easy to see that if $\mathcal{A}$ can win its game with non-negligible probability, so can $\mathcal{B}_2$. Hence, $\mathsf{Adv}^{G_4}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq \mathsf{Adv}^{G_5}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{R\text{-}ind\text{-}cpa}}_{\mathcal{K}}(\mathcal{B}_2)$.

Game 6. We now replace the session key $\mathsf{K} = K_{\mathsf{app}}$ and the MAC key K_{mac} by uniformly random values $\widetilde{K_{\mathsf{app}}}$ and $\widetilde{K_{\mathsf{mac}}}$ in the test session π^* as well as the associated session π^*_{a}. An adversary $\mathcal{A}$ that can efficiently distinguish Game 5 from Game 6 would immediately yield an efficient successful adversary $\mathcal{B}_3$ against the security of KDF: $\mathsf{Adv}^{G_5}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq \mathsf{Adv}^{G_6}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{W^cX\text{-}kdf\text{-}sec}}_{\mathsf{KDF}}(\mathcal{B}_3)$.

Game 7. Next, we abort the game if the associated session π^*_{a} accepts with a different session identifier than the test session, i.e., if $\pi^*_{\mathsf{a}}.\mathsf{sid} \neq \pi^*.\mathsf{sid} \neq \perp$. Since, at this point, all entries in the session identifier

are set except for the partner identity, this can only happen if the adversary can cause π^*_{a} to accept a signature σ_W and MAC τ_W for some identity $W \neq U$. The adversary may indeed sign under an identifier of a previously corrupted party W. However, to succeed $\mathcal{A}$ still needs to forge the corresponding tag τ_W. This tag depends on the MAC key which is derived from the secret value k shared between parties U and V. Similar to Game 3, the probability of an abort happening in this game can be upper-bounded by the success probability of an adversary $\mathcal{B}_4$ against the $\mathsf{U^cV\text{-}eufcma}$-unforgeability of the MAC scheme $\mathcal{M}$. Hence, we have
$\mathsf{Adv}^{G_6}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq \mathsf{Adv}^{G_7}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) + \mathsf{Adv}^{\mathsf{U^cV\text{-}eufcma}}_{\mathcal{M}}(\mathcal{B}_4)$.

To conclude the proof, observe that the adversary expects the challenge value K^* to be a uniformly random string for $b_{\mathsf{test}} = 0$ or to be the output of the key derivation function applied to the output of the key encapsulation mechanism $\mathcal{K}$. These two cases cannot be distinguished by $\mathcal{A}$ since *both* keys are drawn independently and uniformly at random from the key space. Hence, the adversary cannot gain any information about the test bit b_{test} and can do no better than to guess. We thus arrive at the final bound: $\mathsf{Adv}^{G_7}_{\mathcal{C}_{\mathsf{SigMA}}}(\mathcal{A}) \leq 0$. □

5 Conclusion

Hybrid key exchange designs are widely considered as a suitable transitional solution to post-quantum secure key exchange, offering both quantum-resistance as well as preserving today's security guarantees. Despite the profound recent interest in these schemes, the foundational theory behind hybrid key exchange in general, and hybrid key encapsulation mechanisms in particular, is insufficient to establish meaningful security results. Furthermore, no concrete construction for full-fledged hybrid AKE protocols have been proposed so far.

In this work we extended the theory of hybrid KEMs and hybrid AKE by providing security notions that consider adversaries with varying levels of quantum power, thus capturing the adversarial settings that motivate the introduction of hybrid designs.

We examined several combiners for KEMs and prove their robustness in the standard model. We introduced a new combiner, the XOR-then-MAC combiner, which is based on minimal assumptions and constitutes the first hybrid KEM construction which is provably secure against fully quantum adversaries. We further discussed practice-inspired KEM combiners based on dual-PRFs which are closely related to the key schedule used in TLS 1.3.

Finally, we showed how to build post-quantum secure hybrid authenticated key exchange protocols from hybrid key encapsulation mechanisms. We consider it as an interesting open problem to leave the realm of the post-quantum setting and extend the treatment further to fully quantum adversaries.

References

[1] Alagic, G., Gagliardoni, T., Majenz, C.: Unforgeable quantum encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 489–519. Springer, Heidelberg (Apr / May 2018)

[2] Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F., Wunderer, T.: Estimate all the LWE, NTRU schemes! Cryptology ePrint Archive, Report 2018/331 (2018), https://eprint.iacr.org/2018/331

[3] Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum Key Exchange—A New Hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343. USENIX Association, Austin, TX (2016)

[4] Bellare, M.: New proofs for NMAC and HMAC: Security without collision-resistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (Aug 2006)

[5] Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (Aug 1996)

[6] Bellare, M., Canetti, R., Krawczyk, H.: A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In: 30th ACM STOC. pp. 419–428. ACM Press (May 1998)

[7] Bellare, M., Goldreich, O., Mityagin, A.: The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309 (2004), `http://eprint.iacr.org/2004/309`

[8] Bellare, M., Lysyanskaya, A.: Symmetric and dual PRFs from standard assumptions: A generic validation of an HMAC assumption. Cryptology ePrint Archive, Report 2015/1198 (2015), `http://eprint.iacr.org/2015/1198`

[9] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (Aug 1994)

[10] Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (May 1995)

[11] Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.V.: Strengths and weaknesses of quantum computing. SIAM J. Comput. 26(5), 1510–1523 (1997), `https://doi.org/10.1137/S0097539796300933`

[12] Bindel, N., Herath, U., McKague, M., Stebila, D.: Transitioning to a Quantum-Resistant Public Key Infrastructure. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography : 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands Proceedings. pp. 384–405. Springer International Publishing, Cham (2017)

[13] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (Dec 2011)

[14] Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Heidelberg (May 2013)

[15] Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (Aug 2013)

[16] Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 16. pp. 1006–1018. ACM Press (Oct 2016)

[17] Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy. pp. 553–570. IEEE Computer Society Press (May 2015)

[18] Boyd, C., Cliff, Y., Nieto, J.G., Paterson, K.G.: Efficient one-round key exchange in the standard model. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 08. LNCS, vol. 5107, pp. 69–83. Springer, Heidelberg (Jul 2008)

[19] Braithwaite, M.: Google Security Blog: Experimenting with post-quantum cryptography (Jul 2016), https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

[20] Brendel, J., Fischlin, M., Günther, F.: Breakdown resilience of key exchange protocols and the cases of newhope and tls 1.3. Cryptology ePrint Archive, Report 2017/1252 (2017), https://eprint.iacr.org/2017/1252

[21] Brzuska, C.: On the Foundations of Key Exchange. Ph.D. thesis, Technische Universität Darmstadt, Darmstadt, Germany (2013), http://tuprints.ulb.tu-darmstadt.de/3414/

[22] Brzuska, C., Fischlin, M., Warinschi, B., Williams, S.C.: Composability of Bellare-Rogaway key exchange protocols. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) ACM CCS 11. pp. 51–62. ACM Press (Oct 2011)

[23] Costello, C., Easterbrook, K., LaMacchia, B., Longa, P., Naehrig, M.: SIDH Library (Apr 2016), peace-of-mind hybrid key exchange mode https://www.microsoft.com/en-us/research/project/sidh-library/

[24] Dent, A.W.: A Designer's Guide to KEMs. In: Paterson, K.G. (ed.) Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2898, pp. 133–151. Springer (2003), https://doi.org/10.1007/978-3-540-40974-8_12

[25] Dodis, Y., Katz, J.: Chosen-ciphertext security of multiple encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (Feb 2005)

[26] Even, S., Goldreich, O.: On the power of cascade ciphers. ACM Trans. Comput. Syst. 3(2), 108–116 (1985)

[27] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (Aug 1999)

[28] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013)

[29] Giacon, F., Heuer, F., Poettering, B.: KEM combiners. In: Abdalla, M., Dahab, R. (eds.) PKC 2018, Part I. LNCS, vol. 10769, pp. 190–218. Springer, Heidelberg (Mar 2018)

[30] Harnik, D., Kilian, J., Naor, M., Reingold, O., Rosen, A.: On robust combiners for oblivious transfer and other primitives. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 96–113. Springer, Heidelberg (May 2005)

[31] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017, Part I. LNCS, vol. 10677, pp. 341–371. Springer, Heidelberg (Nov 2017)

[32] Jager, T., Kohlar, F., Schäge, S., Schwenk, J.: On the security of TLS-DHE in the standard model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 273–293. Springer, Heidelberg (Aug 2012)

[33] Katz, J., Lindell, A.Y.: Aggregate message authentication codes. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 155–169. Springer, Heidelberg (Apr 2008)

[34] Katz, J., Yung, M.: Scalable protocols for authenticated group key exchange. Journal of Cryptology 20(1), 85–113 (Jan 2007)

[35] Krawczyk, H.: SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 400–425. Springer, Heidelberg (Aug 2003)

[36] Langley, A.: Intent to Implement and Ship: CECPQ1 for TLS (Jul 2016), Google group `https://groups.google.com/a/chromium.org/forum/#!topic/security-dev/DS9pp2U0SAc`

[37] Li, Y., Schäge, S., Yang, Z., Bader, C., Schwenk, J.: New modular compilers for authenticated key exchange. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) ACNS 14. LNCS, vol. 8479, pp. 1–18. Springer, Heidelberg (Jun 2014)

[38] National Institute of Standards and Technology (NIST): Post-quantum cryptography: Nist's plan for the future. `https://csrc.nist.gov/projects/post-quantum-cryptography` (Aug 19, 2015)

[39] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)

[40] Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018), `https://rfc-editor.org/rfc/rfc8446.txt`

[41] de Saint Guilhem, C., Smart, N.P., Warinschi, B.: Generic Forward-Secure Key Agreement Without Signatures. In: Information Security - 20th International Conference, ISC 2017, Ho Chi Minh City, Vietnam, November 22-24, 2017, Proceedings. pp. 114–133. Springer International Publishing (2017), `https://doi.org/10.1007/978-3-319-69659-1_7`

[42] Schank, J., Stebila, D.: A Transport Layer Security (TLS) Extension For Establishing An Additional Shared Secret draft-schanck-tls-additional-keyshare-00. `https://tools.ietf.org/html/draft-schanck-tls-additional-keyshare-00` (apr 2017)

[43] Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 192–216. Springer, Heidelberg (Oct / Nov 2016)

[44] de Valence, H.: SIDH in Go for quantum-resistant TLS 1.3 (Sep 2017), `https://blog.cloudflare.com/sidh-go/`

[45] Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. 22(3), 265–279 (1981), `https://doi.org/10.1016/0022-0000(81)90033-7`

[46] Whyte, W., Fluhrer, S., Zhang, Z., Garcia-Morchon, O.: Quantum-Safe Hybrid (QSH) Key Exchange for Transport Layer Security (TLS) version 1.3 draft-whyte-qsh-tls13-06. `https://tools.ietf.org/html/draft-whyte-qsh-tls13-06` (Oct 2017)

[47] Zhandry, M.: How to construct quantum random functions. In: 53rd FOCS. pp. 679–687. IEEE Computer Society Press (Oct 2012)

[48] Zhang, R., Hanaoka, G., Shikata, J., Imai, H.: On the security of multiple encryption or CCA-security+CCA-security=CCA-security? In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 360–374. Springer, Heidelberg (Mar 2004)

Supplementary Material

A Introduction to Quantum Computing

In the following we give a brief introduction of quantum computation knowledge used in this paper. A standard text for a more complete explanation is for example given by Nielsen and Chuang [39].

Let $\mathcal{H}$ be a complex Hilbert space with inner product $\langle y|x\rangle$ with vectors $|x\rangle, |y\rangle \in \mathcal{H}$. A quantum state is an element in $\mathcal{H}$ of norm 1. Let $\{|x\rangle\}_x$ be a basis for $\mathcal{H}$, then we can represent any quantum state $|y\rangle$ as $|y\rangle = \sum_x \psi_x |x\rangle$ where ψ_x are complex numbers such that $|y\rangle$ has norm 1. Operations on elements of $\mathcal{H}$, i.e., quantum operations, are represented by unitary transformations $\mathbf{U}$. Hence, quantum operations (prior to measurement) are reversible. This impacts the quantization of classical operations, such as for classical or quantum decapsulation oracles, as follows.

Let A be a classical algorithm with input $x \in \{0,1\}^a$ and output $y \in \{0,1\}^b$. Moreover, let $\{0,1\}^a \times \{0,1\}^b \to \{0,1\}^a \times \{0,1\}^b : (x,t) \mapsto (x, t \oplus A(x))$ be a classical reversible mapping. The corresponding unitary transformation $\mathbf{A}$ acting linearly on quantum states is given by $\mathbf{A} : \sum_{x,t} \psi_{x,t} |x,t\rangle \mapsto \sum_{x,t} \psi_{x,t} |x, t \oplus A(x)\rangle$. We may add a workspace register to the input and the output registers to allow for more generality. Thus, the quantized classical algorithm is given as $\mathbf{A} : \sum_{x,t,z} \psi_{x,t,z} |x,t,z\rangle \mapsto \sum_{x,t,z} \psi_{x,t,z} |x, t \oplus A(x), z\rangle$.

B Definitions

In this section we adapt traditional security definitions to our two-stage model.

B.1 One-Way Security of KEMs Against Partially or Fully-Quantum Adversaries.

First we define unified security experiments for one-way security against partially or fully quantum adversaries since some of our results, e.g., Proposition 3 relies on this notion. During the one-way security experiment of KEMs, the adversary's task is to fully recover the session key, not just distinguish it from random as in the indistinguishability notions of Section 2. We can similarly consider classical and quantum adversaries for this security goal, in both the chosen-plaintext and chosen-ciphertext scenarios. Figure 13 shows the corresponding security experiments for one-way chosen-plaintext (Z-ow-cpa) and one-way chosen-ciphertext ($\mathsf{X}^{\mathsf{y}}\mathsf{Z}$-ow-cca) attacks.

$\mathrm{Expt}_{\mathcal{K}}^{\mathsf{Z\text{-}ow\text{-}cpa}}(\mathcal{A})$:
$H \leftarrow_\$ \mathcal{H}_{\mathcal{K}}$
$q_H \leftarrow 0$
$(sk, pk) \leftarrow \mathsf{KeyGen}()$
$(c^*, \kappa^*) \leftarrow \mathsf{Encaps}(pk)$
$\kappa' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{X}}_H(\cdot)}(pk, c^*)$
return $[\![\kappa^* = \kappa']\!]$

$\mathrm{Expt}_{\mathcal{K}}^{\mathsf{X^yZ\text{-}ow\text{-}cca}}(\mathcal{A})$:
$H \leftarrow_\$ \mathcal{H}_{\mathcal{K}}$
$q_D \leftarrow 0,\ q_H \leftarrow 0$
$(sk, pk) \leftarrow \mathsf{KeyGen}()$
$(c^*, \kappa^*) \leftarrow \mathsf{Encaps}(pk)$
$st \leftarrow \mathcal{A}_1^{\mathcal{O}^{\mathsf{X}}_H(\cdot), \mathcal{O}^{\mathsf{y}}_D(\cdot)}(pk, c^*)$
$\kappa' \leftarrow \mathcal{A}_2^{\mathcal{O}^{\mathsf{Z}}_H(\cdot)}(st)$
return $[\![\kappa^* = \kappa']\!]$

Figure 13: Security experiment for one-way security of a KEM $\mathcal{K}$ under chosen-plaintext attacks against a classical ($\mathsf{Z} = \mathsf{C}$) or quantum ($\mathsf{Z} = \mathsf{Q}$) adversary $\mathcal{A}$ (left), and under chosen-ciphertext attack against a two-stage $\mathsf{X}^{\mathsf{y}}\mathsf{Z}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ (right), in the classical or quantum random oracle model. Oracles $\mathcal{O}^{\mathsf{C}}_H(\cdot)$, $\mathcal{O}^{\mathsf{Q}}_H(\cdot)$, $\mathcal{O}^{\mathsf{c}}_D(\cdot)$, and $\mathcal{O}^{\mathsf{q}}_D(\cdot)$ as in Figures 2 and 3.

B.2 PRF Security in the Two-Stage Model.

Two of our combiners, namely the *dual-PRF* combiner $\mathsf{dualPRF}$ in Section 3.2 and the *nested dual-PRF* combiner N in Section 3.3 are based on the security of pseudorandom functions and dual pseudorandom functions defined as in Definitions 4 and 5, respectively.

Definition 4 (Two-Stage PRF Security)**.** Let λ be the security parameter and let $F : \mathsf{Keys} \times \mathsf{In} \to \mathsf{Out}$ be a pseudorandom function. Define $\mathsf{Func}[\mathsf{In}, \mathsf{Out}]$ to be the set of all functions $f : \mathsf{In} \to \mathsf{Out}$. Furthermore, let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two-stage QPT $\mathsf{X^yZ}$ adversary interacting with F in the Game $\mathrm{Expt}_F^{\mathsf{prf\text{-}sec}}(\cdot)$ given in Figure 14. $\mathcal{A}$ wins the game if $\mathrm{Expt}_F^{\mathsf{prf\text{-}sec}}(\mathcal{A}) = 1$.

We say that F is a $\mathsf{X^yZ}$-secure pseudorandom function ($\mathsf{prf\text{-}sec}$) if for all QPT $\mathsf{X^yZ}$ adversaries $\mathcal{A}$ the advantage function

$$\mathsf{Adv}_F^{\mathsf{prf\text{-}sec}}(\mathcal{A}) = \Pr\left[\mathrm{Expt}_F^{\mathsf{prf\text{-}sec}}(\mathcal{A}) = 1\right]$$

is negligible in the security parameter λ.

$\underline{\mathrm{Expt}_F^{\mathsf{prf\text{-}sec}}(\mathcal{A})}$:
$k \leftarrow\!\!\$\ \mathsf{Keys}_F$
$b \leftarrow\!\!\$\ \{0,1\}$
$f \leftarrow\!\!\$\ \mathsf{Func}[\mathsf{In}, \mathsf{Out}]$
$st \leftarrow \mathcal{A}_1^{\mathcal{O}_F^{\mathsf{y}}(\cdot)}()$
$b' \leftarrow \mathcal{A}_2(st)$
return $[[b = b']]$

$\underline{\text{Quantum } \mathcal{O}_F^{\mathsf{y}}(\sum_{x,t,z} \psi_{x,t,z} |x, t, z\rangle)}$:
Return state $\sum_{x,t,z} \psi_{x,t,z} |x, t \oplus \mathcal{O}_F^{\mathsf{y}}(x), z\rangle$

$\underline{\text{Classical } \mathcal{O}_F^{\mathsf{y}}(x)}$:
if $b = 1$ return $f(x)$
else return $F(k, x)$

Figure 14: PRF security definition for a two-stage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$; if $\mathsf{y} = \mathsf{c}$ then $\mathcal{A}_1$ has classical access to the oracle $\mathcal{O}_F^{\mathsf{y}}(\cdot)$, otherwise quantum access.

Definition 5 (Two-Stage Dual PRF Security)**.** Let λ be the security parameter and let $F : \mathsf{Keys} \times \mathsf{In} \to \mathsf{Out}$ be a pseudorandom function. Furthermore, define $F^{\mathsf{swap}} : \mathsf{In} \times \mathsf{Keys} \to \mathsf{Out}$. We say that F is a *dual PRF* if both F and F^{swap} are pseudorandom functions.

We say that F is a $\mathsf{X^yZ}$-secure dual pseudorandom function ($\mathsf{dprf\text{-}sec}$) if for all QPT $\mathsf{X^yZ}$ adversaries $\mathcal{A}$ the advantage function

$$\mathsf{Adv}_F^{\mathsf{dprf\text{-}sec}}(\mathcal{A}) = \max\{\mathsf{Adv}_F^{\mathsf{prf\text{-}sec}}(\mathcal{A}), \mathsf{Adv}_{F^{\mathsf{swap}}}^{\mathsf{prf\text{-}sec}}(\mathcal{A})\}$$

is negligible in the security parameter λ.