Generic Authenticated Key Exchange in the Quantum Random Oracle Model

Kathrin Hövelmanns[1], Eike Kiltz[1], Sven Schäge[1], and Dominique Unruh[2]

[1] Ruhr-Universität Bochum
{kathrin.Hoevelmanns,eike.kiltz,sven.schaege}@rub.de
[2] University of Tartu
unruh@ut.ee

Abstract. We propose $\mathsf{FO}_{\mathsf{AKE}}$, a generic construction of two-message authenticated key exchange (AKE) from any passively secure public key encryption (PKE) in the quantum random oracle model (QROM). Whereas previous AKE constructions relied on a Diffie-Hellman key exchange or required the underlying PKE scheme to be perfectly correct, our transformation allows arbitrary PKE schemes with non-perfect correctness. Dealing with imperfect schemes is one of the major difficulties in a setting involving active attacks. Our direct construction, when applied to schemes such as the submissions to the recent NIST post-quantum competition, is more natural than previous AKE transformations. Furthermore, we avoid the use of (quantum-secure) digital signature schemes which are considerably less efficient than their PKE counterparts. As a consequence, we can instantiate our AKE transformation with any of the submissions to the recent NIST competition, e.g., ones based on codes and lattices.

$\mathsf{FO}_{\mathsf{AKE}}$ can be seen as a generalisation of the well known Fujisaki-Okamoto transformation (for building actively secure PKE from passively secure PKE) to the AKE setting. As a helper result, we also provide a security proof for the Fujisaki-Okamoto transformation in the QROM for PKE with non-perfect correctness which is tighter and tolerates a larger correctness error than previous proofs.

Keywords. Authenticated key exchange, quantum random oracle model, NIST, Fujisaki-Okamoto.

1 Introduction

Authenticated Key Exchange. Besides public key encryption (PKE) and digital signatures, authenticated key exchange (AKE) is arguably one of the most important cryptographic building blocks in modern security systems. In the last two decades, research on AKE protocols has made tremendous progress in developing more solid theoretical foundations [10,19,38,31] as well as increasingly efficient designs of AKE protocols [37,47,44]. Most AKE protocols rely on constructions based on an ad-hoc Diffie-Hellman key exchange that is authenticated either via digital signatures, non-interactive key exchange (usually a Diffie-Hellman key exchange performed on long-term Diffie-Hellman keys), or public key encryption. While in the literature one can find many protocols that use one of the two former building blocks, results for PKE-based authentication are rather rare [8,17]. Even rarer are constructions that only rely on PKE, discarding Diffie-Hellman key exchanges entirely. Notable recent exceptions are [23,24] and the protocol in [2], the latter of which has been criticised for having a flawed security proof and a weak security model [46,39].

The NIST Post-Quantum Competition. Recently, some of the above mentioned designs have gathered renewed interest in the quest of finding AKE protocols that are secure against quantum adversaries, i.e., adversaries equipped with a quantum computer. In particular, the National Institute of Standards and Technology (NIST) announced a competition with the goal to standardise new PKE and signature algorithms [41] with security against quantum adversaries. With the understanding that an AKE protocol can be constructed from low level primitives such as quantum-secure PKE and signature schemes, the NIST did not require the submissions to describe a concrete AKE protocol. Many PKE and signature candidates base their security on the hardness of certain problems over lattices and codes, which are generally believed to resist quantum adversaries.

The quantum ROM. Quantum computers may execute all "offline primitives" such as hash functions on arbitrary superpositions, which motivated the introduction of the quantum (accessible) random oracle model (QROM) [14]. While the adversary's capability to issue quantum queries to the random oracle renders many proof strategies significantly more complicated, it is nowadays generally believed that only proofs in the QROM imply provable security guarantees against quantum adversaries.

AKE and Quantum-Secure Signatures. Digital signatures are useful for the "authentication" part in AKE, but unfortunately all known quantum-secure constructions would add a considerable overhead to the AKE protocol. Therefore, if at all possible, we prefer to build AKE protocols only from PKE schemes, without using signatures.[3] Our ultimate goal is to build a system that remains secure in the presence of quantum computers, meaning that even currently employed (very fast) signatures schemes based on elliptic curves are not an option.

Central Research Question for Quantum-Secure AKE. In summary, motivated by post-quantum secure cryptography and the NIST competition, we are interested in the following question:

> **How to build an actively secure AKE protocol from any passively secure PKE in the quantum random oracle model, without using signatures?**

(The terms "actively secure AKE" and "passively secure PKE" will be made more precise later.) Surprisingly, one of the main technical difficulties is that the underlying PKE scheme might come with a small probability of decryption failure, i.e., first encrypting and then decrypting does not yield the original message. This property is called non-perfect correctness, and it is common for quantum-secure schemes from lattices and codes, rendering them useless for all previous constructions that relied on perfect correctness.[4]

Previous Constructions of AKE from public-key primitives. The generic AKE protocol of Fujioka et al. [23] (itself based on [17]) transforms a passively secure PKE scheme PKE and an actively (i.e., $\mathsf{IND\text{-}CCA}$) secure PKE scheme $\mathsf{PKE}_{\mathrm{cca}}$ into an AKE protocol. We will refer to this

[3] Clearly, PKE requires a working public-key infrastructure (PKI) which in turn requires signatures to certify the public-key. However, a user only has to verify a given certificate once and for all, which means the overhead of a quantum-secure signature can be neglected.

[4] There exist generic transformations that can immunise against decryption errors (e.g., [22]). Even though they are quite efficient in theory, the induced overhead is still not acceptable for practical purposes. While lattice schemes could be rendered perfectly correct by putting a limit on the noise, and setting the modulus of the LWE instance large enough (see, e.g., [12,29]), the security level cannot be maintained without increasing the problem's dimension, accordingly. Since this modification would lead to increased public-key and ciphertext length, many NIST submissions deliberately made the design choice of having imperfect correctness.

transformation as $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{PKE}_{\text{cca}}]$. Since the FSXY transformation is in the standard model, it is likely to be secure with the same proof in the post-quantum setting and thus also in the QROM. The standard way to obtain actively secure encryption from passively secure ones is the Fujisaki-Okamoto transformation $\mathsf{PKE}_{\text{cca}} = \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ [25,26]. In its "implicit rejection" variant [28], it comes with a recently discovered security proof [43] that models the hash functions G and H as quantum random oracles. Indeed, the *combined AKE transformation* $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]]$ transforms passively secure encryption into AKE that is very likely to be secure in the QROM, without using digital signatures, hence giving a first answer to our above question. It has, however, two main drawbacks.

– **Perfect correctness requirement.** Transformation FSXY is not known to have a security proof if the underlying scheme does not satisfy perfect correctness. Likewise, the relatively tight QROM proof for FO that was given in [43] requires the underlying scheme to be perfectly correct, and a generalisation of the proof for schemes with non-perfect correctness is not straightforward. Hence, it is unclear whether $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]]$ can be instantiated with lattice- or code-based encryption schemes.
– **Lack of simplicity.** The Fujisaki-Okamoto transformation already involves hashing the key using hash function H, and FSXY involves even more (potentially redundant) hashing of the (already hashed) session key. Overall, the combined transformation seems overly complicated and hence impractical.

In [24], a transformation was given that started from oneway-secure KEMs, but its security proof was given in the ROM, and its generalisation to the QROM was explicitly left as an open problem. Furthermore, it involves more hashing, similar to transformation FSXY.

Hence, it seems desirable to provide a simplified transformation that gets rid of unnecessary hashing steps, and that can be proven secure in the QROM even if the underlying scheme does not satisfy perfect correctness. As a motivating example, note that the Kyber AKE protocol [16] can be seen as a result of applying such a simplified transformation to the Kyber PKE scheme, although coming without a formal security proof.

1.1 Our Contributions

Our main contribution is a transformation, $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ ("Fujisaki-Okamoto for AKE") that converts any passively secure encryption scheme into an actively secure AKE protocol, with provable security in the quantum random oracle model. It can deal with non-perfect correctness and does not use digital signatures. Our transformation $\mathsf{FO}_{\mathsf{AKE}}$ can be viewed as a modification of the transformation given in [24]. Furthermore, we provide a precise game-based security definition for two-message AKE protocols. As a side result, we also give a security proof for the Fujisaki-Okamoto transformation in the QROM in Section 3 that deals with correctness errors. It can be seen as the KEM analogue of our main result, the AKE proof. Our proof strategy differs from and improves on the bounds of a previously published proof of the Fujisaki-Okamoto transformation for KEMs in the QROM [32].

FO transformation for KEMs. To simplify the presentation of $\mathsf{FO}_{\mathsf{AKE}}$, we first give some background on the Fujisaki-Okamoto transformation for KEMs. In its original form [25,26], FO yields an encryption scheme that is IND-CCA secure in the random oracle model [9] from combining any One-Way secure asymmetric encryption scheme with any one-time secure symmetric encryption scheme. In "A Designer's Guide to KEMs", Dent [21] provided FO-like $\mathsf{IND\text{-}CCA}$ secure KEMs. (Recall that any $\mathsf{IND\text{-}CCA}$ secure Key Encapsulation Mechanism can be combined with any (one-time) chosen-ciphertext secure symmetric encryption scheme to obtain a $\mathsf{IND\text{-}CCA}$ secure PKE scheme [20].) Since all of the transformations mentioned above required the underlying PKE scheme to be perfectly correct, and due to the increased popularity of lattice-based schemes with non-perfect correctness, [28] gave several modularisations of FO-like transformations and proved them robust against correctness errors. The key observation was that FO-like transformations essentially consists of two separate steps and can be dissected into two transformations, as sketched in the introduction of [28]:

- Transformation T: "Derandomise" and "re-encrypt". Starting from an encryption scheme PKE and a hash function G, encryption of $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is defined by

$$\mathsf{Enc}'(pk, m) := \mathsf{Enc}(pk, m; \mathsf{G}(m)),$$

  where $\mathsf{G}(m)$ is used as the random coins for Enc, rendering Enc' deterministic. $\mathsf{Dec}'(sk, c)$ first decrypts c into m' and rejects if $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$ ("re-encryption").
- Transformation $\mathsf{U}^{\not\perp}_m$: "Hashing". Starting from an encryption scheme PKE' and a hash function H, key encapsulation mechanism $\mathsf{KEM}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m[\mathsf{PKE}', \mathsf{H}]$ with "implicit rejection" is defined by

$$\mathsf{Encaps}(pk) := (c \leftarrow \mathsf{Enc}'(pk, m), K := \mathsf{H}(m)), \tag{1}$$

  where m is picked at random from the message space, and

$$\mathsf{Decaps}(sk, c) = \begin{cases} \mathsf{H}(m) & m \neq \bot \\ \mathsf{H}(s, c) & m = \bot \end{cases},$$

  where $m := \mathsf{Dec}(sk, c)$ and s is a random seed which is contained in sk. In the context of the FO transformation, implicit rejection was first introduced by Persichetti [42, Sec. 5.3].

Transformation T was proven secure both in the (classical) ROM and the QROM, and $\mathsf{U}^{\not\perp}_m$ was proven secure in the ROM. To achieve QROM security, [28] gave a modification of $\mathsf{U}^{\not\perp}_m$, called $\mathsf{QU}^{\not\perp}_m$, but its security proof in the QROM suffered from a quartic[5] loss in tightness, and furthermore, most real-world proposals are designed such that they fit the framework of $\mathsf{FO}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m \circ \mathsf{T}$, not $\mathsf{QU}^{\not\perp}_m \circ \mathsf{T}$.

A slightly different modularisation was introduced in [43]: they gave transformations TPunc ("Puncturing and Encrypt-with-Hash") and SXY ("Hashing with implicit reject and reencryption"). SXY differs from $\mathsf{U}^{\not\perp}_m$ in that it reencrypts during decryption. Hence, it can only be applied to deterministic schemes. Even in the QROM, its CCA security tightly reduces to an intermediate notion called Disjoint Simulatability (DS) of ciphertexts. Intuitively, disjoint simulatability means that we can efficiently sample "fake ciphertexts" that are computationally indistinguishable from real PKE ciphertexts ("simulatability"), while the set of possible fake ciphertexts is required to be (almost) disjoint from the set of real ciphertexts. DS is naturally satisfied by many code/lattice-based encryption schemes. Additionally, it can be achieved using transformation Punc, i.e., by puncturing the underlying schemes' message space at one point and using this message to sample fake encryptions. Deterministic DS can be achieved by using transformation TPunc, albeit non-tightly: the reduction suffers from quadratic loss in security and an additional factor of q, the number of the adversary's hash queries.

[5] not just quadratic, but indeed quartic

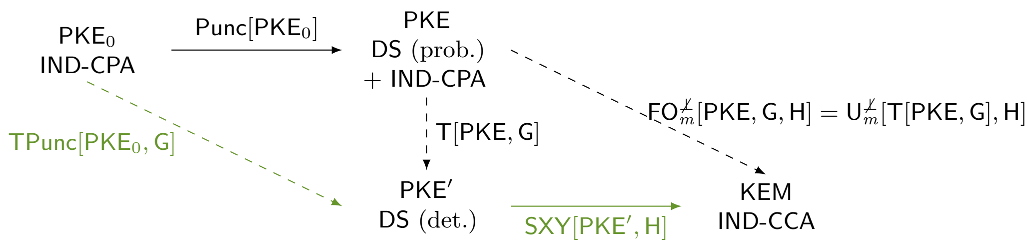

Fig. 1: Comparison of [43]'s modular transformation (green) with ours. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions.

However, the reduction that is given in [43] requires the underlying encryption scheme to be perfectly correct. Later, [32] gave non-modular security proofs for the transformations $\mathsf{FO}^{\not\perp}_m$ and $\mathsf{FO}^{\not\perp}$ as well as a security proof for SXY[6] for schemes with correctness errors, which still suffered from quadratic loss in security and an additional factor of q, the latter of which this work improves to $\sqrt{q}$.

Our transformation $\mathsf{FO}^{\not\perp}_m$ can be applied to any PKE scheme that is both IND-CPA and DS secure. The reduction is tighter than the one that results from combining those of TPunc and SXY in [43], and also than the reduction given in [33]. This is due to our use of the improved Oneway-to-Hiding lemma [3, Thm. 1: "Semi-classical O2H"]. Furthermore, we achieve a better correctness bound (the square of the bound given in [33]) due to a better bound for the generic distinguishing problem. In cases where PKE is not already DS, this requirement can be waived with negligible loss of efficiency: To rely on IND-CPA alone, all that has to be done is to plug in transformation Punc. A visualisation is given in Figure 1.

Security Model for Two-Message Authenticated Key Exchange. We introduce a simple game-based security model for (non-parallel) two-message AKE protocols, i.e., protocols where the responder sends his message only after having received the initiator's message. Technically, in our model, and similar to previous literature, we define several oracles that the attacker has access to. However, in contrast to most other security models, the inner workings of these oracles and their management via the challenger are precisely defined with pseudo-code.

Details on our Models. We define two security notions for two-message AKEs: key indistinguishability against active attacks (IND-AA) and the weaker notion of indistinguishability against active attacks without state reveal in the test session (IND-StAA). IND-AA captures the classical notion of key indistinguishability (as introduced by Bellare and Rogaway [10]) as well as security against reflection attacks, key compromise impersonation (KCI) attacks, and weak forward secrecy (wFS) [37]. It is based on the Canetti-Krawczyk (CK) model and allows the attacker to reveal (all) secret state information as compared to only ephemeral keys. As already pointed out by [17], this makes our model incomparable to the eCK model [38] but strictly stronger than the CK model. Essentially, the IND-AA model states that the session key remains indistinguishable from a random one even if

[6] Note that nomenclature of [33] is a bit misleading: while the respective KEM is called $\mathsf{U}^{\not\perp}_m$, it is actually transformation SXY (it reencrypts during decryption, which $\mathsf{U}^{\not\perp}_m$ does not).

1. the attacker knows either the long-term secret key or the secret state information (but not both) of both parties involved in the test session, as long as it did not modify the message received by the test session,
2. and also if the attacker modified the message received by the test session, as long as it did not obtain the long-term secret key of the test session's peer.

We also consider the slightly weaker model IND-StAA (in which we will prove the security of our AKE protocols), where 2. is substituted by

2'. and also if the attacker modified the message received by the test session, as long as it did neither obtain the long-term secret key of the test session's peer **nor the test session's state**. The latter strategy, we will call a *state attack*.

We remark that IND-StAA security is essentially the same notion that was achieved by the FSXY transformation [23].[7] In the full version we provide a more general perspective on how our model compares to existing ones.

Our Authenticated Key-Exchange Protocol. Our transformation $\mathsf{FO}_{\mathsf{AKE}}$ transforms any passively secure PKE (with potential non-perfect correctness) into an IND-StAA secure AKE. $\mathsf{FO}_{\mathsf{AKE}}$ is a simplification of the transformation $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]]$ mentioned above, where the derivation of the session key K uses only one single hash function H. $\mathsf{FO}_{\mathsf{AKE}}$ can be regarded as the AKE analogue of the Fujisaki-Okamoto transformation.

Transformation $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is described in Figure 2 and uses transform $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ as a building block. (The full construction is given in Figure 15, see Section 5.) Our main security result (Theorem 3) states that $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is an IND-StAA-secure AKE if the underlying probabilistic PKE is DS as well as IND-CPA secure and has negligible correctness error, and furthermore G and H are modeled as quantum random oracles.

The proof essentially is the AKE analogue to the security proof of $\mathsf{FO}_m^{\not\perp}$ we give in Section 3.2: By definition of our security model, it always holds that at least one of the messages m_i, m_j and $\tilde{m}$ is hidden from the adversary (unless it loses trivially) since it may not reveal a party's secret key and its session state at the same time. Adapting the simulation technique in [43], we can simulate the session keys even if we do not know the corresponding secret key sk_i (sk_j, $\tilde{sk}$). Assuming that PKE is DS, we can replace the corresponding ciphertext c_i (c_j, $\tilde{c}$) of the test session with a fake ciphertext, rendering the test session's key completely random from the adversary's view due to PKE's disjointness.

Let us add two remarks. Firstly, we cannot prove the security of $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ in the stronger sense of IND-AA and actually, it is not secure against state attacks. Secondly, note that our security statement involves the probabilistic scheme PKE rather than PKE′. Unfortunately, we were not able to provide a modular proof of AKE solely based on reasonable security properties of $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$. The reason for this is indeed the non-perfect correctness of PKE. This difficulty corresponds to the difficulty to generalise [43]'s result for deterministic encryption schemes with correctness errors discussed above.

[7] The difference is that the model from [23] furthermore allows a "partial reveal" of the test session's state. For simplicity and due to their little practical relevance, we decided not to include such partial session reveal queries in our model. We remark that, however, our protocol could be proven secure in this slightly stronger model.

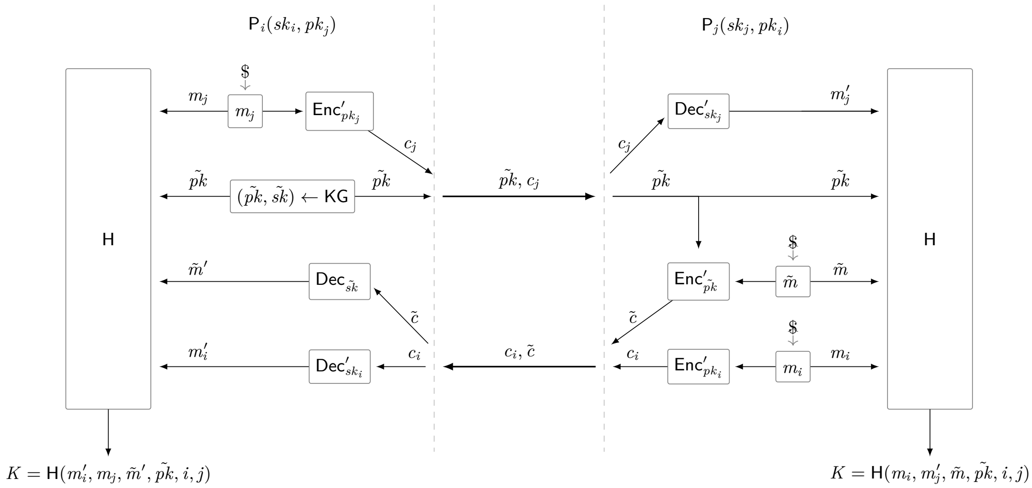

Fig. 2: A visualisation of our authenticated key-exchange protocol $\mathsf{FO}_{\mathsf{AKE}}$. We make the convention that, in case any of the Dec' algorithms returns $\bot$, the session key K is derived deterministically and pseudorandomly from the player's state ("implicit rejection").

Concrete Applications. Our transformation can be applied to any scheme that is IND-CPA secure with post-quantum security, e.g., Frodo [40], Kyber [16], and Lizard [5]. Recall that the additional requirement of DS can be achieved with negligible loss of efficiency. However, in many applications even this negligible loss is inexistent since most of the aforementioned schemes can already be proven DS under the same assumption that their IND-CPA security is based upon.

Subsequent work. Since this paper was published on eprint, there has been more work on CCA security of FO in the QROM ([35,13]), essentially achieving the same level of tightness as this work. [13] achieves more modularity, and covers a class of schemes that is both less and more restrictive at the same time: They only require schemes to be oneway-secure (instead of CPA, as required in this work), but the schemes have to meet an additional injectivity requirement (specified below).

Tightness for FO. Reductions from CCA security to CPA security in the QROM usually suffer from tightness loss in two separate ways: The best known bounds for probabilistic schemes to this date are essentially of the form $\sqrt{q}\sqrt{\epsilon}$, where q is the number of the adversary's hash queries, and ϵ is the reduction's CPA advantage. Hence, the loss consists of both a loss regarding q (q-nontightness), and worse, a quadratic loss regarding the level of CPA security (root-nontightness). For the general setting where one starts from a probabilistic scheme, there have not been tightness improvements since this work:

Essentially, [35] is an update of [32] that makes use of the improved Oneway-to-Hiding bounds given in [3], thereby improving [32]'s bound $q\sqrt{\epsilon}$ to $\sqrt{q}\sqrt{\epsilon}$, with the security requirement switching from onewayness to IND-CPA. The result seems to differ from this work solely in its (nonmodular) proof structure.

In [13], a new modular proof for FO was given by starting from probabilistic onewayness and choosing deterministic oneway-security as their intermediate[8] notion, opposed to our (strictly stronger) intermediate notion of deterministic DS. This approach matches the observation that if one can start from a scheme that already is deterministically oneway-secure (like [12]), derandomisation step T is superfluous. In this case, only transformation U has to be applied, which is proven secure q-tightly. The weaker intermediate notion, however, shifts the root-nontightness to second transformation U. Therefore, the result still is heavily non-tight, even if derandomising via T is skipped. Furthermore, no tightness improvements whatsoever are achieved if the underlying scheme is not already deterministic, and thus has to be derandomised using T first.

MODULARITY. The modular proof of [13] is achieved by introducing an additional notion for the intermediate scheme that deals with correctness errors. Unfortunately, the possibility of correctness errors complicate modular attempts on analysing FO: For underlying probabilistic schemes, [13] requires more than this work since its approach only is applicable if the "intermediate" scheme is injective with overwhelming probability. It is very likely that the modular approach of [13] could be generalised to an AKE proof that similarly is modular and hence, conceptually nicer. But this gain in modularity would come at a cost: The approach only is applicable if the derandomised scheme is essentially injective. We would, therefore, add an unnecessary restriction on the class of schemes that AKE can be based upon.

Open Problems. In the literature, one can find several Diffie-Hellman based protocols that achieve IND-AA security, for example HMQV [37]. However, none of them provides security against quantum computers. We leave as an interesting open problem to design a generic and efficient two-message AKE protocol in our stronger IND-AA model, preferably with a security proof in the QROM to guarantee its security even in the presence of quantum adversaries.

While [13] gave a proof of CCA security that is conceptually cleaner, it still is heavily non-tight due to its root-nontightness, with the root-nontightness stemming from its usage of a standard Oneway-to-Hiding strategy. Recent work [34] proved that for reductions using this standard approach, suffering from quadratic security loss is inevitable. We would like to point out, however, that we do not view this result as an impossibility result[9]. It rather proves impossibility of root-tightness *for a certain type of reduction*, and thereby informs us how to adapt possible future proof strategies: A root-tight proof of CCA security still might be achievable, but the respective reduction would have to be more sophisticated than extracting oneway solutions for the underlying scheme by simply applying Oneway-to-Hiding.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \ldots, n\}$. For a set S, $|S|$ denotes the cardinality of S. For a finite set S, we denote the sampling of a uniform random element x by $x \leftarrow_\$ S$, while we denote the sampling according to some distribution $\mathfrak{D}$ by $x \leftarrow \mathfrak{D}$. By $[\![B]\!]$ we denote the bit that is 1 if the boolean Statement B is true, and otherwise 0.

8 By "intermediate", we mean the deterministic scheme that is to be plugged into one of the U-transforms. In most cases, it is derived by starting from a probabilistic scheme and first applying derandomisation transformation T.

9 A strict impossibility result would have to consist of a concrete scheme as well as a concrete attack, with the latter matching the given upper bound.

ALGORITHMS. We denote deterministic computation of an algorithm A on input x by $y := A(x)$. We denote algorithms with access to an oracle O by A^{O}. Unless stated otherwise, we assume all our algorithms to be probabilistic and denote the computation by $y \leftarrow A(x)$.

GAMES. Following [45,11], we use code-based games. We implicitly assume boolean flags to be initialised to false, numerical types to 0, sets to $\varnothing$, and strings to the empty string ϵ. We make the convention that a procedure terminates once it has returned an output.

2.1 Public-key Encryption

SYNTAX. A public-key encryption scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ consists of three algorithms, and a finite message space $\mathcal{M}$ which we assume to be efficiently recognisable. The key generation algorithm KG outputs a key pair (pk, sk), where pk also defines a finite randomness space $\mathcal{R} = \mathcal{R}(pk)$ as well as a ciphertext space $\mathcal{C}$. The encryption algorithm Enc, on input pk and a message $m \in \mathcal{M}$, outputs an encryption $c \leftarrow \mathsf{Enc}(pk, m)$ of m under the public key pk. If necessary, we make the used randomness of encryption explicit by writing $c := \mathsf{Enc}(pk, m; r)$, where $r \leftarrow_{\$} \mathcal{R}$. The decryption algorithm Dec, on input sk and a ciphertext c, outputs either a message $m = \mathsf{Dec}(sk, c) \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$ to indicate that c is not a valid ciphertext.

Definition 1 (Collision probability of key generation.). *We define*

$$\mu(\mathsf{KG}) := \Pr[(pk, sk) \leftarrow \mathsf{KG}, (pk', sk') \leftarrow \mathsf{KG} : pk = pk'] \ .$$

Definition 2 (Collision probability of ciphertexts.). *We define*

$$\mu(\mathsf{Enc}) := \Pr[(pk, sk) \leftarrow \mathsf{KG}, m, m' \leftarrow_{\$} \mathcal{M}, c \leftarrow \mathsf{Enc}(pk, m), c' \leftarrow \mathsf{Enc}(pk, m') : c = c'] \ .$$

Definition 3 (γ-Spreadness.). *[25] We say that* PKE *is γ-spread iff for all key pairs* $(pk, sk) \in \operatorname{supp}(\mathsf{KG})$ *and all messages $m \in \mathcal{M}$ it holds that*

$$\max_{c \in \mathcal{C}} \Pr[r \leftarrow_{\$} \mathcal{R} : \mathsf{Enc}(pk, m; r) = c] \leq 2^{-\gamma} \ .$$

Definition 4 (Correctness). *[28] We define* $\delta := \mathbf{E}[\max_{m \in \mathcal{M}} \Pr[c \leftarrow \mathsf{Enc}(pk, m) : \mathsf{Dec}(sk, c) \neq m]]$*, where the expectation is taken over* $(pk, sk) \leftarrow \mathsf{KG}$*.*

SECURITY. We now define the notion of Indistinguishability under Chosen Plaintext Attacks (IND-CPA) for public-key encryption.

Definition 5 (IND-CPA). *Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme. We define game* IND-CPA *game as in Figure 3, and the* IND-CPA *advantage function of a quantum adversary* $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ *against* PKE *(such that* A_2 *has binary output) as*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}) := |\Pr[\mathsf{IND\text{-}CPA}^{\mathsf{A}}_1 \Rightarrow 1] - \Pr[\mathsf{IND\text{-}CPA}^{\mathsf{A}}_0 \Rightarrow 1]| \ .$$

We also define IND-CPA *security in the random oracle model model, where* PKE *and adversary* A *are given access to a random oracle.*

DISJOINT SIMULATABILITY. Following [43], we consider PKE where it is possible to efficiently sample fake ciphertexts that are indistinguishable from proper encryptions, while the probability that the sampling algorithm hits a proper encryption is small.

GAME $\mathsf{IND\text{-}CPA}_b$	**GAME** $\mathsf{IND\text{-}CCA}$	$\textsc{Decaps}(c \neq c^*)$
$(pk, sk) \leftarrow \mathsf{KG}$	$(pk, sk) \leftarrow \mathsf{KG}$	$K := \mathsf{Decaps}(sk, c)$
$(m_0^*, m_1^*, \mathrm{st}) \leftarrow \mathsf{A}_1(pk)$	$b \leftarrow_\$ \mathbb{F}_2$	**return** K
$c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$	$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
$b' \leftarrow \mathsf{A}_2(pk, c^*, \mathrm{st})$	$K_1^* \leftarrow_\$ \mathcal{K}$	
return b'	$b' \leftarrow \mathsf{A}^{\textsc{Decaps}}(pk, c^*, K_b^*)$	
	return $[\![b' = b]\!]$	

Fig. 3: Games $\mathsf{IND\text{-}CPA}_b$ for PKE ($b \in \mathbb{F}_2$) and game $\mathsf{IND\text{-}CCA}$ for KEM.

Definition 6. *(*DS*) Let* $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$*, coming with an additional PPT algorithm* $\overline{\mathsf{Enc}}$*. For quantum adversaries* A*, we define the advantage against* PKE*'s disjoint simulatability as*

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE},\overline{\mathsf{Enc}}}(\mathsf{A}) := \left| \Pr[pk \leftarrow \mathsf{KG}, m \leftarrow_\$ \mathcal{M}, c \leftarrow \mathsf{Enc}(pk, m) : 1 \leftarrow \mathsf{A}(pk, c)] \right.$$
$$\left. - \Pr[pk \leftarrow \mathsf{KG}, c \leftarrow \overline{\mathsf{Enc}}(pk) : 1 \leftarrow \mathsf{A}(pk, c)] \right| \ .$$

When there is no chance of confusion, we will drop $\overline{\mathsf{Enc}}$ *from the advantage's subscript for convenience.*

We call PKE ϵ_{dis}*-disjoint if for all* $pk \in \mathrm{supp}(\mathsf{KG})$*,* $\Pr[c \leftarrow \overline{\mathsf{Enc}}(pk) : c \in \mathsf{Enc}(pk, \mathcal{M}; \mathcal{R})] \leq \epsilon_{dis}$*.*

2.2 Key Encapsulation

Syntax. A key encapsulation mechanism $\mathsf{KEM} = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$ consists of three algorithms. The key generation algorithm KG outputs a key pair (pk, sk), where pk also defines a finite key space $\mathcal{K}$. The encapsulation algorithm Encaps, on input pk, outputs a tuple (K, c) where c is said to be an encapsulation of the key K which is contained in key space $\mathcal{K}$. The deterministic decapsulation algorithm Decaps, on input sk and an encapsulation c, outputs either a key $K := \mathsf{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.

We call KEM δ*-correct* if

$$\Pr\left[\mathsf{Decaps}(sk, c) \neq K \mid (pk, sk) \leftarrow \mathsf{KG}; (K, c) \leftarrow \mathsf{Encaps}(pk)\right] \leq \delta \ .$$

Note that the above definition also makes sense in the random oracle model since KEM ciphertexts do not depend on messages.

Security. We now define a security notion for key encapsulation: Indistinguishbility under Chosen Ciphertext Attacks ($\mathsf{IND\text{-}CCA}$).

Definition 7 ($\mathsf{IND\text{-}CCA}$**).** *We define the* $\mathsf{IND\text{-}CCA}$ *game as in Figure 3 and the* $\mathsf{IND\text{-}CCA}$ *advantage function of an adversary* A *(with binary output) against* KEM *as*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) := \left| \Pr[\mathsf{IND\text{-}CCA}^{\mathsf{A}} \Rightarrow 1] - 1/2 \right| \ .$$

2.3 Quantum computation

Qubits. For simplicity, we will treat a *qubit* as a vector $|\varphi\rangle \in \mathbb{C}^2$, i.e., a linear combination $|\varphi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ of the two *basis states* (vectors) $|0\rangle$ and $|1\rangle$ with the additional requirement to the probability amplitudes $\alpha, \beta \in \mathbb{C}$ that $|\alpha|^2 + |\beta|^2 = 1$. The basis $\{|0\rangle, |1\rangle\}$ is called *standard orthonormal computational basis*. The qubit $|\varphi\rangle$ is said to be *in superposition*. Classical bits can be interpreted as quantum bits via the mapping $(b \mapsto 1 \cdot |b\rangle + 0 \cdot |1-b\rangle)$.

Quantum Registers. We will treat a quantum register as a collection of multiple qubits, i.e. a linear combination $|\varphi\rangle := \sum_{x \in \mathbb{F}_2^n} \alpha_x \cdot |x\rangle$, where $\alpha_x \in \mathbb{C}$, with the additional restriction that $\sum_{x \in \mathbb{F}_2^n} |\alpha_x|^2 = 1$. As in the one-dimensional case, we call the basis $\{|x\rangle\}_{x \in \mathbb{F}_2^n}$ the *standard orthonormal computational basis*. We say that $|\varphi\rangle = \sum_{x \in \mathbb{F}_2^n} \alpha_x \cdot |x\rangle$ *contains the classical query* x if $\alpha_x \neq 0$.

Measurements. Qubits can be measured with respect to a basis. In this paper, we will only consider measurements in the standard orthonormal computational basis, and denote this measurement by Measure$(\cdot)$, where the outcome of Measure$(|\varphi\rangle)$ for a single qubit $|\varphi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ will be 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$, and the outcome of measuring a qubit register $|\varphi\rangle = \sum_{x \in \mathbb{F}_2^n} \alpha_x \cdot |x\rangle$ will be x with probability $|\alpha_x|^2$. Note that the amplitudes *collapse* during a measurement, this means that by measuring $\alpha \cdot |0\rangle + \beta \cdot |1\rangle$, α and β are switched to one of the combinations in $\{\pm(1,0),\ \pm(0,1)\}$. Likewise, in the n-dimensional case, all amplitudes are switched to 0 except for the one that belongs to the measurement outcome and which will be switched to 1.

Quantum oracles and quantum Adversaries. Following [14,6], we view a quantum oracle $|\mathsf{O}\rangle$ as a mapping

$$|x\rangle|y\rangle \mapsto |x\rangle|y \oplus \mathsf{O}(x)\rangle \ ,$$

where $\mathsf{O} : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, and model quantum adversaries A with access to O by a sequence U_1, $|\mathsf{O}\rangle$, U_2, $\cdots$, $|\mathsf{O}\rangle$, U_N of unitary transformations. We write $\mathsf{A}^{|\mathsf{O}\rangle}$ to indicate that the oracles are quantum-accessible (contrary to oracles which can only process classical bits).

Quantum random oracle model. We consider security games in the quantum random oracle model (QROM) as their counterparts in the classical random oracle model, with the difference that we consider quantum adversaries that are given **quantum** access to the (offline) random oracles involved, and **classical** access to all other (online) oracles. For example, in the IND-CPA game, the adversary only obtains a classical encryption, like in [18], and unlike in [15]. In the IND-CCA game, the adversary only has access to a classical decryption oracle, unlike in [27] and [1].

Zhandry [48] proved that no quantum algorithm $\mathsf{A}^{|\mathsf{O}\rangle}$, issuing at most q quantum queries to $|\mathsf{O}\rangle$, can distinguish between a random function $\mathsf{O} : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ and a $2q$-wise independent function f_{2q}. For concreteness, we view $f_{2q} : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ as a random polynomial of degree $2q$ over the finite field $\mathbb{F}_{2^n}$. The running time to evaluate f_{2q} is linear in q. In this article, we will use this observation in the context of security reductions, where quantum adversary B simulates quantum adversary $\mathsf{A}^{|\mathsf{O}\rangle}$ issuing at most q queries to $|\mathsf{O}\rangle$. Hence, the running time of B is $\text{Time}(\mathsf{B}) = \text{Time}(\mathsf{A}) + q \cdot \text{Time}(\mathsf{O})$, where $\text{Time}(\mathsf{O})$ denotes the time it takes to simulate $|\mathsf{O}\rangle$. Using the observation above, B can use a $2q$-wise independent function in order to (information-theoretically) simulate $|\mathsf{O}\rangle$, and we obtain that the running time of B is $\text{Time}(\mathsf{B}) = \text{Time}(\mathsf{A}) + q \cdot \text{Time}(f_{2q})$, and the time $\text{Time}(f_{2q})$ to evaluate f_{2q} is linear in q. Following [43] and [36], we make use of the fact that the second term of this running time (quadratic in q) can be further reduced to linear in q in the quantum random-oracle model where B can simply use another random oracle to simulate $|\mathsf{O}\rangle$. Assuming evaluating the random oracle takes one time unit, we write $\text{Time}(\mathsf{B}) = \text{Time}(\mathsf{A}) + q$, which is approximately $\text{Time}(\mathsf{A})$.

Oneway to Hiding with semi-classical oracles. In [3], Ambainis et al. defined semi-classical oracles that return a state that was measured with respect to one of the input registers. In particular, to any subset $S \subset X$, they associated the following semi-classical oracle $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$: Algorithm $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$, when queried on $|\psi, 0\rangle$, measures with respect to the projectors M_1 and M_0, where $M_1 := \sum_{x \in S} |x\rangle\langle x|$ and

$M_0 := \sum_{x \notin S} |x\rangle\langle x|$. The oracle then initialises the second register to $|b\rangle$ for the measured bit b. This means that $|\psi, 0\rangle$ collapses to either a state $|\psi', 0\rangle$ such that $|\psi'\rangle$ only contains elements of $X \setminus S$ or to a state $|\psi', 1\rangle$ such that $|\psi'\rangle$ only contains elements of S. Let FIND denote the event that the latter ever is the case, i.e., that $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$ ever answers with $|\psi', 1\rangle$ for some ψ'. To a quantum-accessible oracle G and a subset $S \subset X$, Ambainis et al. associate the following punctured oracle $\mathsf{G} \setminus \mathsf{S}$ that removes S from the domain of G unless FIND occurs.

$\underline{\mathsf{G} \setminus \mathsf{S}|\psi, \phi\rangle}$

$|\psi', b\rangle := \mathsf{O}_\mathsf{S}^{\mathsf{SC}}|\psi, 0\rangle$

return $U_\mathsf{G}|\psi', \phi\rangle$

Fig. 4: Punctured oracle $\mathsf{G} \setminus \mathsf{S}$ for O2H.

The following theorem is a simplification of statement (2) given in [3, Thm. 1: "Semi-classical O2H"], and of [3, Cor. 1]. It differs in the following way: While [3] consider adversaries that might execute parallel oracle invocations and therefore differentiate between query depth d and number of queries q, we use the upper bound $q \geq d$ for simplicity.

Theorem 1. *Let $S \subset X$ be random. Let $G, H \in Y^X$ be random functions such that $G_{|X \setminus S} = H_{|X \setminus S}$, and let z be a random bitstring. (S, G, H, and z may have an arbitrary joint distribution.) Then, for all quantum algorithms A issuing at most q queries that, on input z, output either 0 or 1,*

$$|\Pr[1 \leftarrow \mathsf{A}^{|\mathsf{G}\rangle}(z)] - \Pr[1 \leftarrow \mathsf{A}^{|\mathsf{H}\rangle}(z)]| \leq 2 \cdot \sqrt{q \Pr[b \leftarrow \mathsf{A}^{|\mathsf{G} \setminus \mathsf{S}\rangle}(z) : \mathrm{FIND}]} \ .$$

If furthermore $S := \{x\}$ for $x \leftarrow_\$ X$, and x and z are independent,

$$\Pr[b \leftarrow \mathsf{A}^{|\mathsf{G} \setminus \mathsf{S}\rangle}(z) : \mathrm{FIND}] \leq \frac{4q}{|X|} \ .$$

Generic quantum Distinguishing Problem with bounded probabilities. For $\lambda \in [0, 1]$, let B_λ be the Bernoulli distribution, i.e., $\Pr[b = 1] = \lambda$ for the bit $b \leftarrow B_\lambda$. Let X be some finite set. The generic quantum distinguishing problem ([4, Lemma 37], [30, Lem. 3]) is to distinguish quantum access to an oracle $F : X \to \mathbb{F}_2$, such that for each $x \in X$, $F(x)$ is distributed according to B_λ, from quantum access to the zero function. We will need the following slight variation. The Generic quantum Distinguishing Problem with Bounded probabilities GDPB is like the quantum distinguishing problem with the difference that the Bernoulli parameter λ_x may depend on x, but still is upper bounded by a global λ. The upper bound we give is the same as in [30, Lem. 3]. It is proven in the full version.

Lemma 1 (Generic Distinguishing Problem with Bounded Probabilities). *[Generic Distinguishing Problem with Bounded Probabilities] Let X be a finite set, and let $\lambda \in [0, 1]$. Then, for any (unbounded, quantum) algorithm A issuing at most q quantum queries,*

$$|\Pr[\mathsf{GDPB}_{\lambda,0}^{\mathsf{A}} \Rightarrow 1] - \Pr[\mathsf{GDPB}_{\lambda,1}^{\mathsf{A}} \Rightarrow 1]| \leq 8(q + 1)^2 \cdot \lambda,$$

where games $\mathsf{GDPB}_{\lambda,b}^{\mathsf{A}}$ (for bit $b \in \mathbb{F}_2$) are defined as follows:

GAME $\mathsf{GDPB}_{\lambda,b}$

$(\lambda_x)_{x\in X} \leftarrow \mathsf{A}_1$
if $\exists x \in X$ s.t. $\lambda_x > \lambda$ **return** 0
if $b = 0$
  $F := 0$
else for all $x \in X$
  $F(x) \leftarrow B_{\lambda_x}$
$b' \leftarrow \mathsf{A}_2^{|F\rangle}$
return b'

3 The FO Transformation: QROM security with correctness errors

In Section 3.1, we modularise transformation TPunc that was given in [43] and that turns any public key encryption scheme that is IND-CPA secure into a deterministic one that is DS. Transformation TPunc essentially consists of first puncturing the message space at one point (transformation Punc, to achieve probabilistic DS), and then applying transformation T. Next, in Section 3.2, we show that transformation $\mathsf{U}_m^{\not\perp}$, when applied to T, transforms any encryption scheme that is DS as well as IND-CPA into a KEM that is IND-CCA secure. We believe that many lattice-based schemes fulfill DS in a natural way,[10] but for the sake of completeness, we will show in the full version how transformation Punc can be used to waive the requirement of DS with negligible loss of efficiency.

3.1 Modularisation of TPunc

We modularise transformation TPunc ("Puncturing and Encrypt-with-Hash") that was given in [43], and that turns any IND-CPA secure PKE scheme into a deterministic one that is DS. Note that apart from reencryption, $\mathsf{TPunc}[\mathsf{PKE}_0, \mathsf{G}]$ given in [43] and our modularisation $\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0], \mathsf{G}]$ are equal. We first give transformation Punc that turns any IND-CPA secure scheme into a scheme that is both DS and IND-CPA. We show that transformation T turns any scheme that is DS as well as IND-CPA secure into a deterministic scheme that is DS.

Transformation Punc: From IND-CPA to probabilistic DS security Transformation Punc turns any IND-CPA secure public-key encryption scheme into a DS secure one by puncturing the message space at one message and sampling encryptions of this message as fake encryptions.

THE CONSTRUCTION. To a public-key encryption scheme $\mathsf{PKE}_0 = (\mathsf{KG}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ with message space $\mathcal{M}_0$, we associate $\mathsf{PKE} := \mathsf{Punc}[\mathsf{PKE}_0, \hat{m}] := (\mathsf{KG} := \mathsf{KG}_0, \mathsf{Enc}, \mathsf{Dec} := \mathsf{Dec}_0)$ with message space $\mathcal{M} := \mathcal{M}_0 \setminus \{\hat{m}\}$ for some message $\hat{m} \in \mathcal{M}$. Encryption and fake encryption sampling of PKE are defined in Figure 5. Note that transformation Punc will only be used as a helper transformation to achieve DS, generically. We prove that Punc achieves DS from IND-CPA security in the full version.

Transformation T: From probabilistic to deterministic DS security Transformation T [7] turns any probabilistic public-key encryption scheme into a deterministic one. The transformed

[10] Fake encryptions could be sampled uniformly random. DS would follow from the LWE assumption, and since LWE samples are relatively sparse, uniform sampling should be disjoint.

$\mathsf{Enc}(pk, m \in \mathcal{M})$	$\overline{\mathsf{Enc}}(pk)$
$c \leftarrow \mathsf{Enc}_0(pk, m)$	$c \leftarrow \mathsf{Enc}_0(pk, \hat{m})$
return c	**return** c

Fig. 5: Encryption and fake encryption sampling of $\mathsf{PKE} = \mathsf{Punc}[\mathsf{PKE}_0]$.

scheme is DS, given that PKE is DS as well as $\mathsf{IND\text{-}CPA}$ secure. Our security proof is tighter than the proof given for TPunc (see [43, Theorem 3.3]) due to our use of the semi-classical O2H theorem.

The Construction. Take an encryption scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$. Assume PKE to be additionally endowed with a sampling algorithm $\overline{\mathsf{Enc}}$ (see Definition 6). To PKE and random oracle $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, we associate $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, where the algorithms of $\mathsf{PKE}' = (\mathsf{KG}' := \mathsf{KG}, \mathsf{Enc}', \mathsf{Dec}', \overline{\mathsf{Enc}}' := \overline{\mathsf{Enc}})$ are defined in Figure 6. Note that Enc' deterministically computes the ciphertext as $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$.

$\mathsf{Enc}'(pk, m)$	$\mathsf{Dec}'(sk, c)$
$c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	$m' := \mathsf{Dec}(sk, c)$.
return c	**if** $m' = \bot$ **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
	return $\bot$
	else return m'

Fig. 6: Deterministic encryption scheme $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$.

The following lemma states that combined $\mathsf{IND\text{-}CPA}$ and DS security of PKE imply the DS security of PKE'.

Lemma 2 (DS security of PKE'). *If PKE is ϵ-disjoint, so is PKE'. For all adversaries A issuing at most q_G (quantum) queries to G, there exist an adversary B_IND and an adversary B_DS such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}'}(\mathsf{A}) &\leq \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{B}_\mathsf{DS}) + 2 \cdot \sqrt{q_\mathsf{G} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}_\mathsf{IND}) + \frac{4q_\mathsf{G}^2}{|\mathcal{M}|}} \\ &\leq \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{B}_\mathsf{DS}) + 2 \cdot \sqrt{q_\mathsf{G} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}_\mathsf{IND})} + \frac{4q_\mathsf{G}}{\sqrt{|\mathcal{M}|}} \ ,\end{aligned}$$

and the running time of each adversary is about that of B.

Proof. It is straightforward to prove disjointness since $\mathsf{Enc}'(pk, \mathcal{M})$ is subset of $\mathsf{Enc}(pk, \mathcal{M}; \mathcal{R})$. Let A be a DS adversary against PKE'. Consider the sequence of games given in Figure 7. Per definition,

$$\begin{aligned}\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}'}(\mathsf{A}) &= |\Pr[G_0^\mathsf{A} \Rightarrow 1] - \Pr[G_1^\mathsf{A} \Rightarrow 1]| \\ &\leq |\Pr[G_0^\mathsf{A} \Rightarrow 1] - \Pr[G_3^\mathsf{A} \Rightarrow 1]| + |\Pr[G_1^\mathsf{A} \Rightarrow 1] - \Pr[G_3^\mathsf{A} \Rightarrow 1]| \ .\end{aligned}$$

To upper bound $|\Pr[G_0^\mathsf{A} \Rightarrow 1] - \Pr[G_3^\mathsf{A} \Rightarrow 1]|$, consider adversary B_DS against the disjoint simulatability of the underlying scheme PKE, given in Figure 8. B_DS runs in the time that is required

```
Games G_0-G_3
01 pk ← KG
02 m* ←$ M
03 c* ← \overline{Enc}(pk)                // G_0
04 r* := G(m*)                            // G_1
05 r* ←$ R                                // G_2-G_3
06 c* := Enc(pk, m*; r*)                  // G_1-G_3
07 b' ← A^{|G⟩}(pk, c*)                   // G_0-G_1, G_3
08 b' ← A^{|H⟩}(pk, c*)                   // G_2
09 return b'
```

```
Game G_4-G_5
10 FIND := false
11 pk ← KG
12 m* ←$ M
13 r* ←$ R
14 c* := Enc(pk, m*; r*)                  // G_4
15 c* := Enc(pk, 0; r*)                   // G_5
16 b' ← A^{|G\{m*}⟩}(pk, c*)
17 return FIND
```

```
G \ {m*}|ψ, φ⟩
18 |ψ', b⟩ := O^{SC}_{{m*}}|ψ, 0⟩
19 if b = 1
20     FIND := true
21 return U_G|ψ', φ⟩
```

Fig. 7: Games G_0 - G_5 for the proof of Lemma 2.

to run A and to simulate G for q_{G} queries. Since B_{DS} perfectly simulates game G_0 if run with a fake ciphertext as input, and game G_3 if run with a random encryption $c \leftarrow \mathsf{Enc}(pk, m^*)$,

$$|\Pr[G_0^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]| = \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{DS}}(\mathsf{B}_{\mathsf{DS}}) \ .$$

It remains to upper bound $|\Pr[G_1^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]|$. We claim that there exists an adversary $\mathsf{B}_{\mathsf{IND}}$ such that

$$|\Pr[G_1^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]| \leq 2\sqrt{q_{\mathsf{G}} \cdot \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathsf{B}_{\mathsf{IND}}) + \frac{4q_{\mathsf{G}}^2}{|\mathcal{M}|}} \ .$$

```
B_DS(pk, c)
01 b' ← A^{|G⟩}(pk, c)
02 return b'
```

```
B_IND,1(pk)
03 m* ←$ M
04 return (0, m*, st := m*)

B_IND,2(pk, c*, st := m*)
05 FIND := false
06 b' ← A^{|G\{m*}⟩}(pk, c*)
07 return FIND
```

```
G \ {m*}|ψ, φ⟩
08 |ψ', b⟩ := O^{SC}_{{m*}}|ψ, 0⟩
09 if b = 1
10     FIND := true
11 return U_G|ψ', φ⟩
```

Fig. 8: Adversaries B_{DS} and $\mathsf{B}_{\mathsf{IND}\text{-}}$ for the proof of Lemma 2.

Game G_2. In game G_2, we replace oracle access to G with oracle acess to H in line 08, where H is defined as follows: we pick a uniformly random r^* in line 05 and let $\mathsf{H}(m) := \mathsf{G}(m)$ for all $m \neq m^*$, and $\mathsf{H}(m^*) := r^*$. Note that this change also affects the challenge ciphertext c^* since it is now defined relative to this new r^*, i.e., we now have $c^* = \mathsf{Enc}(pk, m^*; \mathsf{H}(m^*))$. Since r^* is uniformly random and G is a random oracle, so is H, and since we kept c^* consistent, this change is purely conceptual and

$$\Pr[G_1^{\mathsf{A}} \Rightarrow 1] = \Pr[G_2^{\mathsf{A}} \Rightarrow 1] \ .$$

GAME G_3. In game G_3, we switch back to oracle access to G, but keep c^* unaffected by this change. We now are ready to use Oneway to Hiding with semi-classical oracles. Intuitively, the first part of O2H states that if oracles G and H only differ on point m^*, the probability of an adversary being able to tell G and H apart is directly related to m^* being detectable in its random oracle queries. Detecting m^* is formalised by game G_4, in which each of the random oracle queries of A is measured with respect to projector $|m^*\rangle\langle m^*|$, thereby collapsing the query to either m^* (and switching flag FIND to **true**) or a superposition that does not contain m^* at all. Following the notation of [3], we denote this process by a call to oracle $\mathsf{O}^{\mathsf{SC}}_{\{\mathsf{m}^*\}}$, see line 08. Applying the first statement of Theorem 1 for $S := \{m^*\}$, and $z := (pk, c^* := \mathsf{Enc}(pk, m^*; r^*))$, we obtain

$$|\Pr[G_2^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]| \leq 2 \cdot \sqrt{q_{\mathsf{G}} \cdot \Pr[G_4^{\mathsf{A}} \Rightarrow 1]} \ .$$

GAME G_5. In game G_5, $c^* \leftarrow \mathsf{Enc}(pk, m^*)$ is replaced with an encryption of 0. Since in game G_5, (pk, c^*) is independent of m^*, we can apply the second statement of O2H that upper bounds the probability of finding an independent point m^*, relative to the number of queries and the size of the search space $\mathcal{M}$. We obtain

$$\Pr[G_5^{\mathsf{A}} \Rightarrow 1] \leq \frac{4q_{\mathsf{G}}}{|\mathcal{M}|} \ .$$

To upper bound $|\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - \Pr[G_5^{\mathsf{A}} \Rightarrow 1]|$, consider adversary $\mathsf{B}_{\mathsf{IND}}$ against the $\mathsf{IND\text{-}CPA}$ security of PKE, also given in Figure 8. $\mathsf{B}_{\mathsf{IND}}$ runs in the time that is required to run A and to simulate the measured version of oracle G for q_{G} queries. $\mathsf{B}_{\mathsf{IND}}$ perfectly simulates game G_4 if run in game $\mathsf{IND\text{-}CPA}_0$ and game G_5 if run in game $\mathsf{IND\text{-}CPA}_1$, therefore,

$$|\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - \Pr[G_5^{\mathsf{A}} \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}_{\mathsf{IND}}) \ .$$

Collecting the probabilities yields

$$\Pr[G_4^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}_{\mathsf{IND}}) + \frac{4q_{\mathsf{G}}}{|\mathcal{M}|} \ .$$

□

3.2 Transformation $\mathsf{FO}^{\not\perp}_m$ and correctness errors

Transformation SXY [43] got rid of the additional hash (sometimes called key confirmation) that was included in [28]'s quantum transformation $\mathsf{QU}^{\not\perp}_m$. SXY is essentially the (classical) transformation $\mathsf{U}^{\not\perp}_m$ that was also given in [28], and apart from doing without the additional hash, it comes with a tight security reduction in the QROM. SXY differs from the (classical) transformation $\mathsf{U}^{\not\perp}_m$ only in the regard that it reencrypts during decapsulation. (In [28], reencryption is done during decryption of T.)

The security proof given in [43] requires the underlying encryption scheme to be perfectly correct, and it turned out that their analysis cannot be trivially adapted to take possible decryption failures into account in a generic setting. A discussion of this matter is given in the full version. What we show instead is that the combined transformation $\mathsf{FO}^{\not\perp}_m = \mathsf{U}^{\not\perp}_m[\mathsf{T}[-, \mathsf{G}], \mathsf{H}]$ turns any encryption scheme that is DS as well as $\mathsf{IND\text{-}CPA}$ into a KEM that is $\mathsf{IND\text{-}CCA}$ secure in the QROM, even if the underlying encryption scheme comes with a small probability of decryption failure. Our reduction is tighter as the (combined) reduction in [43] due to our tighter security proof for T.

The Construction. To $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, and random oracles $\mathsf{H} : \mathcal{M} \to \mathcal{K}$, $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, and an additional internal random oracle $\mathsf{H_r} : \mathcal{C} \to \mathcal{K}$ that can not be directly accessed, we associate $\mathsf{KEM} = \mathsf{FO}^{\not\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}^{\not\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$, where the algorithms of $\mathsf{KEM} = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$ are given in Figure 9.

```
Encaps(pk)                      Decaps(sk, c)
m ←$ M                          m' := Dec(sk, c)
c := Enc(pk, m; G(m))           if m' = ⊥ or Enc(pk, m'; G(m')) ≠ c
K := H(m)                          return K := H_r(c)
return (K, c)                   else return K := H(m')
```

Fig. 9: Key encapsulation mechanism $\mathsf{KEM} = \mathsf{FO}^{\not\perp}_m[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] = \mathsf{U}^{\not\perp}_m[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$. Oracle $\mathsf{H_r}$ is used to generate random values whenever reencryption fails. This strategy is called implicit reject. Amongst others, it is used in [28], [43], and [32]. For simplicity of the proof, $\mathsf{H_r}$ is modelled as an internal random oracle that cannot be accessed directly. For implementation, it would be sufficient to use a PRF.

Security of KEM. The following theorem (whose proof is essentially the same as in [43] except for the consideration of possible decryption failure) establishes that $\mathsf{IND\text{-}CCA}$ security of KEM reduces to DS and $\mathsf{IND\text{-}CPA}$ security of PKE, in the quantum random oracle model.

Theorem 2 (PKE $\mathsf{DS} + \mathsf{IND\text{-}CPA} \overset{\text{QROM}}{\Rightarrow} \mathsf{KEM}$ $\mathsf{IND\text{-}CCA}$). *Assume* PKE *to be δ-correct, and to come with a fake sampling algorithm* $\overline{\mathsf{Enc}}$ *such that* PKE *is ϵ_{dis}-disjoint. Then, for any (quantum)* $\mathsf{IND\text{-}CCA}$ *adversary* A *issuing at most q_D (classical) queries to the decapsulation oracle* Decaps*, at most q_H quantum queries to* H*, and at most q_G quantum queries to* G*, there exist (quantum) adversaries* $\mathsf{B_{DS}}$ *and* $\mathsf{B_{IND}}$ *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) \leq 8 \cdot (2 \cdot q_\mathsf{G} + q_\mathsf{H} + q_D + 4)^2 \cdot \delta + \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{B_{DS}}) + 2 \cdot \sqrt{(q_\mathsf{G} + q_\mathsf{H}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B_{IND}}) + \frac{4(q_\mathsf{G} + q_\mathsf{H})^2}{|\mathcal{M}|}} + \epsilon_{dis} \ ,$$

and the running time of $\mathsf{B_{DS}}$ *and* $\mathsf{B_{IND}}$ *is about that of* A.

Proof. Let A be an adversary against the $\mathsf{IND\text{-}CCA}$ security of KEM, issuing at most q_D queries to Decaps, at most q_H queries to the quantum random oracle H, and at most q_G queries to the quantum random oracle G. Consider the sequence of games given in Figure 10.

Game G_0. Since game G_0 is the original $\mathsf{IND\text{-}CCA}$ game,

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) = |\Pr[G_0^\mathsf{A} \Rightarrow 1] - 1/2| \ .$$

Game G_1. In game G_1, we enforce that no decryption failure will occur: For fixed (pk, sk) and message $m \in \mathcal{M}$, let

$$\mathcal{R}_{\text{bad}}(pk, sk, m) := \{r \in \mathcal{R} \mid \mathsf{Dec}(sk, \mathsf{Enc}(pk, m; r)) \neq m\}$$

GAMES G_0 - G_6

01 $(pk, sk) \leftarrow \mathsf{KG}$
02 $\mathsf{H_r} \leftarrow_\$ \mathcal{K}^{\mathcal{C}}$
03 $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ // G_0, G_4 - G_6
04 Pick $2q$-wise hash f // G_1 - G_3
05 $\mathsf{G} := \mathsf{G}_{pk,sk}$ // G_1 - G_3
06 $\mathsf{H} \leftarrow_\$ \mathcal{K}^{\mathcal{M}}$ // G_0 - G_1
07 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}}$ // G_2 - G_6
08 $\mathsf{H} := \mathsf{H_q}(\mathsf{Enc}(pk, -; \mathsf{G}(-)))$ // G_2 - G_6
09 $b \leftarrow_\$ \mathbb{F}_2$
10 $m^* \leftarrow \mathcal{M}$
11 $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$ // G_0 - G_4
12 $c^* \leftarrow \overline{\mathsf{Enc}}(pk)$ // G_5 - G_6
13 $K_0^* := \mathsf{H}(m^*)$ // G_0 - G_1
14 $K_0^* := \mathsf{H_q}(c^*)$ // G_2 - G_5
15 $K_0^* \leftarrow_\$ \mathcal{K}$ // G_6
16 $K_1^* \leftarrow_\$ \mathcal{K}$
17 $b' \leftarrow \mathsf{A}^{\text{Decaps}, |\mathsf{H}\rangle, |\mathsf{G}\rangle}(pk, c^*, K_b^*)$
18 **return** $[\![b' = b]\!]$

Decaps($c \neq c^*$) // G_0 - G_2

19 $m' := \mathsf{Dec}(sk, c)$
20 **if** $m' = \bot$
or $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
21 **return** $K := \mathsf{H_r}(c)$
22 **else**
23 **return** $K := \mathsf{H}(m')$ // G_0 - G_1
24 **return** $K := \mathsf{H_q}(c)$ // G_2 - G_6

Decaps($c \neq c^*$) // G_3 - G_6

25 **return** $K := \mathsf{H_q}(c)$

$\mathsf{G}_{pk,sk}(m)$

26 $r := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
27 **return** r

Fig. 10: Games G_0 - G_6 for the proof of Theorem 2. f (lines 04 and 26) is an internal $2q$-wise independent hash function, where $q := q_\mathsf{G} + q_\mathsf{H} + 2 \cdot q_D + 1$, that cannot be accessed by A. $\mathsf{Sample}(Y)$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_\$ Y$. $\mathsf{Sample}(Y; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(Y)$ using explicitly given randomness $f(m)$.

denote the set of “bad” randomness. We replace random oracle G in line 05 with $\mathsf{G}_{pk,sk}$ that only samples from good randomness. Further, define

$$\delta(pk, sk, m) := {}^{|\mathcal{R}_{\text{bad}}(pk,sk,m)|}/_{|\mathcal{R}|} \tag{2}$$

as the fraction of bad randomness, and $\delta(pk, sk) := \max_{m \in \mathcal{M}} \delta(pk, sk, m)$. With this notation, $\delta = \mathbf{E}[\max_{m \in \mathcal{M}} \delta(pk, sk, m)]$, where the expectation is taken over $(pk, sk) \leftarrow \mathsf{KG}$.

To upper bound $|\Pr[G_0^\mathsf{A} = 1] - \Pr[G_1^\mathsf{A} = 1]|$, we construct an (unbounded, quantum) adversary B against the generic distinguishing problem with bounded probabilities GDPB (see Lemma 1) in Figure 11, issuing $q_\mathsf{G} + q_D + 1$ queries to F. B draws a key pair $(pk, sk) \leftarrow \mathsf{KG}$ and computes the parameters $\lambda(m)$ of the generic distinguishing problem as $\lambda(m) := \delta(pk, sk, m)$, which are bounded by $\lambda := \delta(pk, sk)$. To analyze B, we first fix (pk, sk). For each $m \in \mathcal{M}$, by the definition of game $\mathsf{GDPB}_{\lambda,1}$, the random variable $\mathsf{F}(m)$ is bernoulli-distributed according to $B_{\lambda(m)} = B_{\delta(pk,sk,m)}$. By construction, the random variable $\mathsf{G}(m)$ defined in line 28 if $\mathsf{F}(m) = 0$ and in line 30 if $\mathsf{F}(m) = 1$ is uniformly distributed in $\mathcal{R}$. Therefore, G is a (quantum-accessible) random oracle, and $\mathsf{B}^{|\mathsf{F}\rangle}$ perfectly simulates game G_0 if executed in game $\mathsf{GDPB}_{\lambda,1}$. Since $\mathsf{B}^{|\mathsf{F}\rangle}$ also perfectly simulates game G_1 if executed in game $\mathsf{GDPB}_{\lambda,0}$,

$$|\Pr[G_0^\mathsf{A} = 1] - \Pr[G_1^\mathsf{A} = 1]| = |\Pr[\mathsf{GDPB}_{\lambda,1}^\mathsf{B} = 1] - \Pr[\mathsf{GDPB}_{\lambda,0}^\mathsf{B} = 1]| \ ,$$

and according to Lemma 1,

$$|\Pr[\mathsf{GDPB}_{\lambda,1}^\mathsf{B} = 1] - \Pr[\mathsf{GDPB}_{\lambda,0}^\mathsf{B} = 1]| \leq 8 \cdot (q_\mathsf{G} + q_D + 2)^2 \cdot \delta \ .$$

$\mathsf{B}_1 = \mathsf{B}'_1$

01 $(pk, sk) \leftarrow \mathsf{KG}$
02 **for** $m \in \mathcal{M}$
03 $\quad \lambda(m) := \delta(pk, sk, m)$
04 **return** $(\lambda(m))_{m \in \mathcal{M}}$

$\mathsf{B}_2^{|\mathsf{H_r}\rangle, |\mathsf{H}\rangle, |\mathsf{F}\rangle}$

05 Pick $2q$-wise hash f
06 $b \leftarrow_\$ \mathbb{F}_2$
07 $m^* \leftarrow \mathcal{M}$
08 $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$
09 $K_0^* := \mathsf{H}(m^*)$
10 $K_1^* \leftarrow_\$ \mathcal{K}$
11 $b' \leftarrow \mathsf{A}^{\text{Decaps}, |\mathsf{H}\rangle, |\mathsf{G}\rangle}(pk, c^*, K_b^*)$
12 **return** $[\![b' = b]\!]$

$\mathsf{B'}_2^{|\mathsf{H_r}\rangle, |\mathsf{H_q}\rangle, |\mathsf{F}\rangle}$

13 Pick $2q$-wise hash f
14 $\mathsf{H} := \mathsf{H_q}(\mathsf{Enc}(pk, -; \mathsf{G}(-)))$
15 $b \leftarrow_\$ \mathbb{F}_2$
16 $m^* \leftarrow \mathcal{M}$
17 $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$
18 $K_0^* := \mathsf{H_q}(c^*)$
19 $K_1^* \leftarrow_\$ \mathcal{K}$
20 $b' \leftarrow \mathsf{A}^{\text{Decaps}, |\mathsf{H}\rangle, |\mathsf{G}\rangle}(pk, c^*, K_b^*)$
21 **return** $[\![b' = b]\!]$

$\text{Decaps}(c \neq c^*)$ // Adversary B

22 $m' := \mathsf{Dec}'(sk, c)$
23 **if** $m' = \bot$
or $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
24 $\quad$ **return** $K := \mathsf{H_r}(c)$
25 **else return** $K := \mathsf{H}(m')$

$\text{Decaps}(c \neq c^*)$ // Adversary B'

26 **return** $K := \mathsf{H_q}(c)$

$\mathsf{G}(m)$

27 **if** $\mathsf{F}(m) = 0$
28 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
29 **else**
30 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
31 **return** $\mathsf{G}(m)$

Fig. 11: Adversaries B and B' executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ with access to F (and additional oracles $\mathsf{H_r}$ and H or $\mathsf{H_q}$, respectively) for the proof of Theorem 2. Parameters $\delta(pk, sk, m)$ are defined in Equation (2). Function f (lines 28 and 30) is an internal $2q$-wise independent hash function, where $q := q_\mathsf{G} + q_D + 1$ for B, and $q_\mathsf{G} + q_\mathsf{H} + 1$ for B', that cannot be accessed by A.

Game G_2. In game G_2, we prepare getting rid of the secret key by plugging in encryption into random oracle H: Instead of drawing $\mathsf{H} \leftarrow_\$ \mathcal{K}^\mathcal{M}$, we draw $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^\mathcal{C}$ in line 07 and define $\mathsf{H} := \mathsf{H_q}(\mathsf{Enc}(pk, -; \mathsf{G}(-)))$ in line 08. For consistency, we also change key K_0^* in line 14 from letting $K_0^* := \mathsf{H}(m^*)$ to letting $K_0^* := \mathsf{H_q}(c^*)$, which is a purely conceptual change since $c^* = \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$. Additionally, we make the change of H explicit in oracle Decaps, i.e., we change oracle Decaps in line 24 such that it returns $K := \mathsf{H_q}(c)$ whenever $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) = c$. Since G only samples from good randomness, encryption is rendered perfectly correct and hence, injective. Since encryption is injective, H still is uniformly random. Furthermore, since we only change Decaps for ciphertexts c where $c = \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$, we maintain consistency of H and Decaps. In conclusion, A's view is identical in both games and

$$\Pr[G_1^\mathsf{A} = 1] = \Pr[G_2^\mathsf{A} = 1] \ .$$

Game G_3. In game G_3, we change oracle Decaps such that it always returns $K := \mathsf{H_q}(c)$, as opposed to returning $K := \mathsf{H_r}(c)$ as in game G_2 whenever decryption or reencryption fails (see line 21). We argue that this change does not affect A's view: If there exists no message m such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, oracle DECAPS($c$) returns a random value (that can not possibly correlate to any random oracle query to H) in both games, therefore DECAPS(c) is a random value independent of all other input to A in both games. And if there exists some message m such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, DECAPS($c$) would have returned $\mathsf{H_q}(c)$ in both games, anyway: Since $\mathsf{G}(m) \in \mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m)$ for all messages m, it holds that $m' := \mathsf{Dec}(sk, c) = m \neq \bot$ and that $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) = c$. Hence, A's view is identical in both games and

$$\Pr[G_2^{\mathsf{A}} = 1] = \Pr[G_3^{\mathsf{A}} = 1] \ .$$

GAME G_4. In game G_4, we switch back to using $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{pk,sk}$. With the same reasoning as for the gamehop from game G_0 to G_1,

$$\begin{aligned} |\Pr[G_3^{\mathsf{A}} = 1] - \Pr[G_4^{\mathsf{A}} = 1]| &= |\Pr[\mathsf{GDPB}_{\lambda,1}^{\mathsf{B}'} = 1] - \Pr[\mathsf{GDPB}_{\lambda,0}^{\mathsf{B}'} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 2)^2 \cdot \delta \ , \end{aligned}$$

where adversary B' (that issues at most issuing $q_{\mathsf{G}} + q_{\mathsf{H}} + 1$ queries to F) is also given in Figure 11.

So far, we established

$$\mathrm{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \leq |\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - 1/2| + 8 \cdot (2 \cdot q_{\mathsf{G}} + q_{\mathsf{H}} + q_D + 4)^2 \cdot \delta \ .$$

The rest of the proof proceeds similiar to the proof in [43], aside from the fact that we consider the particular scheme $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ instead of a generic encryption scheme that is deterministically DS.

GAME G_5. In game G_5, the challenge ciphertext c^* gets decoupled from message m^* by sampling $c^* \leftarrow \overline{\mathsf{Enc}}(pk)$ in line 12 instead of letting $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$. Consider the adversary $\mathsf{C_{DS}}$ against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ given in Figure 12. Since $\mathsf{C_{DS}}$ perfectly simulates game G_4 if run with deterministic encryption $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$ of a random message m^*, and game G_5 if run with a fake ciphertext,

$$|\Pr[G_4^{\mathsf{A}} = 1] - \Pr[G_5^{\mathsf{A}} = 1]| = \mathrm{Adv}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}^{\mathsf{DS}}(\mathsf{C_{DS}}), \ ,$$

and according to Lemma 2, there exist an adversary $\mathsf{B_{DS}}$ and an adversary $\mathsf{B_{IND}}$ with roughly the same running time such that

$$\mathrm{Adv}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}^{\mathsf{DS}}(\mathsf{C_{DS}}) \leq \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{DS}}(\mathsf{B_{DS}}) + 2 \cdot \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathsf{B_{IND}}) + \frac{4(q_{\mathsf{G}} + q_{\mathsf{H}})^2}{|\mathcal{M}|}} \ .$$

GAME G_6. In game G_6, the game is changed in line 15 such that it always uses a randomly picked challenge key. Since both K_0^* and K_1^* are independent of all other input to A in game G_6,

$$\Pr[G_6^{\mathsf{A}} \Rightarrow 1] = 1/2 \ .$$

It remains to upper bound $|\Pr[G_5^{\mathsf{A}} = 1] - \Pr[G_6^{\mathsf{A}} = 1]|$. To this end, it is sufficient to upper bound the probability that any of the queries to $\mathsf{H_q}$ could possibly contain c^*. Each query to $\mathsf{H_q}$ is either a classical query, triggered by A querying DECAPS on some ciphertext c, or a query in superposition, triggered by A querying H. Since queries to DECAPS on c^* are explicitly forbidden, the only possibility would be one of A's queries to H. A's queries to H trigger queries to $\mathsf{H_q}$ that are

$\underline{\mathsf{C_{DS}}^{|\mathsf{G}\rangle,|\mathsf{H_r}\rangle|\mathsf{H_q}\rangle}(pk, c^*)}$

$b \leftarrow_\$ \mathbb{F}_2$

$K_0^* := \mathsf{H_q}(c^*)$

$K_1^* \leftarrow_\$ \mathcal{K}$

$b' \leftarrow \mathsf{A}^{\textsc{Decaps},|\mathsf{H}\rangle,|\mathsf{G}\rangle}(pk, c^*, K_b^*)$

return $[\![b' = b]\!]$

$\underline{\textsc{Decaps}(c \neq c^*)}$

return $K := \mathsf{H_q}(c)$

Fig. 12: Adversary $\mathsf{C_{DS}}$ (with access to additional oracles $\mathsf{H_r}$ and $\mathsf{H_q}$) against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ for the proof of Theorem 2.

of the form $\sum_m \alpha_m|\mathsf{Enc}(pk, m; \mathsf{G}(m))\rangle$. They cannot contain c^* unless there exists some message m such that $\mathsf{Enc}(pk, m; \mathsf{G}(m)) = c^*$. Since we assume PKE to be ϵ_{dis}-disjoint,

$$|\Pr[G_5^{\mathsf{A}} = 1] - \Pr[G_6^{\mathsf{A}} = 1]| \leq \epsilon_{\text{dis}} \ .$$

3.3 CCA security wihout disjoint simulatability.

In the full version we show that transformation Punc can be used to waive the requirement of DS: Plugging in transformation Punc (before using $\mathsf{FO}_m^{\not\perp}$) achieves IND-CCA security from IND-CPA security alone, as long as PKE is γ-spread (see Definition 3).

4 Two-Message Authenticated Key Exchange

A two-message key exchange protocol $\mathsf{AKE} = (\mathsf{KG}, \mathsf{Init}, \mathsf{Der_{init}}, \mathsf{Der_{resp}})$ consists of four algorithms. Given the security parameter, the key generation algorithm KG outputs a key pair (pk, sk). The initialisation algorithm Init, on input sk and pk', outputs a message M and a state st. The responder's derivation algorithm $\mathsf{Der_{resp}}$, on input sk', pk and M, outputs a key K, and also a message M'. The initiator's derivation algorithm $\mathsf{Der_{init}}$, on input sk, pk', M' and st, outputs a key K.

Running a Key Exchange Protocol between two Parties. To run a two-message key exchange protocol, the algorithms $\mathsf{KG}, \mathsf{Init}$, $\mathsf{Der_{init}}$, and $\mathsf{Der_{resp}}$ are executed in an interactive manner between two parties P_i and P_j with key pairs $(sk_i, pk_i), (sk_j, pk_j) \leftarrow \mathsf{KG}$. To execute the protocol, the parties call the algorithms in the following way:

1. P_i computes $(M, \text{st}) \leftarrow \mathsf{Init}(sk_i, pk_j)$ and sends M to P_j.
2. P_j computes $(M', K') \leftarrow \mathsf{Der_{resp}}(sk_j, pk_i, M)$ and sends M' to P_i.
3. P_i computes $K := \mathsf{Der_{init}}(sk_i, pk_j, M', \text{st})$.

Note that in contrast to the holder P_i, the peer P_j will not be required to save any (secret) state information besides the key K'.

Our Security Model. We consider N parties $\mathsf{P}_1, \ldots, \mathsf{P}_N$, each holding a key pair (sk_i, pk_i), and possibly having several sessions at once. The sessions run the protocol with access to the party's long-term key material, while also having their own set of (session-specific) local variables. The local variables of each session, identified by the integer sID, are the following:

Party P_i (pk_i, sk_i)		Party P_j (pk_j, sk_j)
$(M, \mathrm{st}) \leftarrow \mathsf{Init}(sk_i, pk_j)$	$\xrightarrow{\quad M \quad}$	
		$(M', K') \leftarrow \mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M)$
$K := \mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \mathrm{st})$	$\xleftarrow[\quad M' \quad]{}$	

1. An integer **holder** $\in [N]$ that points to the party running the session.
2. An integer **peer** $\in [N]$ that points to the party the session is communicating with.
3. A string **sent** that holds the message sent by the session.
4. A string **received** that holds the message received by the session.
5. A string **st** that holds (secret) internal state values and intermediary results required by the session.
6. A string **role** that holds the information whether the session's key was derived by $\mathsf{Der}_{\mathsf{init}}$ or $\mathsf{Der}_{\mathsf{resp}}$.
7. The session key K.

In our security model, the adversary A is given black-box access to the set of processes Init, $\mathsf{Der}_{\mathsf{resp}}$ and $\mathsf{Der}_{\mathsf{init}}$ that execute the AKE algorithms. To model the attacker's control of the network, we allow A to establish new sessions via EST, to call either INIT and $\mathrm{DER}_{\mathrm{init}}$ or $\mathrm{DER}_{\mathrm{resp}}$, each at most once per session (see Figure 13, page 23). Since both derivation processes can be called on arbitrary input, A may relay their input faithfully as well as modify the data on transit. Moreover, the attacker is additionally granted queries to reveal both secret process data, namely using oracles REVEAL, REV-STATE and CORRUPT (see Figure 14, page 24). Oracles REVEAL and REV-STATE both can be queried on an arbitrary session ID, with oracle REVEAL revealing the respective session's key (if already defined), and oracle REV-STATE revealing the respective session's internal state. Oracle CORRUPT can be queried on an arbitrary number $i \in [N]$ to reveal the respective party's long-term key material. Usage of this oracle allows the attacker to corrupt the test session's holder, the oracle therefore models the possibility of KCI attacks. Combined usage of oracles REV-STATE and CORRUPT allows the attacker to obtain the state as well as the long-term secret key on both sides of the session, the oracles therefore model the possibility of MEX attacks. After choosing a test session, either the session's key or a uniformly random key is returned. The attacker's task is to distinguish these two cases, to this end it outputs a bit.

Definition 8 (Key Indistinguishability of AKE).
We define games $\mathsf{IND\text{-}AA}_b$ *and* $\mathsf{IND\text{-}StAA}_b$ *for* $b \in \mathbb{F}_2$ *as in Figure 13 and Figure 14.*
We define the $\mathsf{IND\text{-}AA}$ *advantage function of an adversary* A *against* AKE *as*

$$\mathrm{Adv}^{\mathsf{IND\text{-}AA}}_{\mathsf{AKE}}(\mathsf{A}) := |\Pr[\mathsf{IND\text{-}AA}_1^{\mathsf{A}} \Rightarrow 1] - \Pr[\mathsf{IND\text{-}AA}_0^{\mathsf{A}} \Rightarrow 1]| \ ,$$

and the $\mathsf{IND\text{-}StAA}$ *advantage function of an adversary* A *against* AKE *excluding test-state-attacks as*

$$\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{A}) := |\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{A}} \Rightarrow 1] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{A}} \Rightarrow 1]| \ .$$

We call a session *completed* iff $\mathrm{sKey}[\mathrm{sID}] \neq \perp$, which implies that either $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}, m)$ or $\mathrm{DER}_{\mathrm{init}}(\mathrm{sID}, m)$ was queried for some message m. We say that a completed session sID *was recreated*

```
GAME IND-AA_b
01 cnt := 0                               // session counter
02 sID* := 0                              // test session's id
03 for n ∈ [N]
04    (pk_n, sk_n) ← KG
05 b' ← A^O(pk_1, ··· , pk_N)
06 if Trivial(sID*)
07    return 0
08 return b'

EST((i, j) ∈ [N]^2)
09 cnt ++
10 holder[cnt] := i
11 peer[cnt] := j
12 return cnt

DER_resp(sID, M)
13 if holder[sID] = ⊥
14    return ⊥                            // Session not established
15 if sKey[sID] ≠ ⊥ return ⊥              // no re-use
16 if role[sID] = "initiator" return ⊥
17 role[sID] := "responder"
18 (j, i) := (holder[sID], peer[sID])
19 (M', K') ← Der_resp(sk_j, pk_i, M)
20 sKey[sID] := K'
21 (received[sID], sent[sID]) := (M, M')
22 return M'
```

```
GAME IND-StAA_b
23 cnt := 0                               // session counter
24 sID* := 0                              // test session's id
25 for n ∈ [N]
26    (pk_n, sk_n) ← KG
27 b' ← A^O(pk_1, ··· , pk_N)
28 if ATTACK(sID*)
29    return 0
30 return b'

INIT(sID)
31 if holder[sID] = ⊥
32    return ⊥                            // Session not established
33 if sent[sID] ≠ ⊥ return ⊥              // no re-use
34 role[sID] := "initiator"
35 (i, j) := (holder[sID], peer[sID])
36 (M, st) ← Init(sk_i, pk_j)
37 (sent[sID], state[sID]) := (M, st)
38 return M

DER_init(sID, M')
39 if holder[sID] = ⊥ or state[sID] = ⊥
40    return ⊥                            // Session not initalised
41 if sKey[sID] ≠ ⊥ return ⊥              // no re-use
42 (i, j) := (holder[sID], peer[sID])
43 st := state[sID]
44 sKey[sID] := Der_init(sk_i, pk_j, M', st)
45 received[sID] := M'
```

Fig. 13: Games $\mathsf{IND\text{-}AA}_b$ and $\mathsf{IND\text{-}StAA}_b$ for AKE, where $b \in \mathbb{F}_2$. The collection of oracles O used in lines 05 and 27 is defined by O := {EST, INIT, $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$, REVEAL, REV-STATE, CORRUPT, TEST}. Oracles REVEAL, REV-STATE, CORRUPT, and TEST are given in Figure 14. Game $\mathsf{IND\text{-}StAA}_b$ only differs from $\mathsf{IND\text{-}AA}_b$ in ruling out one more kind of attack: A's bit b' does not count in games $\mathsf{IND\text{-}AA}_b$ if helper procedure Trivial returns **true**, see line 06. In games $\mathsf{IND\text{-}StAA}_b$, A's bit b' does not count already if procedure ATTACK (that includes Trivial and additionally checks for state-attacks on the test session) returns **true**, see line 28.

iff there exists a session $\mathrm{sID}' \neq \mathrm{sID}$ such that $(\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}]) = (\mathrm{holder}[\mathrm{sID}'], \mathrm{peer}[\mathrm{sID}'])$, $\mathrm{role}[\mathrm{sID}] = \mathrm{role}[\mathrm{sID}']$, $\mathrm{sent}[\mathrm{sID}] = \mathrm{sent}[\mathrm{sID}']$, $\mathrm{received}[\mathrm{sID}] = \mathrm{received}[\mathrm{sID}']$ and $\mathrm{state}[\mathrm{sID}] = \mathrm{state}[\mathrm{sID}']$. We say that two completed sessions sID_1 and sID_2 *match* iff $(\mathrm{holder}[\mathrm{sID}_1], \mathrm{peer}[\mathrm{sID}_1]) = (\mathrm{peer}[\mathrm{sID}_2], \mathrm{holder}[\mathrm{sID}_2])$, $(\mathrm{sent}[\mathrm{sID}_1], \mathrm{received}[\mathrm{sID}_1]) = (\mathrm{received}[\mathrm{sID}_2], \mathrm{sent}[\mathrm{sID}_2])$, and $\mathrm{role}[\mathrm{sID}_1] \neq \mathrm{role}[\mathrm{sID}_2]$. We say that A *tampered with the test session* sID^* if at the end of the security game, there exists no matching session for sID^* Nonexistence of a matching session implies that A must have called the derivation process on a message of its own choosing.

Helper procedure Trivial (Figure 14) is used in all games to exclude the possibility of trivial attacks, and helper procedure ATTACK (also Figure 14) is defined in games $\mathsf{IND\text{-}StAA}_b$ to exclude the possibility of trivial attacks as well as one nontrivial attack that we will discuss below. During execution of Trivial, the game creates list $\mathfrak{M}(\mathrm{sID}^*)$ of all matching sessions that were executed

```
Trivial(sID*)                               // helper procedure to exclude trivial attacks
46 if sKey[sID*] = ⊥ return true                       // test session was never completed
47 v := false
48 (i, j) := (holder[sID*], peer[sID*])
49 if revealed[sID*] return true               // A trivially learned the test session's key
50 if corrupted[i] and revState[sID*]
51    return true       // A may simply compute Der(sk_i, pk_j, received[sID*], state[sID*])
52 𝔐(sID*) := ∅                                     // create list of matching sessions
53 for 1 ≤ ptr ≤ cnt
54    if (sent[ptr], received[ptr]) = (received[sID*], sent[sID*])
        and (holder[ptr], peer[ptr]) = (j, i) and role[ptr] ≠ role[sID*]
55      𝔐(sID*) := 𝔐(sID*) ∪ {ptr}                                    // session matches
56      if revealed[ptr] v := true   // A trivially learned the test session's key via matching session
57      if corrupted[j] and revState[ptr]
58         v := true           // A may simply compute Der(sk_j, pk_i, received[ptr], state[ptr])
59 if |𝔐(sID*)| > 1 return false   // reward for adversary - protocol was not appropr. random.
60 if v = true return true
61 if 𝔐(sID*) = ∅ and corrupted[j] return true   // A tampered with test session, knowing sk_j
62 return false

ATTACK(sID*)        // helper procedure to exclude trivial attacks as well as state-attacks
63 if Trivial(sID*) return true                                           // trivial attack
64 if 𝔐(sID*) = ∅ and revState[sID*] return true                          // state-attack
65 return false

REVEAL(sID)
66 if sKey[sID] = ⊥ return ⊥
67 revealed[sID] := true
68 return sKey[sID]

CORRUPT(i ∈ [N])
69 if corrupted[i] return ⊥
70 corrupted[i] := true
71 return sk_i

REV-STATE(sID)
72 if state[sID] = ⊥ return ⊥
73 revState[sID] := true
74 return state[sID]

TEST(sID)    // only one query
75 sID* := sID
76 if sKey[sID*] = ⊥
77    return ⊥
78 K*_0 := sKey[sID*]
79 K*_1 ←$ 𝒦
80 return K*_b
```

Fig. 14: Helper procedures Trivial and ATTACK and oracles REVEAL, REV-STATE, CORRUPT, and TEST of games IND-AA and IND-StAA defined in Figure 13.

throughout the game (see line 55), and A's output bit b' counts in games IND-AA_b only if Trivial returns false, i.e., if test session sID^* was completed and all of the following conditions hold:

1. A did not obtain the key of sID^* by querying REVEAL on sID^* or any matching session, see lines 49 and 56.
2. A did not obtain both the holder i's secret key sk_i and the test session's internal state, see line 51. We enforce that $\neg\mathrm{corrupted}[i]$ or $\neg\mathrm{revState}[\mathrm{sID}^*]$ since otherwise, A is allowed to obtain all information required to trivially compute $\mathsf{Der}(sk_i, pk_j, \mathrm{received}[\mathrm{sID}^*], \mathrm{state}[\mathrm{sID}^*])$.
3. A did not obtain both the peer's secret key sk_j and the internal state of any matching session, see line 58. We enforce that $\neg\mathrm{corrupted}[j]$ or $\neg\mathrm{revState}[\mathrm{sID}]$ for all sID s. th. $\mathrm{sID} \in \mathfrak{M}(\mathrm{sID}^*)$ for the same reason as discussed in 2: A could trivially compute $\mathsf{Der}(sk_j, pk_i, \mathrm{received}[\mathrm{sID}], \mathrm{state}[\mathrm{sID}])$ for some matching session sID.

4. A did not both tamper with the test session and obtain the peer j's secret key sk_j, see line 61. We enforce that $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ or $\neg\text{corrupted}[j]$ to exclude the following trivial attack: A could learn the peer's secret key sk_j via query CORRUPT[j] and either
   - receive a message M by querying INIT on sID*, compute $(M', K') \leftarrow \mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M)$ without having to call DER_{resp}, and then call $\text{DER}_{\text{init}}(\text{sID}^*, M')$, thereby ensuring that $\text{sKey}[\text{sID}^*] = K'$,
   - or compute $(M, \text{st}) \leftarrow \mathsf{Init}(sk_j, pk_i)$ without having to call INIT, receive a message M' by querying $\text{DER}_{\text{resp}}(\text{sID}^*, M)$, and trivially compute $\mathsf{Der}_{\mathsf{init}}(sk_j, pk_i, M', \text{st})$.

A's output bit b' only counts in games $\mathsf{IND\text{-}StAA}_b$ if ATTACK returns false, i.e., if both of the following conditions hold:

1. Trivial returns **false**
2. A did not both tamper with the test session and obtain its internal state, see line 64. We enforce that $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ or $\neg\text{revState}[\text{sID}^*]$ in game IND-StAA for the following reason: In an active attack, given that the test session's internal state got leaked, it is possible for some protocols to choose a message M' such that the result of algorithm $\mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \text{st})$ can be computed without knowledge of any of the long-term keys sk_i or sk_j. In this setting, an adversary might query INIT on sID*, learn the internal state st by querying REV-STATE on sID*, choose its own message M' without a call to DER_{resp} and finally call $\text{DER}_{\text{init}}(\text{sID}^*, M')$, thereby being enabled to anticipate the resulting key.

5 Transformation from PKE to AKE

Transformation $\mathsf{FO}_{\mathsf{AKE}}$ constructs a IND-StAA-secure AKE protocol from a PKE scheme that is both DS and IND-CPA secure. If we plug in transformation Punc before applying $\mathsf{FO}_{\mathsf{AKE}}$, we achieve IND-StAA-security from CPA security alone.

The Construction. To a PKE scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$, and random oracles G and H, we associate

$$\mathsf{AKE} = \mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] = (\mathsf{KG}, \mathsf{Init}, \mathsf{Der}_{\mathsf{resp}}, \mathsf{Der}_{\mathsf{init}}) \ .$$

The algorithms of AKE are defined in Figure 15.

IND-StAA Security of $\mathsf{FO}_{\mathsf{AKE}}$. The following theorem establishes that IND-StAA security of AKE reduces to DS and IND-CPA security of PKE (see Definition 6).

Theorem 3 (PKE DS + IND-CPA $\Rightarrow$ AKE IND-StAA)**.** *Assume* PKE *to be δ-correct, and to come with a sampling algorithm $\overline{\mathsf{Enc}}$ such that it is ϵ-disjoint. Then, for any* IND-StAA *adversary* B *that establishes S sessions and issues at most q_{R} (classical) queries to* REVEAL*, at most q_{G} (quantum) queries to random oracle* G *and at most q_{H} (quantum) queries to random oracle* H*, there exists an adversary A_{DS} against the disjoint simulatability of* T[PKE, G] *issuing at most $q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S$ queries to* G *such that*

$$\begin{aligned}\text{Adv}_{\mathsf{AKE}}^{\mathsf{IND\text{-}StAA}}(\mathsf{B}) \leq\ & 2 \cdot S \cdot (S + 3 \cdot N) \cdot \text{Adv}_{\mathsf{T[PKE,G]}}^{\mathsf{DS}}(\mathsf{A}_{\mathsf{DS}}) + 32 \cdot (S + 3 \cdot N) \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\ & + 4 \cdot S \cdot (S + N) \cdot \epsilon_{dis} + S^2 \cdot (N+1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + 2 \cdot S^2 \cdot \mu(\mathsf{KG}) \ ,\end{aligned}$$

$\mathsf{Init}(sk_i, pk_j)$:

$m_j \leftarrow_\$ \mathcal{M}$
$c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
$(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{KG}$
$M := (\tilde{pk}, c_j)$
$\mathrm{st} := (\tilde{sk}, m_j, M)$
return (M, st)

$\mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M)$:

Parse $(\tilde{pk}, c_j) := M$
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m'_j := \mathsf{Dec}(sk_j, c_j)$
if $m'_j = \perp$
  or $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
    $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, i, j, M, M')$
else
    $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
return (M', K')

$\mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \mathrm{st})$:

Parse $(c_i, \tilde{c}) := M'$
Parse $(\tilde{sk}, m_j, M := (\tilde{pk}, c_j)) := \mathrm{st}$
$m'_i := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
if $m'_i = \perp$
  or $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
  if $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$
  else
    $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', i, j, M, M')$
else if $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, i, j, M, M')$
else $K := \mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
return K

Fig. 15: IND-StAA secure AKE protocol $\mathsf{AKE} = \mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. Oracles H'_{R} and $\mathsf{H}'_{\mathsf{L1}}$, $\mathsf{H}'_{\mathsf{L2}}$ and $\mathsf{H}'_{\mathsf{L3}}$ are used to generate random values whenever reencryption fails. (For encryption, this strategy is called *implicit reject* Amongst others, it is used in [28], [43] and [32].) For simplicity of the proof, H'_{R} and $\mathsf{H}'_{\mathsf{L1}}$, $\mathsf{H}'_{\mathsf{L2}}$ and $\mathsf{H}'_{\mathsf{L3}}$ are internal random oracles that cannot be accessed directly. For implementation, it would be sufficient to use a PRF.

and the running time of A_{DS} *is about that of* B. *Due to Lemma 2, there exist adversaries* C_{DS} *and* $\mathsf{C}_{\mathsf{IND}}$ *against* PKE *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{B}) \leq{} & 2 \cdot S \cdot (S + 3 \cdot N) \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{C}_{\mathsf{DS}}) \\ & + 4 \cdot S \cdot (S + 3 \cdot N) \cdot \sqrt{(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C}_{\mathsf{IND}}) + \frac{4(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2}{|\mathcal{M}|}} \\ & + 32 \cdot (S + 3 \cdot N) \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta + 4 \cdot S \cdot (S + N) \cdot \epsilon_{dis} \\ & + S^2 \cdot (N + 1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + 2 \cdot S^2 \cdot \mu(\mathsf{KG}) \enspace ,\end{aligned}$$

and the running times of C_{DS} *and* $\mathsf{C}_{\mathsf{IND}}$ *is about that of* B.

Proof Sketch. To prove IND-StAA security of $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$, we consider an adversary B with black-box access to the protocols' algorithms and to oracles that reveal keys of completed sessions, internal states, and long-term secret keys of participating parties as specified in game IND-StAA (see Figure 13). Intuitively, B will always be able to obtain all-but-one of the three secret messages m_i, m_j and $\tilde{m}$ that are picked during execution of the test session between P_i and P_j:

1. We first consider the case that B executed the test session honestly. Note that on the right-hand side of the protocol there exists no state. We assume that B has learned the secret key of party P_j and hence knows m_j. Additionally, B could either learn the secret key of party P_i and thereby, compute m_i, or the state on the left-hand side of the protocol including $\tilde{sk}$, and thereby, compute $\tilde{m}$, but not both.
2. In the case that B did not execute the test session honestly, B is not only forbidden to obtain the long-term secret key of the test session's peer, but also to obtain the test session's state

due to our restriction in game IND-StAA. Given that B modified the exchanged messages, the test session's side is decoupled from the other side. If the test session is on the right-hand side, messages m_j and $\tilde{m}$ can be obtained, but message m_i can not because we forbid to learn peer i's secret key. If the test session is on the left-hand side, messages m_i and $\tilde{m}$ can be obtained, but message m_j can not because we forbid both to learn the test session's state and to learn peer j's secret key.

In every possible scenario of game IND-StAA, at least one message can not be obtained trivially and is still protected by PKE's IND-CPA security, and the respective ciphertext can be replaced with fake encryptions due to PKE's disjoint simulatability. Consequently, the session key K is pseudorandom. A detailed, game-based proof is given in the full version.

So far we have ignored the fact that B has access to an oracle that reveals the keys of completed sessions. This implicitly provides B a decryption oracle with respect to the secret keys sk_i and sk_j. In our proof, we want to make use of the technique from [43] to simulate the decryption oracles by patching encryption into the random oracle H. In order to extend their technique to PKE schemes with non-perfect correctness, during the security proof we also need to patch random oracle G in a way that $(\mathsf{Enc}', \mathsf{Dec}')$ (relative to the patched G) provides perfect correctness. This strategy is the AKE analogue to the technique used in our analysis of the Fujisaki-Okamoto transformation given in Section 3, in particular, during our proof of Theorem 2. The latter also explains why our transformation does not work with any deterministic encryption scheme, but only with the ones that are derived by using transformation T. For more details on this issue, we also refer to the full version.

5.1 IND-StAA security wihout disjoint simulatability

In the full version we show that transformation Punc can be used to waive the requirement of DS: Plugging in transformation Punc before using $\mathsf{FO}_{\mathsf{AKE}}$ achieves IND-StAA security from IND-CPA security alone, as long as PKE is γ-spread.

Acknowledgments

We would like to thank the anonymous reviewers of Eurocrypt 2018, Crypto 2019 and Asiacrypt 2019 for their helpful comments and suggestions. This work was supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC 2092 CASA, 390781972), ERC Project ERCC (FP7/615074), the German Federal Ministry of Education and Research (BMBF) Project DigiSeal (16KIS0695), the United States Air Force Office of Scientific Research (AFOSR) via AOARD Grant "Verification of Quantum Cryptography" (FA2386-17-1-4022), the project "Research and preparation of an ERC grant application on Certified Quantum Security" (MOBERC12), the ERC Consolidator grant "Certified Quantum Security" (819317), the Estonian Centre of Exellence in IT (EXCITE) funded by the ERDF, and by the institutional research funding IUT2-1 of the Estonian Ministry of Education and Research.

References

1. Alagic, G., Jeffery, S., Ozols, M., Poremba, A.: On non-adaptive quantum chosen-ciphertext attacks and learning with errors. CoRR abs/1808.09655 (2018) 11

2. Alawatugoda, J., Boyd, C., Stebila, D.: Continuous after-the-fact leakage-resilient key exchange. In: Susilo, W., Mu, Y. (eds.) ACISP 14: 19th Australasian Conference on Information Security and Privacy. Lecture Notes in Computer Science, vol. 8544, pp. 258–273. Springer, Heidelberg, Germany, Wollongong, NSW, Australia (Jul 7–9, 2014) 1
3. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. Cryptology ePrint Archive, Report 2018/904 (2018), http://eprint.iacr.org/2018/904 5, 7, 11, 12, 16
4. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th Annual Symposium on Foundations of Computer Science. pp. 474–483. IEEE Computer Society Press, Philadelphia, PA, USA (Oct 18–21, 2014) 12
5. Banik, S., Isobe, T.: Some cryptanalytic results on lizard. Cryptology ePrint Archive, Report 2017/346 (2017), http://eprint.iacr.org/2017/346 7
6. Beals, R., Buhrman, H., Cleve, R., Mosca, M., Wolf, R.: Quantum lower bounds by polynomials. In: 39th Annual Symposium on Foundations of Computer Science. pp. 352–361. IEEE Computer Society Press, Palo Alto, CA, USA (Nov 8–11, 1998) 11
7. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 535–552. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007) 13
8. Bellare, M., Canetti, R., Krawczyk, H.: A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In: 30th Annual ACM Symposium on Theory of Computing. pp. 419–428. ACM Press, Dallas, TX, USA (May 23–26, 1998) 1
9. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93: 1st Conference on Computer and Communications Security. pp. 62–73. ACM Press, Fairfax, Virginia, USA (Nov 3–5, 1993) 3
10. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) Advances in Cryptology – CRYPTO'93. Lecture Notes in Computer Science, vol. 773, pp. 232–249. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 22–26, 1994) 1, 5
11. Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006) 9
12. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: Ntru prime. Cryptology ePrint Archive, Report 2016/461 (2016), http://eprint.iacr.org/2016/461 2, 8
13. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of cca security in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/590 (2019), https://eprint.iacr.org/2019/590 7, 8
14. 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.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011) 2, 11
15. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013, Part II. Lecture Notes in Computer Science, vol. 8043, pp. 361–379. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013) 11
16. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634 (2017), http://eprint.iacr.org/2017/634 3, 7
17. Boyd, C., Cliff, Y., Nieto, J.G., Paterson, K.G.: Efficient one-roundkey exchange in the standard model. ACISP 08: 13th Australasian Conference on Information Security and Privacy (2008) 1, 2, 5
18. Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: Gennaro, R., Robshaw, M.J.B. (eds.) Advances in Cryptology – CRYPTO 2015, Part II. Lecture Notes

in Computer Science, vol. 9216, pp. 609–629. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015) 11
19. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) Advances in Cryptology – EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 453–474. Springer, Heidelberg, Germany, Innsbruck, Austria (May 6–10, 2001) 1
20. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing 33(1), 167–226 (2003) 4
21. Dent, A.W.: A designer's guide to KEMs. In: Paterson, K.G. (ed.) 9th IMA International Conference on Cryptography and Coding. Lecture Notes in Computer Science, vol. 2898, pp. 133–151. Springer, Heidelberg, Germany, Cirencester, UK (Dec 16–18, 2003) 4
22. Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology – EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 342–360. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004) 2
23. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography. Lecture Notes in Computer Science, vol. 7293, pp. 467–484. Springer, Heidelberg, Germany, Darmstadt, Germany (May 21–23, 2012) 1, 2, 6
24. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In: Chen, K., Xie, Q., Qiu, W., Li, N., Tzeng, W.G. (eds.) ASIACCS 13: 8th ACM Symposium on Information, Computer and Communications Security. pp. 83–94. ACM Press, Hangzhou, China (May 8–10, 2013) 1, 3
25. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) Advances in Cryptology – CRYPTO'99. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999) 3, 9
26. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013) 3
27. Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology – CRYPTO 2016, Part III. Lecture Notes in Computer Science, vol. 9816, pp. 60–89. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016) 11
28. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017) 3, 4, 9, 16, 17, 26
29. Howgrave-Graham, N., Silverman, J.H., Whyte, W.: Choosing parameter sets forwithand. In: Menezes, A. (ed.) Topics in Cryptology – CT-RSA 2005. Lecture Notes in Computer Science, vol. 3376, pp. 118–135. Springer, Heidelberg, Germany, San Francisco, CA, USA (Feb 14–18, 2005) 2
30. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 9614, pp. 387–416. Springer, Heidelberg, Germany, Taipei, Taiwan (Mar 6–9, 2016) 12
31. 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.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 273–293. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012) 1
32. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018) 3, 5, 7, 17, 26

33. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: Ind-cca-secure key encapsulation mechanism in the quantum random oracle model, revisited. Cryptology ePrint Archive, Report 2017/1096 (July 2018), https://eprint.iacr.org/2017/1096/ 5
34. Jiang, H., Zhang, Z., Ma, Z.: On the non-tightness of measurement-based reductions for key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/494 (2019), https://eprint.iacr.org/2019/494 8
35. Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/134 (2019), https://eprint.iacr.org/2019/134 7
36. Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 552–586. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018) 11
37. Krawczyk, H.: HMQV: A high-performance secure Diffie-Hellman protocol. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 546–566. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005) 1, 5, 8
38. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007: 1st International Conference on Provable Security. Lecture Notes in Computer Science, vol. 4784, pp. 1–16. Springer, Heidelberg, Germany, Wollongong, Australia (Nov 1–2, 2007) 1, 5
39. Li, Y., Schäge, S.: No-match attacks and robust partnering definitions: Defining trivial attacks for security protocols is not trivial. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 1343–1360. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017) 1
40. Naehrig, M., Alkim, E., Bos, J., Ducas, L., Easterbrook, K., LaMacchia, B., Longa, P., Mironov, I., Nikolaenko, V., Peikert, C., Raghunathan, A., Stebila, D.: FrodoKEM. Tech. rep., National Institute of Standards and Technology (2017), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions 7
41. NIST: National institute for standards and technology. postquantum crypto project (2017), http://csrc.nist.gov/groups/ST/post-quantum-crypto/ 2
42. Persichetti, E.: Improving the efficiency of code-based cryptography. Ph.D. thesis (2012), http://persichetti.webs.com/Thesis%20Final.pdf 4
43. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 520–551. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018) 3, 4, 5, 6, 9, 11, 13, 14, 16, 17, 20, 26, 27
44. Schäge, S.: TOPAS: 2-pass key exchange with full perfect forward secrecy and optimal communication complexity. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015: 22nd Conference on Computer and Communications Security. pp. 1224–1235. ACM Press, Denver, CO, USA (Oct 12–16, 2015) 1
45. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332 (2004), http://eprint.iacr.org/2004/332 9
46. Toorani, M.: On continuous after-the-fact leakage-resilient key exchange. In: Proceedings of the Second Workshop on Cryptography and Security in Computing Systems. pp. 31:31–31:34. CS2 '15, ACM, New York, NY, USA (2015), http://doi.acm.org/10.1145/2694805.2694811 1
47. Yao, A.C.C., Zhao, Y.: OAKE: a new family of implicitly authenticated Diffie-Hellman protocols. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013: 20th Conference on Computer and Communications Security. pp. 1113–1128. ACM Press, Berlin, Germany (Nov 4–8, 2013) 1
48. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012) 11