Generic Authenticated Key Exchange in the Quantum Random Oracle Model

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

July 2, 2019

[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 generalization 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. Our reduction fixes several gaps in a previous proof (CRYPTO 2018), is tighter, and tolerates a larger correctness error.

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 [11, 19, 35, 30] as well as increasingly efficient designs of AKE protocols [34, 45, 41]. 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] and the protocol in [2], the latter of which has been criticized for having a flawed security proof and a weak security model [43, 36].

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 standardize new PKE and signature algorithms [38] 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. Natural 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.[1] 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.[2]

Previous Constructions of AKE from PKE. 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 transformation as $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{PKE}_{\mathrm{cca}}]$. 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 $\mathsf{PKE}_{\mathrm{cca}} = \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ [24, 25]. In its "implicit rejection" variant [27], it comes with a recently discovered security proof [40] 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 [40] requires the underlying scheme to be perfectly correct, and a generalization 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.

[1] 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.

[2] There exist generic transformations that can immunize 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., [13, 28]), 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.

- **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.

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. 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 [31], which, as we will explain later, contains a number of flaws and drawbacks.

1.1.1 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 [24, 25], FO yields an encryption scheme that is $\mathsf{IND\text{-}CCA}$ secure in the random oracle model [10] 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, [27] gave several modularizations 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 [27]:

- Transformation T: "Derandomization" and "re-encryption". 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

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

  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 [39, 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, [27] 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 loss in tightness, and 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 modularization was introduced in [40]: 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 (due to the use of the oneway-to-hiding lemma).

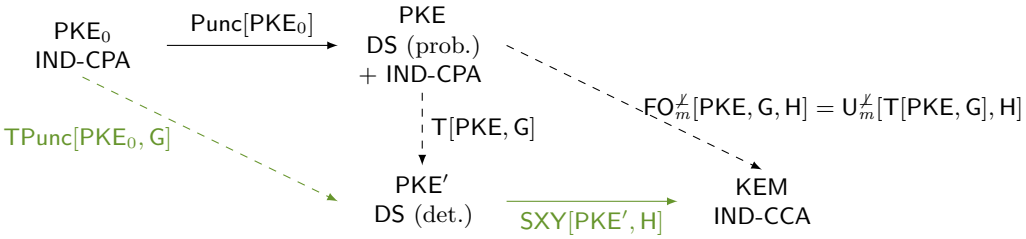


Figure 1: Comparison of [40]'s modular transformation (green) with ours. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions.

FO for KEMs: previous issues and security proof. However, the reduction that is given in [40] requires the underlying encryption scheme to be perfectly correct. While [31, 32] ([32] refers to the full version of [31] in its last revision from July 2018) gave security proofs for the non-modular transformations $\mathsf{FO}^{\not\perp}_m$ and $\mathsf{FO}^{\not\perp}$ [32, Thms. 1 and 2] as well as a security proof for SXY[3] (see [32, Thm. 6]) for schemes with correctness errors. We identify some flaws and drawbacks which we will discuss in Appendix A. In a nutshell, two main issues arise: The first issue is that to prove the non-modular statements, a lemma is used whose formal statement is unclear. One of its requirements might be unsatisfiable, rendering the proof impossible to verify. We structure our proof differently by following [40]'s modular approach as far as possible.[4] For more details on this issue and our strategy to avoid it, we refer to Appendix A.

The second issue is that the security statement given in [32, Thm. 6] is based on prerequisites that are not met by most lattice-based encryption schemes. Recall that SXY is only applicable to deterministic schemes since it reencrypts, and the issue stated above is due to the correctness definition for deterministic schemes that is used.[5] It is not straightforward to give a correctness definition for deterministic encryption schemes such that it fits known strategies to prove SXY tightly secure, but also is achievable by most lattice-based schemes. We circumvent this difficulty by resorting to a non-modularized proof that assumes a non-deterministic scheme.[6] Lastly, we want to stress that the statement of [32, Thm. 6] is not proven, and it is unclear how it could be proven with the standard notion of $\mathsf{IND\text{-}CCA}$ security. More details on these issues are also given in Appendix A.

Our transformation $\mathsf{FO}^{\not\perp}_m$ can be applied to any PKE scheme that is both $\mathsf{IND\text{-}CPA}$ and DS secure.

[3] Note that nomenclature of [32] is a bit misleading: while the their KEM from Figure 13 (and Theorem 6) is called $\mathsf{U}^{\not\perp}_m$, it is actually transformation SXY (it reencrypts during decryption, which $\mathsf{U}^{\not\perp}_m$ does not).

[4] We will first prove that $\mathsf{T}[-, \mathsf{G}]$ turns any suitable scheme into a scheme that is deterministically DS, and then plug in this result into [40]'s tight security proof for $\mathsf{U}^{\not\perp}_m$.

[5] The definition of correctness, in the deterministic setting, effectively requires that the scheme is perfectly correct for almost all public keys.

[6] When plugging in $\mathsf{T}[-, \mathsf{G}]$ into $\mathsf{U}^{\not\perp}_m$, we can change random oracle G during the security proof such that the scheme is rendered perfectly correct, a necessary condition to proceed with the tight security proof.

The reduction is tighter than the one that results from combining those of TPunc and SXY in [40], and also than the reduction given in [32]. 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 [32]) 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 $\mathsf{IND\text{-}CPA}$ alone, all that has to be done is to plug in transformation Punc. A visualization is given in Figure 1.

1.1.2 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 ($\mathsf{IND\text{-}AA}$) and the weaker notion of indistinguishability against active attacks without state reveal in the test session ($\mathsf{IND\text{-}StAA}$). $\mathsf{IND\text{-}AA}$ captures the classical notion of key indistinguishability (as introduced by Bellare and Rogaway [11]) as well as security against reflection attacks, key compromise impersonation (KCI) attacks, and weak forward secrecy (wFS) [34]. 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 [35] but strictly stronger than the CK model. Essentially, the $\mathsf{IND\text{-}AA}$ model states that the session key remains indistinguishable from a random one even if

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.

Note that $\mathsf{IND\text{-}AA}$ only excludes trivial attacks and is hence the strongest notion of security that can be achieved by any (non-parallel) two-message AKE protocol (relative to the set of oracle queries we allow).

We also consider the slightly weaker model $\mathsf{IND\text{-}StAA}$ (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 $\mathsf{IND\text{-}StAA}$ security is essentially the same notion that was achieved by the FSXY transformation [23].[7] In Appendix B we provide a more general perspective on how our model compares to existing ones.

1.1.3 Our Authenticated Key-Exchange Protocol.

Our transformation $\mathsf{FO}_{\mathsf{AKE}}$ transforms any passively secure PKE (with potential non-perfect correctness) into an $\mathsf{IND\text{-}StAA}$ 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 14, see Section 5.) Our main security result (Theorem 3) states that $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is an $\mathsf{IND\text{-}StAA}$-secure AKE if the underlying probabilistic PKE is DS as well as $\mathsf{IND\text{-}CPA}$ secure and has negligible correctness error, and furthermore G and H are modeled as quantum random oracles.

[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.

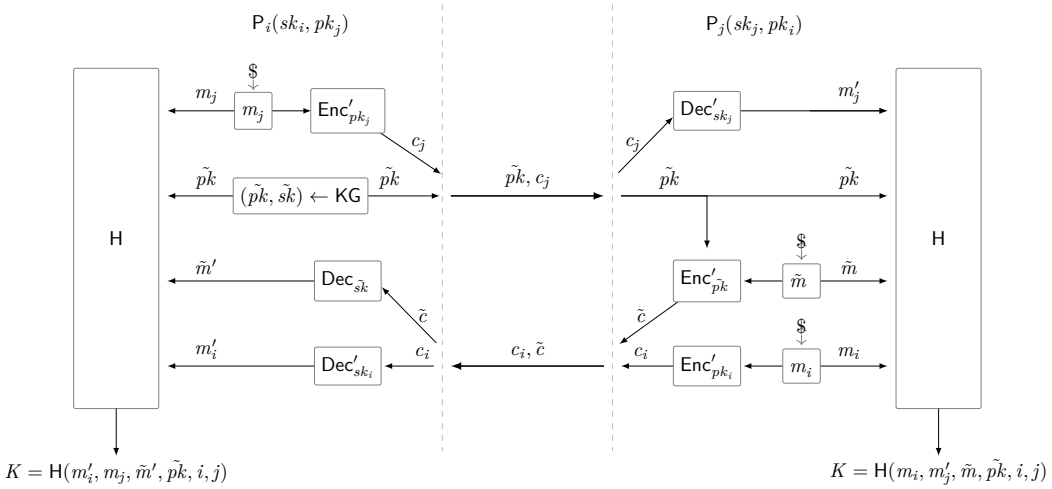


Figure 2: A visualisation of our authenticated key-exchange protocol $\mathsf{FO}_{\mathsf{AKE}}$. We make the convention that, in case any of the Dec' algorithms returns $\perp$, the session key K is derived deterministically and pseudorandomly from the player's state ("implicit rejection").

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). Adapting the simulation technique in [40], 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 $\mathsf{IND\text{-}AA}$ 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 generalize [40]'s result for deterministic encryption schemes with correctness errors discussed above.

Concrete Applications. Our transformation can be applied to any DS and $\mathsf{IND\text{-}CPA}$ secure PKE scheme with post-quantum security, e.g., Frodo [37], Kyber [16], and Lizard [5]. In fact, applying $\mathsf{FO}_{\mathsf{AKE}}$ to Kyber provides a formal security proof for the AKE protocol described in [16]. Note that most of the mentioned schemes are already DS secure under the same assumption as it is used for $\mathsf{IND\text{-}CPA}$ security and as mentioned above, the requirement of DS security can be waived with negligible loss of efficiency.

1.1.4 Open Problems.

In the literature, one can find several Diffie-Hellman based protocols that achieve $\mathsf{IND\text{-}AA}$ security, for example HMQV [34]. 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 $\mathsf{IND\text{-}AA}$ model, preferably with a security proof in the QROM. While we were able to generalize (and tighten) the proof of CCA security given in [40] for the *combined* transformation $\mathsf{FO}_m^{\not\perp} := \mathsf{U}_m^{\not\perp} \circ \mathsf{T}$ such that it covers encryption schemes that come with non-perfect correctness, it still remains an open problem to generalize the security proof of $\mathsf{U}_m^{\not\perp}$ such that it is applicable to *any* deterministic encryption scheme that is DS, even if it is not perfectly correct for more than neglibly many key pairs.

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.

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 [42, 12], we use code-based games. We implicitly assume boolean flags to be initialized 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 recognizable. 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 2.1 (Collision probability of key generation.). We define

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

Definition 2.2 (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 2.3 (γ-Spreadness.). [24] We say that PKE is γ-spread iff for all key pairs $(pk, sk) \in \mathrm{supp}(\mathsf{KG})$ 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 2.4 (Correctness). [27] 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 ($\mathsf{IND\text{-}CPA}$) for public-key encryption.

Definition 2.5 ($\mathsf{IND\text{-}CPA}$). Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. We define game $\mathsf{IND\text{-}CPA}$ game as in Figure 3, and the $\mathsf{IND\text{-}CPA}$ 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 $\mathsf{IND\text{-}CPA}$ security in the random oracle model model, where PKE and adversary A are given access to a random oracle.

DISJOINT SIMULATABILITY. Following [40], 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.

Definition 2.6 (DS) [40] Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, together with a PPT algorithm $\overline{\mathsf{Enc}}$. For quantum adversaries A, we define the *advantage against* PKE*'s disjoint simulatability* as

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

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

GAME IND-CPA$_b$	**GAME** IND-CCA	DECAPS($c \neq c^*$)
01 $(pk, sk) \leftarrow \mathsf{KG}$	06 $(pk, sk) \leftarrow \mathsf{KG}$	12 $K := \mathsf{Decaps}(sk, c)$
02 $(m_0^*, m_1^*, \mathrm{st}) \leftarrow \mathsf{A}_1(pk)$	07 $b \leftarrow_\$ \mathbb{F}_2$	13 **return** K
03 $c^* \leftarrow \mathsf{Enc}(pk, m_b^*)$	08 $(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
04 $b' \leftarrow \mathsf{A}_2(pk, c^*, \mathrm{st})$	09 $K_1^* \leftarrow_\$ \mathcal{K}$	
05 **return** b'	10 $b' \leftarrow \mathsf{A}^{\text{DECAPS}}(pk, c^*, K_b^*)$	
	11 **return** $[\![b' = b]\!]$	

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

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 (IND-CCA).

Definition 2.7 (IND-CCA). We define the IND-CCA game as in Figure 3 and the IND-CCA *advantage function of an adversary* A *(with binary output) against* KEM as

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

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 $\mathsf{O} : \mathbb{F}_2^n \to \mathbb{F}_2^m$, 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 [26] and [1].

Zhandry [46] 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 $\mathsf{O} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ and a $2q$-wise independent function f_{2q}. For concreteness, we view $f_{2q} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ 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 $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q \cdot \mathrm{Time}(\mathsf{O})$, where $\mathrm{Time}(\mathsf{O})$ 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 $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q \cdot \mathrm{Time}(f_{2q})$, and the time $\mathrm{Time}(f_{2q})$ to evaluate f_{2q} is linear in q. Following [40] and [33], 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 $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q$, which is approximately $\mathrm{Time}(\mathsf{A})$.

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 $\mathsf{O}_S^{\mathsf{SC}}$: Algorithm $\mathsf{O}_S^{\mathsf{SC}}$, 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 initializes 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 $\mathsf{O}_S^{\mathsf{SC}}$ ever answers with $|\psi', 1\rangle$ for some ψ'. To a quantum oracle $|\mathsf{G}\rangle$ and a subset $S \subset X$, Ambainis et al. associate the following punctured oracle $|\mathsf{G} \setminus \mathsf{S}\rangle$ that removes S from the domain of $|\mathsf{G}\rangle$ unless FIND occurs.

$|\mathsf{G} \setminus \mathsf{S}\rangle|\psi, \phi\rangle$
01 $|\psi', b\rangle := \mathsf{O}_S^{\mathsf{SC}}|\psi, 0\rangle$
02 **return** $|\mathsf{G}\rangle|\psi', \phi\rangle$

Figure 4: Punctured oracle $|\mathsf{G} \setminus \mathsf{S}\rangle$ for OW2H.

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 2.8 *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: "Preimage search in a random function"], [29, 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 [29, Lem. 3].

Lemma 2.9 (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,*

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

where games $\mathsf{GDPB}^{\mathsf{A}}_{\lambda,b}$ (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'

Proof. In this proof, let CGDPB_λ denote the game GDPB_λ as defined in [4] and [29], i.e., defined such that $\lambda_x = \lambda$ for all x. (Hence, we call it constant GDPB). The bound on GDPB_λ can be reduced to the known bound on CGDPB_λ by coupling the Bernoulli parameter to obtain the dependence on each $x \in X$: Let A be an adversary against game GDPB_λ, issuing at most q queries. Without loss of generality, we can assume that $\lambda > 0$. Consider adversary B against game CGDPB_λ, given in Figure 16.

$\underline{\mathsf{B}_1}$
$(\lambda_x)_{x\in X} \leftarrow \mathsf{A}_1$
$\lambda := \max_{x\in X} \lambda_x$
for all $x \in X$
  $\mu_x := \frac{\lambda_x}{\lambda}$
  $G(x) \leftarrow B_{\mu_x}$
return λ

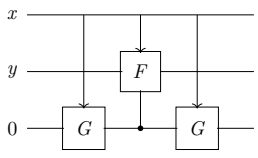
$\underline{\mathsf{B}_2^{|\mathsf{F}\rangle}}$
$b' \leftarrow \mathsf{A}_2^{|\mathsf{F}\cdot\mathsf{G}\rangle}$
return b'

Figure 5: Adversary B for the proof of Lemma 4.

For each $x \in X$, B picks $G(x)$ according to B_{μ_x}, where $\mu_x := \frac{\lambda_x}{\lambda} \in [0,1]$. B then executes A with oracle access to $|\mathsf{F}\cdot\mathsf{G}\rangle$ and returns A's output bit. If $F(x)$ is distributed according to B_λ for each x, then $(F\cdot G)(x)$ is distributed according to B_{λ_x}, and if F is the constant zero function, so is $F \cdot G$, hence B perfectly simulates game GDPB_λ for A and

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

We now argue that B can realize A's oracle access to $|\mathsf{F}\cdot\mathsf{G}\rangle$ in a way such that any query to $|\mathsf{F}\cdot\mathsf{G}\rangle$ by A triggers at most one query to $|\mathsf{F}\rangle$. To verify this claim, consider the following state transitions:



The dot indicates execution of $F(x)$, conditioned on $G(x)$. It's easy to see that $|x, y, 0\rangle$ transitions to $|x, y \oplus F(x), 0\rangle$ if $G(x) = 1$, and that $|x, y, 0\rangle$ transitions to $|x, y, 0\rangle$ if $G(x) = 0$, hence $|x, y, 0\rangle$ transitions to $|x, y \oplus (F \cdot G)(x), 0\rangle$, either way, and B can answer queries to $|\mathsf{F} \cdot \mathsf{G}\rangle$ by querying $|\mathsf{F}\rangle$ just once. Since B issues at most q queries to $|\mathsf{F}\rangle$, we can apply [29, Lem. 3] and obtain

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

□

3 The FO Transformation: QROM security with correctness errors

In Section 3.1, we modularize transformation TPunc that was given in [40] and that turns any public key encryption scheme that is $\mathsf{IND\text{-}CPA}$ 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 DS), and then applying transformation T. Next, in Section 3.2, we show that transformation $\mathsf{U}^{\not\perp}_m$, when applied to T, transforms any encryption scheme that is DS as well as $\mathsf{IND\text{-}CPA}$ into a KEM that is $\mathsf{IND\text{-}CCA}$ secure. We believe that many lattice-based schemes fulfill DS in a natural way,[8] but for the sake of completeness, we will show in Appendix D how transformation Punc can be used to waive the requirement of DS with negligible loss of efficiency.

3.1 Modularization of TPunc

We modularize transformation TPunc ("Puncturing and Encrypt-with-Hash") that was given in [40], and that turns any $\mathsf{IND\text{-}CPA}$ secure PKE scheme into a deterministic one that is DS. Note that apart from reencryption, $\mathsf{TPunc}[\mathsf{PKE}_0, \mathsf{G}]$ given in [40] and our modularization $\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0], \mathsf{G}]$ are equal. We first give transformation Punc that turns any $\mathsf{IND\text{-}CPA}$ secure scheme into a scheme that is both DS and $\mathsf{IND\text{-}CPA}$. In Section 3.1, we show that transformation T turns any scheme that is DS as well as $\mathsf{IND\text{-}CPA}$ secure into a deterministic scheme that is DS.

3.1.1 Transformation Punc: From IND-CPA to probabilistic DS security

Transformation Punc turns any $\mathsf{IND\text{-}CPA}$ 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 4. Note that transformation Punc will only be used as a helper transformation to achieve DS, generically. For more details on Punc, we refer to Appendix D.

$\underline{\mathsf{Enc}(pk, m \in \mathcal{M})}$	$\underline{\overline{\mathsf{Enc}}(pk)}$
01 $c \leftarrow \mathsf{Enc}_0(pk, m)$	03 $c \leftarrow \mathsf{Enc}_0(pk, \hat{m})$
02 **return** c	04 **return** c

Figure 6: Encryption and fake encryption sampling of $\mathsf{PKE} = \mathsf{Punc}[\mathsf{PKE}_0]$.

3.1.2 Transformation T: From probabilistic to deterministic DS security

Transformation T [7] turns any probabilistic public-key encryption scheme into a deterministic one. The transformed 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 [40, Theorem 3.3]) due to our use of the semi-classical O2H theorem.

[8] 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.

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 5. Note that Enc' deterministically computes the ciphertext as $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$.

$\underline{\mathsf{Enc}'(pk, m)}$	$\underline{\mathsf{Dec}'(sk, c)}$
01 $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	03 $m' := \mathsf{Dec}(sk, c)$.
02 **return** c	04 **if** $m' = \bot$ **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
	05 **return** $\bot$
	06 **else return** m'

Figure 7: Deterministic encryption scheme $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$.

The following lemma states that combined IND-CPA and DS security of PKE imply the DS security of PKE'.

Lemma 3.1 (DS security of PKE'). *If PKE is ϵ-disjoint, so is PKE'. For all adversaries A issuing at most q_G queries to $|\mathsf{G}\rangle$, there exist an adversary $\mathsf{B}_{\mathsf{IND}}$ 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 6. Per definition,

$$\begin{aligned}\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}'}(\mathsf{A}) &= |\Pr[G^\mathsf{A}_0 \Rightarrow 1] - \Pr[G^\mathsf{A}_1 \Rightarrow 1]| \\ &\leq |\Pr[G^\mathsf{A}_0 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]| + |\Pr[G^\mathsf{A}_1 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]| \ .\end{aligned}$$

$\underline{\text{Games } G_0\text{-}G_2}$		$\underline{\text{Game } G_4\text{-}G_5}$		$\underline{|\mathsf{G} \setminus \{\mathsf{m}^*\}\rangle|\psi, \phi\rangle}$
01 $pk \leftarrow \mathsf{KG}$		10 FIND := **false**		18 $\|\psi', b\rangle := \mathsf{O}^{\mathsf{SC}}_{\{\mathsf{m}^*\}}\|\psi, 0\rangle$
02 $m^* \leftarrow_\$ \mathcal{M}$		11 $pk \leftarrow \mathsf{KG}$		19 **if** $b = 1$
03 $c^* \leftarrow \overline{\mathsf{Enc}}(pk)$	$/\!/ G_0$	12 $m^* \leftarrow_\$ \mathcal{M}$		20 FIND := **true**
04 $r^* := \mathsf{G}(m^*)$	$/\!/ G_1$	13 $r^* \leftarrow_\$ \mathcal{R}$		21 **return** $\|\mathsf{G}\rangle\|\psi', \phi\rangle$
05 $r^* \leftarrow_\$ \mathcal{R}$	$/\!/ G_2\text{-}G_3$	14 $c^* := \mathsf{Enc}(pk, m^*; r^*)$	$/\!/ G_4$	
06 $c^* := \mathsf{Enc}(pk, m^*; r^*)$	$/\!/ G_1\text{-}G_3$	15 $c^* := \mathsf{Enc}(pk, 0; r^*)$	$/\!/ G_5$	
07 $b' \leftarrow \mathsf{A}^{\|G\rangle}(pk, c^*)$	$/\!/ G_0\text{-}G_1,\ G_3$	16 $b' \leftarrow \mathsf{A}^{\|\mathsf{G}\setminus\{\mathsf{m}^*\}\rangle}(pk, c^*)$		
08 $b' \leftarrow \mathsf{A}^{\|H\rangle}(pk, c^*)$	$/\!/ G_2$	17 **return** FIND		
09 **return** b'				

Figure 8: Games G_0 - G_5 for the proof of Lemma 1.

To upper bound $|\Pr[G^\mathsf{A}_0 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]|$, consider adversary B_{DS} against the disjoint simulatability of the underlying scheme PKE, given in Figure 7. B_{DS} runs in the time that is required 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 $c \leftarrow \mathsf{Enc}(pk, m^*)$,

$$|\Pr[G^\mathsf{A}_0 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{B}_{\mathsf{DS}}) \ .$$

It remains to upper bound $|\Pr[G^\mathsf{A}_1 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]|$. We claim that there exists an adversary $\mathsf{B}_{\mathsf{IND}}$ such that

$$|\Pr[G^\mathsf{A}_1 \Rightarrow 1] - \Pr[G^\mathsf{A}_3 \Rightarrow 1]| \leq 2\sqrt{q_\mathsf{G} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}_{\mathsf{IND}}) + \frac{4q_\mathsf{G}^2}{|\mathcal{M}|}} \ .$$

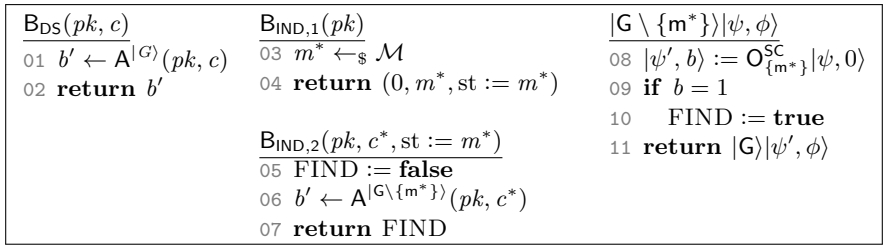


Figure 9: Adversaries B_{DS} and $\mathsf{B}_{\mathsf{IND}\text{-}}$ for the proof of Lemma 1.

GAME G_2. In game G_2, we replace oracle access to $|\mathsf{G}\rangle$ with oracle acess to $|\mathsf{H}\rangle$ 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 $\mathsf{H}(m^*) := r^*$. Since G is a random oracle, 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 $|\mathsf{G}\rangle$. Applying Theorem 5 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 Theorem 5 to 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 IND-CPA security of PKE, also given in Figure 7. $\mathsf{B}_{\mathsf{IND}}$ runs in the time that is required to run A and to measure and simulate 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 [40] got rid of the additional hash (sometimes called key confirmation) that was included in [27]'s quantum transformation $\mathsf{QU}^{\not\perp}_m$. SXY is essentially the (classical) transformation $\mathsf{U}^{\not\perp}_m$ that was also given in [27], 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 [27], reencryption is done during decryption of T.)

The security proof given in [40] 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: SXY starts from a deterministic encryption scheme PKE', and it is unclear how to reasonably define correctness for deterministic encryption schemes such that it fits the proof's strategy. The correctness term δ we have to consider reduces to the probability that for the sampled key pair, *at least one* message exists that inhibits decryption failure, i.e., the probability that the scheme is not perfectly correct for the sampled key pair. But with this definition, the security statements given in the theorem are not meaningful for most lattice-based encryption schemes since in most cases, there exist some messages inducing decryption failure for each key pair. 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 IND-CPA into a KEM that is IND-CCA secure in the QROM, even if the underlying encryption scheme comes with a small probability of decryption failure. This is achieved by modifying random oracle G during the proof such that the encryption scheme is rendered perfectly correct. Our reduction is tighter as the (combined) reduction in [40] due to our tighter security proof for T (see Section 3.1).

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 $\mathsf{H}_\mathsf{r} : \mathcal{C} \to \mathcal{K}$ 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 8.

$\underline{\mathsf{Encaps}(pk)}$	$\underline{\mathsf{Decaps}(sk, c)}$
01 $m \leftarrow_\$ \mathcal{M}$	05 $m' := \mathsf{Dec}(sk, c)$
02 $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$	06 **if** $m' = \perp$ **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
03 $K := \mathsf{H}(m)$	07 **return** $K := \mathsf{H}_\mathsf{r}(c)$
04 **return** (K, c)	08 **else return** $K := \mathsf{H}(m')$

Figure 10: 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 H_r is used to generate random values whenever reencryption fails. This strategy is called implicit reject. Amongst others, it is used in [27], [40], and [31]. For simplicity of the proof, 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. The following theorem (whose proof is essentially the same as in [40] except for the consideration of possible decryption failure) establishes that IND-CCA security of KEM reduces to DS and IND-CPA security of PKE, in the quantum random oracle model.

Theorem 3.2 (PKE DS + IND-CPA $\overset{\text{QROM}}{\Rightarrow}$ KEM IND-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)* IND-CCA *adversary* A *issuing at most q_D (classical) queries to the decapsulation oracle* Decaps*, at most q_H quantum queries to $|\mathsf{H}\rangle$, and at most q_G quantum queries to $|\mathsf{G}\rangle$, there exist (quantum) adversaries* B_DS *and* B_IND *such that*

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

and the running time of B_DS *and* B_IND *is about that of* A.

Proof. Let A be an adversary against the IND-CCA security of KEM, issuing at most q_D queries to Decaps, at most q_H queries to the quantum random oracle $|\mathsf{H}\rangle$, and at most q_G queries to the quantum random oracle $|\mathsf{G}\rangle$. Consider the sequence of games given in Figure 9.

Game G_0. Since game G_0 is the original IND-CCA game,

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) = |\Pr[G^\mathsf{A}_0 \Rightarrow 1] - 1/2| \ .$$

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

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

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

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

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 $(pk, sk) \leftarrow \mathsf{KG}$.

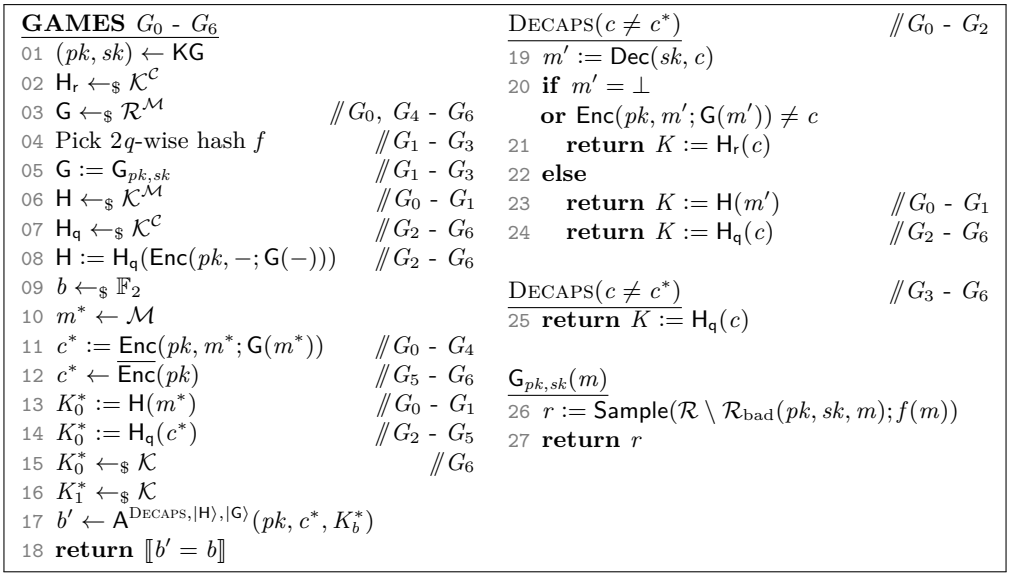


Figure 11: Games G_0 - G_6 for the proof of Theorem 1. f (lines 04 and 26) is an internal $2q$-wise independent hash function, where $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 2 \cdot q_D + 1$, that cannot be accessed by A. $\mathsf{Sample}(Y)$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_\$ Y$. $\mathsf{Sample}(Y; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(Y)$ using explicitly given randomness $f(m)$.

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 4) in Figure 10, issuing $q_{\mathsf{G}} + q_D + 1$ queries to $|\mathsf{F}\rangle$. 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 19 if $\mathsf{F}(m) = 0$ and in line 21 if $\mathsf{F}(m) = 1$ is uniformly distributed in $\mathcal{R}$. Therefore, G is a (quantum) 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 4,

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

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

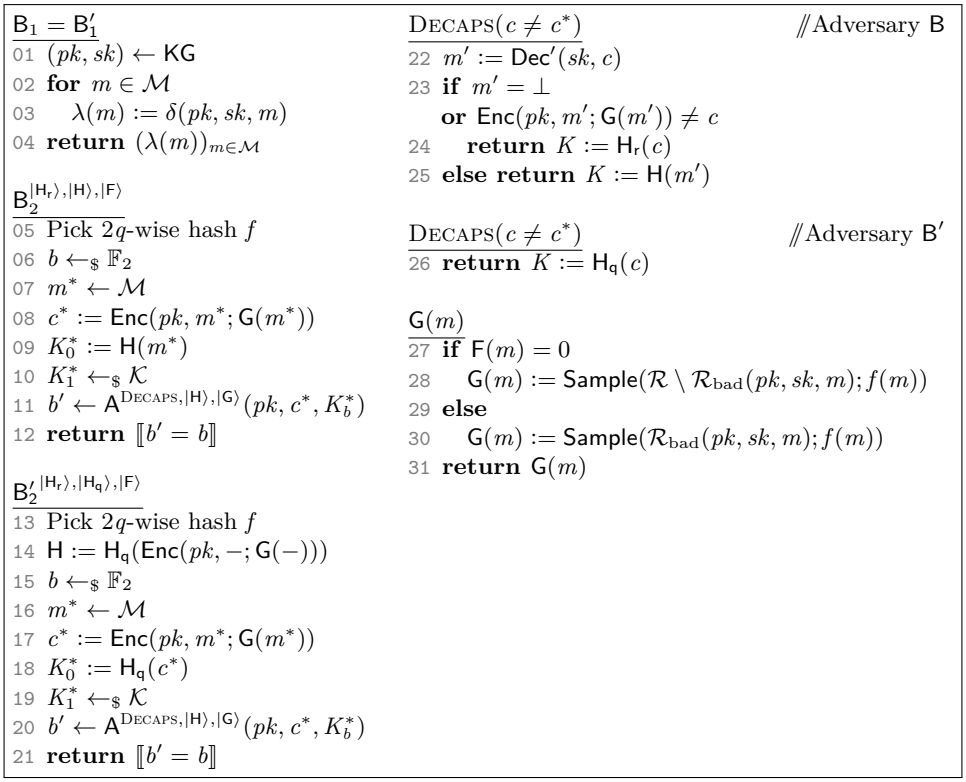


Figure 12: Adversaries B and B' executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ with access to $|\mathsf{F}\rangle$ (and additional oracles $|\mathsf{H_r}\rangle$ and $|\mathsf{H}\rangle$ or $|\mathsf{H_q}\rangle$, respectively) for the proof of Theorem 1. Parameters $\delta(pk, sk, m)$ are defined in Equation (5). Function f (lines 19 and 21) 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.

oracle $\textsc{Decaps}(c)$ returns a random value (that can not possibly correlate to any random oracle query to $|\mathsf{H}\rangle$) in both games, therefore $\textsc{Decaps}(c)$ is a random value independent of all other input to A in both games. And if there exists some message m such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, $\textsc{Decaps}(c)$ would have returned $\mathsf{H_q}(c)$ in both games, anyway: Since $\mathsf{G}(m) \in \mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk, sk, m)$ 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 $|\mathsf{F}\rangle$) is also given in Figure 10.

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 [40], 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 C_{DS} against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ given in Figure 11. Since 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,

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

and according to Lemma 1, there exist an adversary B_{DS} and an adversary $\mathsf{B}_{\mathsf{IND}}$ with roughly the same running time such that

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

$\mathsf{C}_{\mathsf{DS}}^{|\mathsf{G}\rangle,|\mathsf{H}_\mathsf{r}\rangle|\mathsf{H}_\mathsf{q}\rangle}(pk, c^*)$

01 $b \leftarrow_{\$} \mathbb{F}_2$
02 $K_0^* := \mathsf{H}_\mathsf{q}(c^*)$
03 $K_1^* \leftarrow_{\$} \mathcal{K}$
04 $b' \leftarrow \mathsf{A}^{\textsc{Decaps},|\mathsf{H}\rangle,|\mathsf{G}\rangle}(pk, c^*, K_b^*)$
05 **return** $[\![b' = b]\!]$

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

06 **return** $K := \mathsf{H}_\mathsf{q}(c)$

Figure 13: Adversary C_{DS} (with access to additional oracles $|\mathsf{H}_\mathsf{r}\rangle$ and $|\mathsf{H}_\mathsf{q}\rangle$) against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ for the proof of Theorem 1.

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 $|\mathsf{H}_\mathsf{q}\rangle$ could possibly contain c^*. Each query to $|\mathsf{H}_\mathsf{q}\rangle$ is either a classical query, triggered by A querying Decaps on some ciphertext c, or a query in superposition, triggered by A querying $|\mathsf{H}\rangle$. Since queries to Decaps on c^* are explicitly forbidden, the only possibility would be one of A's queries to $|\mathsf{H}\rangle$. A's queries to $|\mathsf{H}\rangle$ trigger queries to $|\mathsf{H}_\mathsf{q}\rangle$ that are 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,

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

3.2.1 CCA security wihout disjoint simulatability.

The following theorem establishes that plugging in transformation Punc before using $\mathsf{FO}^{\not\perp}_m$ achieves IND-CCA security from IND-CPA security alone, as long as PKE is γ-spread (see Definition 3).

Theorem 3.3 (CCA security of $\mathsf{FO}^{\not\perp}_m \circ \mathsf{Punc}$.). *Assume PKE_0 to be δ-correct and γ-spread, and let $\hat{m} \in \mathcal{M}$. Let $\mathsf{KEM} := \mathsf{FO}^{\not\perp}_m[\mathsf{Punc}[\mathsf{PKE}, \hat{m}], \mathsf{G}, \mathsf{H}]$. Then, for any (quantum) IND-CCA adversary A issuing at most q_D (classical) queries to the decapsulation oracle Decaps, at most q_{H} quantum queries to $|\mathsf{H}\rangle$, and at most q_{G} quantum queries to $|\mathsf{G}\rangle$, there exist (quantum) CPA adversaries B_1 and B_2 against PKE_0 such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) \leq\ & (8 \cdot (3 \cdot q_{\mathsf{G}} + 2 \cdot q_{\mathsf{H}} + q_D + 6)^2 + 1) \cdot \delta + \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) \\ & + 2 \cdot \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4(q_{\mathsf{G}} + q_{\mathsf{H}})^2}{|\mathcal{M}| - 1}} + 2^{-\gamma} \ ,\end{aligned}$$

and the running time of B_1 and B_2 is about that of A.

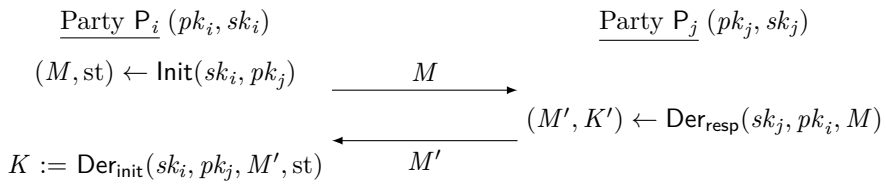
Since the proof is somewhat similar to the proof of Theorem 1, it is outsourced to Appendix D.

4 Two-Message Authenticated Key Exchange

A two-message key exchange protocol $\mathsf{AKE} = (\mathsf{KG}, \mathsf{Init}, \mathsf{Der}_{\mathsf{init}}, \mathsf{Der}_{\mathsf{resp}})$ consists of four algorithms. Given the security parameter, the key generation algorithm KG outputs a key pair (pk, sk). The initialization algorithm Init, on input sk and pk', outputs a message m and a state st. The responder's derivation algorithm $\mathsf{Der}_{\mathsf{resp}}$, on input sk', pk and m, outputs a key K, and also a message m'. The initiator's derivation algorithm $\mathsf{Der}_{\mathsf{init}}$, 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 $\mathsf{KG}, \mathsf{Init}, \mathsf{Der}_{\mathsf{init}}$, and $\mathsf{Der}_{\mathsf{resp}}$ 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 $(M, \mathrm{st}) \leftarrow \mathsf{Init}(sk_i, pk_j)$ and sends M to P_j.
2. P_j computes $(M', K') \leftarrow \mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M)$ and sends M' to P_i.
3. P_i computes $K := \mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \mathrm{st})$.



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:

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 12, page 21) and to relay their outputs faithfully as well as modifying the data on transit. Moreover, the attacker is additionally granted queries to reveal both secret process data, namely using REVEAL and REV-STATE queries, and parties' secret keys using CORRUPT queries, see Figure 13, page 22. 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.

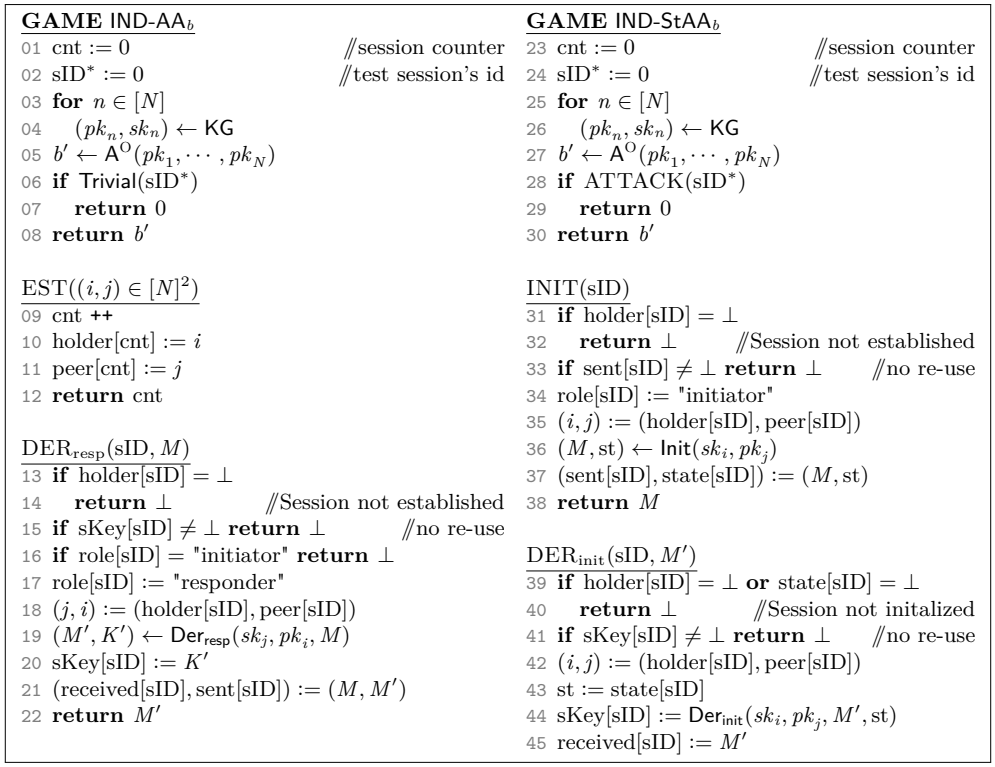


Figure 14: Games IND-AA$_b$ and IND-StAA$_b$ for AKE, where $b \in \mathbb{F}_2$. The collection of oracles O used in lines 05 and 27 is defined by O := {EST, INIT, DER$_{\mathrm{resp}}$, DER$_{\mathrm{init}}$, REVEAL, REV-STATE, CORRUPT, TEST}. Oracles REVEAL, REV-STATE, CORRUPT, and TEST are given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

Definition 4.1 (Key Indistinguishability of AKE). We define games IND-AA$_b$ and IND-StAA$_b$ for $b \in \mathbb{F}_2$ as in Figure 12 and Figure 13. We define the IND-AA *advantage function of an adversary* A *against* AKE as

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

and the IND-StAA *advantage function of an adversary* A *against* AKE *excluding test-state-attacks* as

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

We call a session *completed* iff sKey[sID] $\neq \perp$, which implies that either DER$_{\mathrm{resp}}$(sID, m) or DER$_{\mathrm{init}}$(sID, m) was queried for some message m. We say that a completed session sID *was recreated* iff there exists a session sID$' \neq$ sID such that (holder[sID], peer[sID]) = (holder[sID$'$], peer[sID$'$]), role[sID] = role[sID$'$], sent[sID] = sent[sID$'$], received[sID] = received[sID$'$] and state[sID] = state[sID$'$]. We say that two completed sessions sID$_1$ and sID$_2$ *match* iff (holder[sID$_1$], peer[sID$_1$]) = (peer[sID$_2$], holder[sID$_2$]), (sent[sID$_1$], received[sID$_1$]) = (received[sID$_2$], sent[sID$_2$]), and role[sID$_1$] $\neq$ role[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*.

Helper procedure Trivial (Figure 13) is used in all games to exclude the possibility of trivial attacks, and helper procedure ATTACK (also Figure 13) is defined in games IND-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 $\mathfrak{M}$(sID*) of all matching sessions that were executed throughout the game (see line 55), and A's output bit b' only 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:

```
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                            //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)                                  //only one query
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)
75 sID* := sID
76 if sKey[sID*] = ⊥
77   return ⊥
78 K_0^* := sKey[sID*]
79 K_1^* ←$ 𝒦
80 return K_b^*
```

Figure 15: Helper procedures Trivial and ATTACK and oracles REVEAL, REV-STATE, CORRUPT, and TEST of games IND-AA and IND-StAA defined in Figure 12.

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 ¬corrupted[i] or ¬revState[sID*] since otherwise, A is allowed to obtain all information required to trivially compute $\mathsf{Der}(sk_i, pk_j, \text{received[sID*]}, \text{state[sID*]})$.

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 ¬corrupted[j] or ¬revState[sID] for all sID s. th. sID ∈ 𝔐(sID*) for the same reason as discussed in 2: A could trivially compute $\mathsf{Der}(sk_j, pk_i, \text{received[sID]}, \text{state[sID]})$ 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 𝔐(sID*) ≠ ∅ or ¬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 sKey[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 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 . For some protocols, this attack is possible even without knowledge of any of the static secret keys. 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 14.

$\underline{\mathsf{Init}}(sk_i, pk_j)$:
01 $m_j \leftarrow_{\$} \mathcal{M}$
02 $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
03 $(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{KG}$
04 $M := (\tilde{pk}, c_j)$
05 $\text{st} := (\tilde{sk}, m_j, M)$
06 **return** (M, st)

$\underline{\mathsf{Der}_{\mathsf{resp}}}(sk_j, pk_i, M)$:
07 Parse $(\tilde{pk}, c_j) := M$
08 $m_i, \tilde{m} \leftarrow_{\$} \mathcal{M}$
09 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
10 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
11 $M' := (c_i, \tilde{c})$
12 $m'_j := \mathsf{Dec}(sk_j, c_j)$
13 **if** $m'_j = \bot$
or $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
14 $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, i, j, M, M')$
15 **else**
16 $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
17 **return** (M', K')

$\underline{\mathsf{Der}_{\mathsf{init}}}(sk_i, pk_j, M', \text{st})$:
18 Parse $(c_i, \tilde{c}) := M'$
19 Parse $(\tilde{sk}, m_j, M := (\tilde{pk}, c_j)) := \text{st}$
20 $m'_i := \mathsf{Dec}(sk_i, c_i)$
21 $\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
22 **if** $m'_i = \bot$
or $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
23 **if** $\tilde{m}' = \bot$
24 $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$
25 **else**
26 $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', i, j, M, M')$
27 **else if** $\tilde{m}' = \bot$
28 $K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, i, j, M, M')$
29 **else** $K := \mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
30 **return** K

Figure 16: 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 [27], [40] and [31].) 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.

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 5.1 (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 $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ issuing at most $q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S$ queries to* G *such that*

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

and the running time of A_{DS} *is about that of* B. *Due to Lemma 1, there exist adversaries* C_{DS} *and* $\mathsf{C}_{\mathsf{IND}}$ *against* PKE *such that*

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

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

Proof Sketch. To prove $\mathsf{IND\text{-}StAA}$ 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 $\mathsf{IND\text{-}StAA}$ (see Figure 12). 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 $\mathsf{IND\text{-}StAA}$. 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 $\mathsf{IND\text{-}StAA}$, at least one message can not be obtained trivially and is still protected by PKE's $\mathsf{IND\text{-}CPA}$ security, and the respective ciphertext can be replaced with fake encryptions due to PKE's disjoint simulatability. Consequently, the session key K is pseudorandom. 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 [40] 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 1. 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 refer to Section 3.2.

Proof. Let B be an adversary against the $\mathsf{IND\text{-}StAA}$ security of AKE, establishing S sessions and issuing 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. We will first examine the case that B executed the test session honestly (i.e., the case that $\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$, where $\mathfrak{M}(\mathrm{sID}^*)$ is defined in Figure 13 , line 55, as the list of matching sessions that were executed throughout game $\mathsf{IND\text{-}StAA}$), in the second part we will examine the case that B tampered with the test session (i.e., the case that $\mathfrak{M}(\mathrm{sID}^*) = \varnothing$).

$$\begin{aligned}&|\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1]| \\ &\qquad \leq |\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]| \\ &\qquad\quad + |\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing]| \ .\end{aligned}$$

Lemma 5.2 *There exists an adversary* A *such that*

$$\begin{aligned}|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) \neq \varnothing]| \\ \leq 2 \cdot S \cdot (S+N) \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}) + 32 \cdot (S+N) \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S + 1)^2 \cdot \delta \\ + 4 \cdot S^2 \cdot \epsilon_{dis} + S^2 \cdot (N+1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + 2 \cdot S^2 \cdot \mu(\mathsf{KG}) \ ,\end{aligned}$$

and the running time of A *is about that of* B*.*

The upper bound is proven in appendix E. Intuition is as follows: While B might have obtained the secret key of the initialising session's peer in both cases, B might not both reveal its internal state and corrupt its holder, hence either the message that belongs to its holder (i.e., m_i^*) or the message that belongs to its ephemeral key (i.e., $\tilde{m}^*$) are still protected by PKE's $\mathsf{IND\text{-}CPA}$ security, and the respective ciphertext can hence be replaced with a fake ciphertext (due to $\mathsf{T}[\mathsf{PKE},\mathsf{G}]$'s disjoint simulatability).

Lemma 5.3 *There exists an adversary* A' *such that*

$$\begin{aligned}|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) = \varnothing]| \\ \leq 4 \cdot SN \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}') + 64 \cdot N \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S)^2 \cdot \delta + 4 \cdot SN\epsilon_{dis} \ ,\end{aligned}$$

and the running time of A *is about that of* B*.*

The upper bound is proven in appendix F. The proof is essentially the same and only differs in the following way: since no matching sessions exists, B is neither allowed to reveal the test session's state nor to corrupt its peer. Depending on whether role[sID*] = "initiator" or role[sID*] = "responder", we can rely on the secrecy of either m_i^* or m_j^*.

Folding A and A' into one adversary A_{DS}, we obtain

$$\begin{aligned}|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1]| \\ \leq 2 \cdot S \cdot (S + 3 \cdot N) \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}_{\mathsf{DS}}) + 32 \cdot (S + 3 \cdot N) \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\ + 4 \cdot S \cdot (S+N) \cdot \epsilon_{\mathrm{dis}} + S^2 \cdot (N+1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + 2 \cdot S^2 \cdot \mu(\mathsf{KG}) \ .\end{aligned}$$

5.1 IND-StAA security wihout disjoint simulatability

The following theorem establishes that plugging in transformation Punc before using $\mathsf{FO}_{\mathsf{AKE}}$ achieves $\mathsf{IND\text{-}StAA}$ security from $\mathsf{IND\text{-}CPA}$ security alone, as long as PKE is γ-spread.

Theorem 5.4 (IND-StAA security of $\mathsf{FO}_{\mathsf{AKE}} \circ \mathsf{Punc}$.). *Assume* PKE_0 *to be* δ*-correct and* γ*-spread, and let* $\hat{m} \in \mathcal{M}$*. Let* $\mathsf{AKE} := \mathsf{FO}_{\mathsf{AKE}}[\mathsf{Punc}[\mathsf{PKE}, \hat{m}], \mathsf{G}, \mathsf{H}]$*. Then, for any* $\mathsf{IND\text{-}StAA}$ *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 exist adversaries* B_1 *and* B_2 *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{B}) \leq 2S \cdot (S + 3 \cdot N) \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2)} \right) \\ + (S+3N) \cdot \left(8q^2 \cdot (S+4) + S\right) \cdot \delta + S \cdot (S+3N) \cdot 2^{-\gamma} \\ + \frac{S(8q \cdot (S+3N) + S^2)}{\sqrt{|\mathcal{M}| - 1}} + S \cdot (3S^2 + 2) \cdot \mu(\mathsf{KG}) \ ,\end{aligned}$$

and the running time of B_1 *and* B_2 *is about that of* B*.*

Since the proof is somewhat similar to the proof of Theorem 3, it is outsourced to Appendix G.

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) (Cited on page 31.)

[2] Alawatugoda, J., Boyd, C., Stebila, D.: Continuous after-the-fact leakage-resilient key exchange. In: ACISP 14. pp. 258–273. LNCS (2014) (Cited on page 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 (Cited on page 5, 29, 32.)

[4] Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th FOCS. pp. 474–483 (2014) (Cited on page 33.)

[5] Banik, S., Isobe, T.: Some cryptanalytic results on lizard. Cryptology ePrint Archive, Report 2017/346 (2017), http://eprint.iacr.org/2017/346 (Cited on page 8.)

[6] Beals, R., Buhrman, H., Cleve, R., Mosca, M., Wolf, R.: Quantum lower bounds by polynomials. In: 39th FOCS. pp. 352–361 (1998) (Cited on page 31.)

[7] Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: CRYPTO 2007. pp. 535–552. LNCS (2007) (Cited on page 11.)

[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 ACM STOC. pp. 419–428 (1998) (Cited on page 1.)

[9] Bellare, M., Hofheinz, D., Kiltz, E.: Subtleties in the definition of IND-CCA: When and how should challenge decryption be disallowed? Journal of Cryptology 28(1), 29–48 (Jan 2015) (Cited on page 30.)

[10] Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: ACM CCS 93. pp. 62–73 (1993) (Cited on page 3.)

[11] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: CRYPTO'93. pp. 232–249. LNCS (1994) (Cited on page 1, 6.)

[12] Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: EUROCRYPT 2006. pp. 409–426. LNCS (2006) (Cited on page 8.)

[13] 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 (Cited on page 2.)

[14] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT 2011. pp. 41–69. LNCS (2011) (Cited on page 2, 31.)

[15] Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: CRYPTO 2013. pp. 361–379. LNCS (2013) (Cited on page 31.)

[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 (Cited on page 3, 8.)

[17] Boyd, C., Cliff, Y., Nieto, J.G., Paterson, K.G.: Efficient one-round key exchange in the standard model. In: ACISP 08. pp. 69–83. LNCS (2008) (Cited on page 1, 2, 6.)

[18] Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: CRYPTO 2015. pp. 609–629. LNCS (2015) (Cited on page 31.)

[19] Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: EUROCRYPT 2001. pp. 453–474. LNCS (2001) (Cited on page 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) (Cited on page 3.)

[21] Dent, A.W.: A designer's guide to KEMs. In: 9th IMA International Conference on Cryptography and Coding. pp. 133–151. LNCS (2003) (Cited on page 3.)

[22] Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: EUROCRYPT 2004. pp. 342–360. LNCS (2004) (Cited on page 2.)

[23] Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. In: PKC 2012. pp. 467–484. LNCS (2012) (Cited on page 1, 2, 6.)

[24] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: CRYPTO'99. pp. 537–554. LNCS (1999) (Cited on page 3, 9.)

[25] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology 26(1), 80–101 (Jan 2013) (Cited on page 3.)

[26] Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: CRYPTO 2016. pp. 60–89. LNCS (2016) (Cited on page 31.)

[27] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: TCC 2017. pp. 341–371. LNCS (2017) (Cited on page 3, 4, 9, 13, 14, 23, 29.)

[28] Howgrave-Graham, N., Silverman, J.H., Whyte, W.: Choosing parameter sets forwithand. In: CT-RSA 2005. pp. 118–135. LNCS (2005) (Cited on page 2.)

[29] Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: PKC 2016. pp. 387–416. LNCS (2016) (Cited on page 33, 34.)

[30] Jager, T., Kohlar, F., Schäge, S., Schwenk, J.: On the security of TLS-DHE in the standard model. In: CRYPTO 2012. pp. 273–293. LNCS (2012) (Cited on page 1.)

[31] Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: CRYPTO 2018. pp. 96–125. LNCS (2018) (Cited on page 3, 5, 14, 23, 28.)

[32] 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/ (Cited on page 5, 28, 29.)

[33] Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: EUROCRYPT 2018. pp. 552–586. LNCS (2018) (Cited on page 32.)

[34] Krawczyk, H.: HMQV: A high-performance secure Diffie-Hellman protocol. In: CRYPTO 2005. pp. 546–566. LNCS (2005) (Cited on page 1, 6, 8, 30.)

[35] LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: ProvSec 2007. pp. 1–16. LNCS (2007) (Cited on page 1, 6.)

[36] Li, Y., Schäge, S.: No-match attacks and robust partnering definitions: Defining trivial attacks for security protocols is not trivial. In: ACM CCS 17. pp. 1343–1360 (2017) (Cited on page 1.)

[37] 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 (Cited on page 8.)

[38] NIST: National institute for standards and technology. postquantum crypto project (2017), http://csrc.nist.gov/groups/ST/post-quantum-crypto/ (Cited on page 2.)

[39] Persichetti, E.: Improving the efficiency of code-based cryptography. Ph.D. thesis (2012), http://persichetti.webs.com/Thesis%20Final.pdf (Cited on page 4.)

[40] Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: EUROCRYPT 2018. pp. 520–551. LNCS (2018) (Cited on page 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 17, 23, 24, 29, 30, 32.)

[41] Schäge, S.: TOPAS: 2-pass key exchange with full perfect forward secrecy and optimal communication complexity. In: ACM CCS 15. pp. 1224–1235 (2015) (Cited on page 1.)

[42] 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 (Cited on page 8.)

[43] 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 (Cited on page 1.)

[44] Unruh, D.: Revocable quantum timed-release encryption. In: EUROCRYPT 2014. pp. 129–146. LNCS (2014) (Cited on page 28.)

[45] Yao, A.C.C., Zhao, Y.: OAKE: a new family of implicitly authenticated Diffie-Hellman protocols. In: ACM CCS 13. pp. 1113–1128 (2013) (Cited on page 1.)

[46] Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: CRYPTO 2012. pp. 758–775. LNCS (2012) (Cited on page 32.)

A Problems in and comparison with the proofs of [31].

In this section we will discuss some problems we encountered in the proofs of [31]. We refer to its current eprint version [32]. Due to the structure of the non-modular proofs of [32, Thms. 1 and 2], the original OW2H lemma [44, Lem. 31: "One-way to hiding"] cannot be used to decouple the challenge plaintext from the adversary's view since random oracles H and G are not independent of each other. As a consequence, a new lemma called "One-way to hiding with redundant oracle" is introduced (see [32, Lem. 3]). It is applied in the proof of Theorem 1 (to upper bound the distinguishing advantage between games 5 and 6). Unfortunately, we were not able to verify that Lemma 3 can be applied during the proof: In lemma 3, the assumption is made that $\mathsf{O}(x)$ is uniformly distributed. To justify the game-hop from game 5 to game 6, both $\mathsf{G}(m^*)$ as well as $\mathsf{H}(c^*)$ have to be replaced with random values. To this end, the lemma's oracle O is identified with $\mathsf{G} \times \mathsf{H}_1(\mathsf{Enc}(-;\mathsf{G}(-)))$, and x with m^*. We claim that with this identification, $\mathsf{O}(x) = (\mathsf{G}(m^*), \mathsf{H}_1(\mathsf{Enc}(m^*;\mathsf{G}(m^*))))$ can not be seen as uniformly random since encryption might not be injective. Intuitively, either G can be kept a random oracle, but then the second part of O is not, or random oracle G could be kept sampling only good randomness (as before game G_5), rendering the second part of O random, but then the first part is not. Either way, it is unclear why Lemma 3 can be applied.[9] Intuitively, the problem is that G and H are intertwined. During our proof, we circumvent this difficulty by following [40]'s modular approach as far as we managed to: In [40], the original OW2H lemma only needs to be applied for random oracle G (to prove that PKE' is deterministically DS, as reflected in Figure 1). Once deterministic DS is achieved, oracle H does not have to be reprogrammed (instead, a fake encryption is sampled) and hence, OW2H does not have to be applied again.

To explain in which sense we followed the modular approach of [40] *as far as we managed to*, we will point out some issues regarding the security claim for SXY[10] [32, Thm. 6] in an attempt to illustrate the difficulties in proving SXY secure if the underlying scheme comes with non-perfect correctness: [32,

[9] It might be possible to apply Lemma 3 twice, once for G (while it is random) and once for $\mathsf{H}_1(\mathsf{Enc}(-;\mathsf{G}(-)))$ (after switching to "good" G). But that would lead to structurally different reductions, and furthermore, to nested square roots. While we also cannot exclude the possibility that this issue could be resolved by applying [3, Thm. 1: "Semi-classical O2H"], this approach would also result in structurally different reductions and would require a stronger security assumption for the underlying scheme.

[10] Recall that while the KEM discussed in theorem 6 is called $\mathsf{U}^{\not\perp}_m$, it differs from the original transformation $\mathsf{U}^{\not\perp}_m$ since it reencrypts.

Thm. 6] states that SXY turns any PKE scheme that is oneway-secure into a KEM that is IND-CCA secure, with the correctness term δ being included into the upper bound as a summand $4q_E\sqrt{\delta}$, where q_E is said to denote the number of queries to an encryption oracle.

The first drawback is that for deterministic schemes, the correctness term δ defined in [27] and used in [32, Thm. 6] reduces to the probability that for the sampled key pair, *at least one* message exists that inhibits decryption failure, i.e., the probability that the scheme is not perfectly correct for the sampled key pair. With this definition, the security statements given in the theorem are not meaningful for most lattice-based encryption schemes since in most cases, there exist some messages inducing decryption failure for each key pair, though this fraction might be small. Unfortunately, it is not straightforward to reasonably define correctness for deterministic encryption schemes such that it fits existing proof strategies, but also is being met by lattice-based schemes at the same time. We also would like to mention that the statement of [32, Thm. 6], in the case where the underlying scheme is DS, follows trivially (and with a better upper bound) from [40, Thm. 4.2: "Security of SXY in the QROM"].[11]

Another issue is that the statement is claimed to follow directly from combining some proofs that were given before. However, none of the mentioned proofs include an encryption oracle, and it is unclear how this encryption oracle can be introduced such that its definition makes sense and still enables a reduction to deal with correctness errors: Either pk is not given to the reduction that deals with correctness errors and hence, game IND-CCA cannot be simulated, or pk is given to the reduction and hence, introducing oracle access to the encryption oracle makes no sense. We note that the notion of IND-CCA security could be modified such that instead of being given pk, the adversary has access to an encapsulation oracle. This alteration could allow for a reduction, but it is straightforward that this security notion would be strictly weaker.

The problems discussed above reflect why we weren't able to generalize [40]'s modular analysis in a straightforward manner: In fact, we did not manage to define correctness for deterministic encryption schemes such that the definition bridges the gap between what is achievable by most lattice-based schemes and what is needed to fit existing proof strategies. This difficulty is solved by resorting to a non-modularized proof: What we show is that the KEM resulting from applying $\mathsf{FO}_m^{\not\perp} := \mathsf{U}_m^{\not\perp} \circ \mathsf{T}$ is IND-CCA secure in the QROM. To this end, we first prove that $\mathsf{T}[-, \mathsf{G}]$ turns any suitable scheme into a scheme that is deterministically DS, and then plug in this result into [40]'s tight security proof. When plugging in $\mathsf{T}[-, \mathsf{G}]$ into $\mathsf{U}_m^{\not\perp}$, we can change random oracle G during the security proof such that the scheme is rendered perfectly correct, a necessary condition to proceed with the tight security proof. Distinguishing G from its "perfected" version allows for a reduction to a distinguishing problem. To generalize this strategy for *any* scheme, however, one would have to come up with a reduction that distinguishes access to an encryption oracle from access to an oracle that only answers with perfect encryptions, and as mentioned above, it might prove difficult to formalize this indistinguishability property in a meaningful manner such that it is compatible with the standard notion of IND-CCA security. We hope that our proofs achieve better auditability due to their at least somewhat more modular structure.

B On Comparing Security Models in Key Exchange

In the literature on key exchange one can find several widely-used game-based security models and many variants of them. All of these models have in common that they formalize the idea that the key that is computed by two parties should be indistinguishable from random to the attacker (and thus suitable for the application of symmetric primitives). When compared in more detail, AKE models typically differ in two ways. The first one is the set of capabilities they conceptually grant to the attacker to reveal secret values. In comparison to the classical works on key exchange, more recent models have extended the set of attacker queries over the past two decades considerably. These differences are cryptographically very meaningful. For example, a security model that allows the attacker to reveal the long-term key of the test session is stronger than one in which such an attack is excluded. Often new attack capabilities are given explicit names, e.g. attacks in which the attacker corrupts the secret key of the test session are typically called key compromise impersonation (KCI) attacks [34]. The second way in which security models differ is the concrete formalization of these attacks, i.e. the algorithmic steps the challenger (in the context of key exchange often called execution environment) has to perform to answer attack

[11] One could simply insert as the first game hop an abort if the key pair renders the scheme non-perfectly correct, thereby obtaining the upper bound $\delta \ll 4q_E\sqrt{\delta}$.

queries. This not only includes the final computation of responses to attack queries but also the sometimes complicated bookkeeping operations required to keep track of which secret values have already been revealed to exclude trivial attacks. In particular, two formalizations of the same attack concept, say KCI attacks, may differ considerably. Unfortunately, in the literature these formalizations are often very vague and present rather informal descriptions of how attacker queries are handled. In cryptography, such an approach is very problematic since it lacks preciseness and allows for misinterpretations. We stress that in cryptography unspecified subtleties can make a huge difference in the expressiveness of cryptographic definitions, as for example, shown in[9] for definitions of chosen-ciphertext security. This work introduces two security models. Conceptually we do not introduce new attack queries but stick to the state-of-art in the field. We claim that our stronger model captures all attacks that are addressed in state-of-the-art security models. Our weaker model deviates from this by excluding attacks that reveal the ephemeral state of the test session in case it initiates the communication. Where our model excells is in the way we rigorously formalize the security definition: we opt for a precise pseudo-code representation of the security model as it is common in security models for cryptographic primitives. Let us go into more detail: as detailed before our stronger model is a formalization of a very strong security notion covering many advanced security features like weak PFS, KCI security, and security against reflection attacks [34]. Our weaker model is equivalent to that, except that it does not allow the attacker to obtain the (temporary) secret state that is held by initiator oracles in the time interval after the sending of the first message and before receiving the responder's message. (After obtaining the responder's message the session key is computed and all other state information may be erased.) In practice, the time interval for this attack is relatively small compared to the lifetime of the session key. However, an active attacker may increase it by withholding messages and delaying their arrival. We note that for practical reasons (for example to avoid denial of service attacks), in real-world implementations message delays cannot grow too large as otherwise the initiator will abort, assuming the receiver is not reachable. We also remark that the responder directly computes the session key after receiving the initiator's message, so formally no state is computed by the responder at all.

C CCA KEMs without disjoint simulatability.

Recall that transformation Punc punctures the message space at one message and samples encryptions of this message as fake encryptions, see Figure 4.

The following lemma states that $\mathsf{IND\text{-}CPA}$ security of PKE_0 implies DS security of PKE. Note that we do not specifiy ϵ_{dis} due to the following reason: While ϵ_{dis}-disjointness would follow naturally from injective encryption, this requirement might not be met by many schemes. We shift this issue to our proof of Theorem 2, in which we will achieve disjointness by switching G to a function that renders the encryption scheme perfectly correct and hence, injective.

Lemma C.1 (DS security of Punc). *If* PKE_0 *is* δ*-correct, so is* PKE. *For all adversaries* A*, there exists an* $\mathsf{IND\text{-}CPA}$ *adversary* B *such that*

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}) \ .$$

Proof. Let A be a DS adversary against PKE. Consider the games given in Figure 17.

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{A}) = |\Pr[G^{\mathsf{A}} \Rightarrow 1] - \frac{1}{2}| \ .$$

Consider the $\mathsf{IND\text{-}CPA}$ adversary $\mathsf{B} := (\mathsf{B}_1, \mathsf{B}_2)$ also given in Figure 17. Since B perfectly simulates game G,

$$|\Pr[G^{\mathsf{A}} \Rightarrow 1] - \frac{1}{2}| = \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{B}) \ .$$

The following lemma states that $\mathsf{IND\text{-}CPA}$ security of PKE_0 translates to $\mathsf{IND\text{-}CPA}$ security of PKE. Its proof is straightforward.

Lemma C.2 ($\mathsf{IND\text{-}CPA}$ security of Punc). *For all* $\mathsf{IND\text{-}CPA}$ *adversaries* A *there exists an adversary* B *such that*

$$\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}) \ .$$

Game G	$\mathsf{B}_1(pk)$
01 $pk \leftarrow \mathsf{KG}_0$	08 $m \leftarrow_\$ \mathcal{M}_0 \setminus \{\hat{m}\}$
02 $m \leftarrow_\$ \mathcal{M}_0 \setminus \{\hat{m}\}$	09 **return** $(m, \hat{m})$
03 $b \leftarrow_\$ \mathbb{F}_2$	
04 $c_0 \leftarrow \mathsf{Enc}_0(pk, m)$	$\mathsf{B}_2(c)$
05 $c_1 \leftarrow \mathsf{Enc}_0(pk, \hat{m})$	10 $b' \leftarrow \mathsf{A}(pk, c)$
06 $b' \leftarrow \mathsf{A}(pk, c_b)$	11 **return** b'
07 **return** $[\![b' = b]\!]$	

Figure 17: Game G and IND-CPA adversary $\mathsf{B} = (\mathsf{B}_1, \mathsf{B}_2)$ for the proof of Lemma 5.

The following corollary follows directly from Lemma 5, Lemma 6 and Lemma 1. It states that combining T with Punc turns IND-CPA security into DS security.

Corollary C.3 (DS security of TPunc). *For all adversaries* A *issuing at most* q_G *queries to* $|\mathsf{G}\rangle$*, there exist two adversaries* B_1 *and* B_2 *such that*

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0,\hat{m}],\mathsf{G}]}(\mathsf{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2 \cdot \sqrt{q_\mathsf{G} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4q_\mathsf{G}^2}{|\mathcal{M}| - 1}} \ ,$$

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

C.1 Proof of Theorem 2

The following theorem establishes that $\mathsf{FO}^{\not\perp}_m \circ \mathsf{Punc}$ turns IND-CPA security into IND-CCA security, in the quantum random oracle model, as long as PKE is γ-spread.

Theorem (CCA security of $\mathsf{FO}^{\not\perp}_m \circ \mathsf{Punc}$.). Assume PKE_0 to be δ-correct and γ-spread, and let $\hat{m} \in \mathcal{M}$. Let $\mathsf{KEM} := \mathsf{FO}^{\not\perp}_m[\mathsf{Punc}[\mathsf{PKE}, \hat{m}], \mathsf{G}, \mathsf{H}]$. Then, for any (quantum) IND-CCA adversary A issuing at most q_D (classical) queries to the decapsulation oracle DECAPS, at most q_H quantum queries to $|\mathsf{H}\rangle$, and at most q_G quantum queries to $|\mathsf{G}\rangle$, there exist (quantum) CPA adversaries B_1 and B_2 against PKE_0 such that

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) \leq\ & (8 \cdot (3 \cdot q_\mathsf{G} + 2 \cdot q_\mathsf{H} + q_D + 6)^2 + 1) \cdot \delta + \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) \\ & + 2 \cdot \sqrt{(q_\mathsf{G} + q_\mathsf{H}) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4(q_\mathsf{G} + q_\mathsf{H})^2}{|\mathcal{M}| - 1}} + 2^{-\gamma} \ ,\end{aligned}$$

and the running time of B_1 and B_2 is about that of A.

Proof. Let A be an adversary against the IND-CCA security of KEM, issuing at most q_D queries to DECAPS, at most q_H queries to the quantum random oracle $|\mathsf{H}\rangle$, and at most q_G queries to the quantum random oracle $|\mathsf{G}\rangle$. It is easy to verify that we can apply the first 5 game-hops of our proof of Theorem 1 to $\mathsf{FO}^{\not\perp}_m[\mathsf{Punc}[\mathsf{PKE}, \hat{m}], \mathsf{G}, \mathsf{H}]$: We first enforce that no decryption failure will occur by replacing G with oracle $\mathsf{G}_{pk,sk}$ that only samples from good randomness. Note that since PKE_0 is δ-correct, so is $\mathsf{Punc}[\mathsf{PKE}, \hat{m}]$. Afterwards, we plug encryption into the random oracle, and then change oracle DECAPS such that it always returns $K := \mathsf{H}_\mathsf{q}(c)$, as opposed to implicitly rejecting whenever decryption or reencryption fails. Both changes are not recognizable by A. After evening out the decapsulation oracle, we switch back to using random oracle G and obtain the upper bound

$$\mathrm{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{KEM}}(\mathsf{A}) \leq |\Pr[G^\mathsf{A}_4 \Rightarrow 1] - 1/2| + 8 \cdot (2 \cdot q_\mathsf{G} + q_\mathsf{H} + q_D + 4)^2 \cdot \delta \ .$$

Since for $\mathsf{PKE} = \mathsf{Punc}[\mathsf{PKE}, \hat{m}]$, our fake encryptions are encryptions of $\hat{m}$, we next replace the challenge ciphertext c^* with an encryption of $\hat{m}$. We know that there exists an adversary C_DS against the disjoint simulatability of $\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0, \hat{m}], \mathsf{G}]$ such that

$$|\Pr[G^\mathsf{A}_4 = 1] - \Pr[G^\mathsf{A}_5 = 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0,\hat{m}],\mathsf{G}]}(\mathsf{C}_\mathsf{DS}), \ ,$$

and according to Corollary 1, there exist CPA adversaries B_1 and B_2 against PKE_0 such that

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0,\hat{m}],\mathsf{G}]}(\mathsf{C}_{\mathsf{DS}}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2\cdot\sqrt{(q_{\mathsf{G}}+q_{\mathsf{H}})\cdot\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4(q_{\mathsf{G}}+q_{\mathsf{H}})^2}{|\mathcal{M}|-1}} \ .$$

GAMES G_5 - G_8

01 $(pk, sk) \leftarrow \mathsf{KG}$
02 Pick $2q$-wise hash f // G_7 - G_8
03 $\mathsf{G} := \mathsf{G}_{pk,sk}$ // G_7 - G_8
04 $\mathsf{H} := \mathsf{H}_{\mathsf{q}}(\mathsf{Enc}_0(pk, -; \mathsf{G}(-)))$
05 $b \leftarrow_{\$} \mathbb{F}_2$
06 $m^* \leftarrow \mathcal{M} \setminus \{\hat{m}\}$
07 $c^* \leftarrow \mathsf{Enc}_0(pk, \hat{m})$
08 **if** $\mathsf{Enc}_0(pk, \hat{m}; \mathsf{G}(\hat{m})) = c^*$
09 ABORT // G_6 - G_8
10 $K_0^* := \mathsf{H}_{\mathsf{q}}(c^*)$ // G_5 - G_7
11 $K_0^* \leftarrow_{\$} \mathcal{K}$ // G_8
12 $K_1^* \leftarrow_{\$} \mathcal{K}$
13 $b' \leftarrow \mathsf{A}^{\textsc{Decaps},|\mathsf{H}\rangle,|\mathsf{G}\rangle}(pk, c^*, K_b^*)$
14 **return** $[\![b' = b]\!]$

$\textsc{Decaps}(c \neq c^*)$
15 **return** $K := \mathsf{H}_{\mathsf{q}}(c)$

$\mathsf{G}_{pk,sk}(m)$
16 $r := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
17 **return** r

Figure 18: Games G_5 - G_8 for the proof of Theorem 2.

To justify that we can replace the real key with random, we give a sequence of games in Figure 18.

Game G_6. In game G_6, we abort in line 09 if the deterministic encryption hits the challenge ciphertext, i.e., if $\mathsf{Enc}_0(pk, \hat{m}; \mathsf{G}(\hat{m}) = c^*$. Since G is a random oracle, and PKE is γ-spread,

$$|\Pr[G_5^{\mathsf{A}} \Rightarrow 1] - \Pr[G_6^{\mathsf{A}} \Rightarrow 1]| \leq 2^{-\gamma} \ .$$

Game G_7. In game G_7, we enforce that no decryption failure will occur once more: again, we switch to $\mathsf{G}_{pk,sk}$. With the same argument as for former game-hops,

$$|\Pr[G_6^{\mathsf{A}} \Rightarrow 1] - \Pr[G_7^{\mathsf{A}} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{C}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}}_{\lambda,0} = 1]| \ ,$$

where C is given in Figure 19, and according to Lemma 4,

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

Game G_8. In game G_8, 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_8,

$$\Pr[G_8^{\mathsf{A}} \Rightarrow 1] = 1/2 \ .$$

It remains to upper bound $|\Pr[G_7^{\mathsf{A}} \Rightarrow 1] - \Pr[G_8^{\mathsf{A}} \Rightarrow 1]|$. To this end, it is sufficient to upper bound the probability that any of the queries to $|\mathsf{H}_{\mathsf{q}}\rangle$ could possibly contain c^*. Each query to $|\mathsf{H}_{\mathsf{q}}\rangle$ is either a classical query, triggered by A querying $\textsc{Decaps}$ on some ciphertext c, or a query in superposition, triggered by A querying $|\mathsf{H}\rangle$. Since queries to $\textsc{Decaps}$ on c^* are explicitly forbidden, the only possibility would be one of A's queries to $|\mathsf{H}\rangle$. A's queries to $|\mathsf{H}\rangle$ trigger queries to $|\mathsf{H}_{\mathsf{q}}\rangle$ that are of the form $\sum_m \alpha_m|\mathsf{Enc}_0(pk, m; \mathsf{G}(m))\rangle$. They cannot contain c^* unless there exists some message m such that $\mathsf{Enc}_0(pk, m; \mathsf{G}(m)) = c^*$. We claim that this is impossible. First we consider the case that $m = \hat{m}$: It would be required that $\mathsf{Enc}_0(pk, \hat{m}; \mathsf{G}(\hat{m})) = c^*$, but the game aborts if this ever should be the case. It remains to show that no other message could possibly encrypt to c^* unless c^* induced decryption failure: Assume that there exists some message $m \neq \hat{m}$ such that $\mathsf{Enc}_0(pk, m; \mathsf{G}(m)) = c^*$. Since all sampled randomness is good, it is implied that $\hat{m} \neq m = \mathsf{Dec}_0(sk, c^*)$. Since c^* was a random encryption of $\hat{m}$, the probability of c^* inducing decryption failure can be upper bounded by δ, hence

$$|\Pr[G_7^{\mathsf{A}} \Rightarrow 1] - \Pr[G_8^{\mathsf{A}} \Rightarrow 1]| \leq \delta \ .$$

Collecting the probabilities yields the theorem's upper bound.

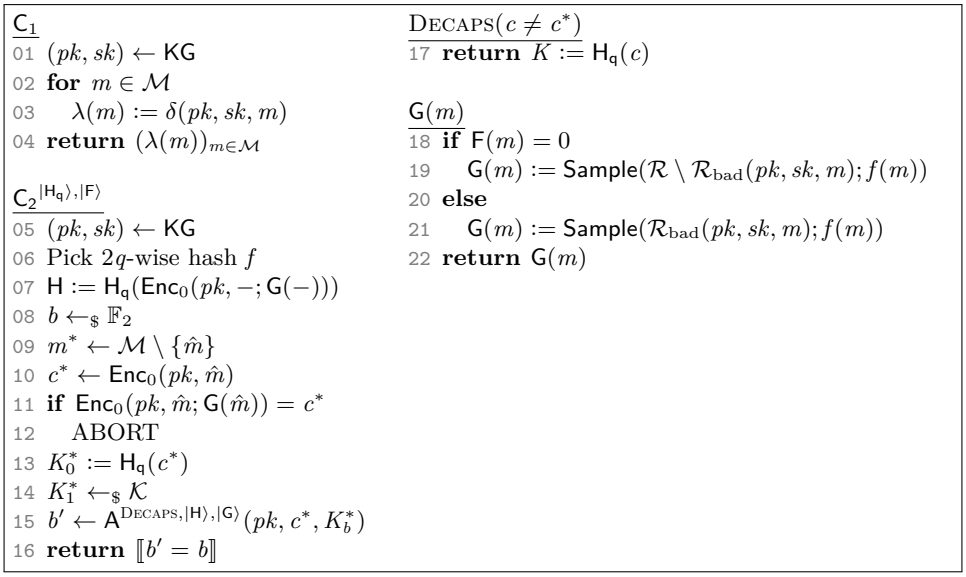


Figure 19: Adversary C executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ for the proof of Theorem 2.

D Proof of Lemma 2

Faithful execution of the protocol ($\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$). Recall that we are proving an upper bound for $|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]|$. First, we will enforce that indeed, we only need to consider the case where $\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$, afterwards we ensure that exactly one matching session exists. Consider the sequence of games given in Figure 20.

Games $G_{0,b}$. Since for both bits b, game $G_{0,b}$ is the original game $\mathsf{IND\text{-}StAA}_b$,

$$\begin{aligned}&|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]|\\ &\quad = |\Pr[G_{0,1}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[G_{0,0}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]| \ .\end{aligned}$$

Games $G_{1,b}$. Both games $G_{1,b}$ abort in line 07 if $\mathfrak{M}(\mathrm{sID}^*) = \varnothing$. Since $\Pr[G_{0,b}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] = \Pr[G_{1,b}^{\mathsf{B}} \Rightarrow 1]$ for both bits b,

$$\begin{aligned}&|\Pr[G_{0,1}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[G_{0,0}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]|\\ &\quad = |\Pr[G_{1,1}^{\mathsf{B}} \Rightarrow 1] - \Pr[G_{1,0}^{\mathsf{B}} \Rightarrow 1]| \ .\end{aligned}$$

Games $G_{2,b}$. Both games $G_{2,b}$ abort in line 09 if $|\mathfrak{M}(\mathrm{sID}^*)| > 1$, i.e., if more than one matching session exists. Due to the difference lemma,

$$|\Pr[G_{1,b}^{\mathsf{B}} \Rightarrow 1] - \Pr[G_{2,b}^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{Abort in line 09}]$$

for both bits b. We claim

$$\Pr[\text{Abort in line 09}] \leq (S-1) \cdot \mu(\mathsf{Enc}) \cdot \max\{\mu(\mathsf{Enc}), \mu(\mathsf{KG})\} \leq S \cdot \mu(\mathsf{Enc}) \ .$$

To verify this bound, we first consider the case that role[sID*] = "initiator": Let $i^* := \mathrm{holder}[\mathrm{sID}^*]$ and $j^* := \mathrm{peer}[\mathrm{sID}^*]$. To create more than one matching session, B has to establish and derive two distinct ("responder") sessions $\mathrm{sID} \neq \mathrm{sID}'$ with holder j^* and peer i^* via oracle call to $\mathrm{DER}_{\mathrm{resp}}$, such that $\mathrm{sent}[\mathrm{sID}] = \mathrm{sent}[\mathrm{sID}']$. This means that for $(c_i, \tilde{c}) := \mathrm{sent}[\mathrm{sID}]$ and $(c_i', \tilde{c}') := \mathrm{sent}[\mathrm{sID}']$ it holds that both $c_i = c_i'$ and $\tilde{c} = \tilde{c}'$. All ciphertexts were generated by faithfully executing $\mathrm{DER}_{\mathrm{resp}}$, and therefore encryptions of messages that were drawn at random. Since both $\tilde{pk}$ and pk_{j^*} were also generated faithfully,

$$\Pr[\text{Abort in line 09} \ \wedge \ \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}] \leq (S-1) \cdot \mu(\mathsf{Enc})^2 \leq S \cdot \mu(\mathsf{Enc}) \ .$$

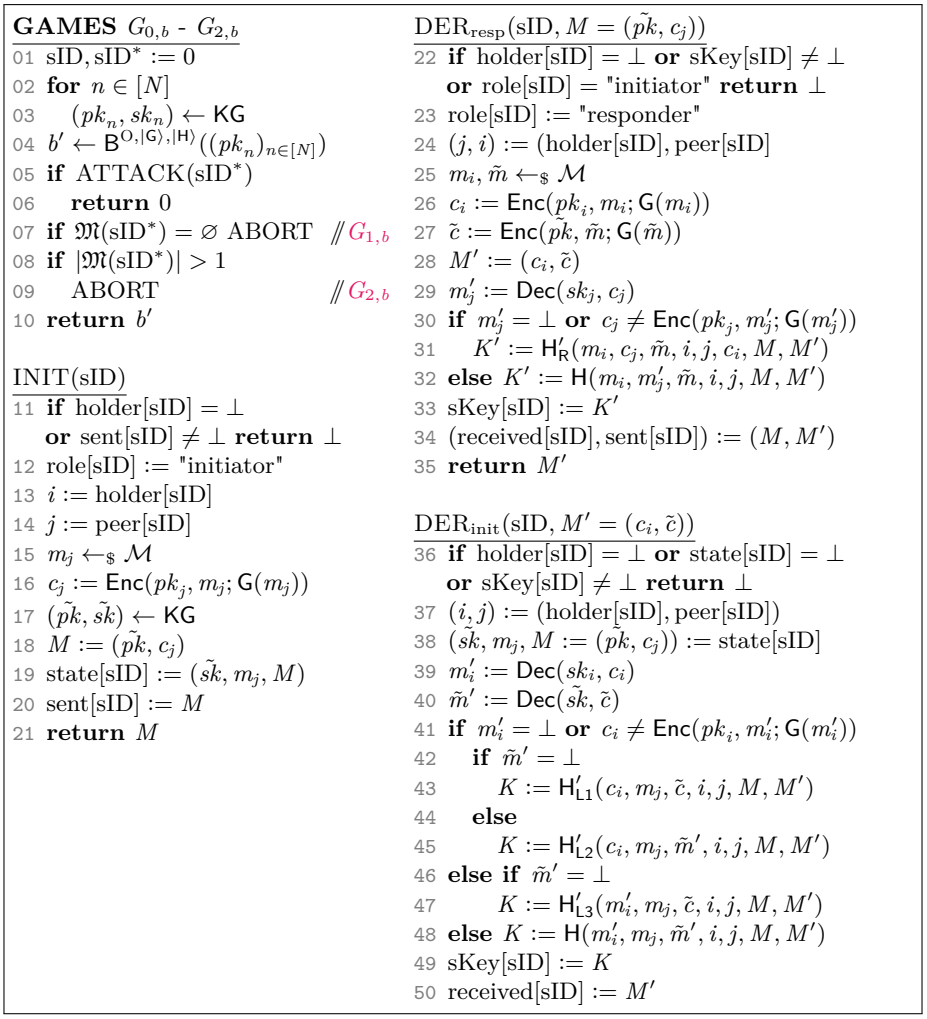


Figure 20: Games $G_{0,b}$ - $G_{2,b}$ for case one of the proof of Theorem 3. Helper procedure ATTACK and oracles TEST, EST, CORRUPT, REVEAL and REV-STATE remains as in the original IND-StAA game (see Figures 12 and 13).

Now we consider the case that role[sID^*] = "responder": Let $j^* :=$ holder[sID^*] and $i^* :=$ peer[sID^*]. To create more than one matching session, B has to establish and derive two distinct sessions $\mathrm{sID} \neq \mathrm{sID}'$ with holder i^* and peer j^* via oracle call to INIT such that sent[sID] = sent[sID']. This means that for $(\tilde{pk}, c_j) :=$ sent[sID] and $(\tilde{pk}', c'_j) :=$ sent[sID'] it holds that both $\tilde{pk} = \tilde{pk}'$ and $c_j = c'_j$. Both public keys and both ciphertexts were generated by faithfully executing INIT, the latter therefore being encryptions of messages that were drawn at random, and

$$\Pr[\text{Abort in line 09} \ \wedge \ \text{role}[\mathrm{sID}^*] = \text{"responder"}] \leq (S-1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}).$$

So far, we established

$$\begin{aligned} &|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]| \\ &\quad \leq |\Pr[G_{2,1}^{\mathsf{B}} \Rightarrow 1] - \Pr[G_{2,0}^{\mathsf{B}} \Rightarrow 1]| + 2 \cdot S \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) \ . \end{aligned}$$

Since games $G_{2,b}$ abort unless $|\mathfrak{M}(\mathrm{sID}^*)| = 1$, we treat the matching session's ID as unique from this point on and denote it by sID'. Note that it is ensured that one of the two sessions was executed as a "initiator" session, while the other was executed as a "responder" session. Let $\mathrm{sID}^*_{\mathrm{init}}$ denote the "initiator"

session, i.e., pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ such that $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] = \text{"initiator"}$, and let $\mathrm{sID}^*_{\mathrm{resp}}$ denote the other session. B's bit b' only counts in $\mathsf{IND\text{-}StAA}_b$ (and also in $G_{2,b}$) if no trivial attack was executed: ATTACK returns **true** (and hence the game returns 0) if B did obtain both the initialising session's internal state and the secret key of its holder. We will therefore examine

- case ($\neg$st): the case that the initialising session's state was not revealed, i.e., the case that $\neg\mathrm{revState}[\mathrm{sID}^*_{\mathrm{init}}]$,
- and case ($\neg sk$): the case that the initialising session's holder was not corrupted, i.e., the case that $\neg\mathrm{corrupted}[\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]]$

Since cases ($\neg$st) and ($\neg sk$) are mutually exclusive if the game outputs 1,

$$\begin{aligned}|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1]| \leq\ & |\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \\ & + |\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \ .\end{aligned}$$

Case ($\neg$st). We claim that there exists an adversary $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}$ such that

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \\ &\qquad \leq 2S^2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}) + 32 \cdot S \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 1)^2 \cdot \delta \\ &\qquad\quad 2S^2 \cdot \epsilon_{\mathrm{dis}} + 2S^2 \cdot \mu(\mathsf{KG}) \ . \end{aligned} \tag{3}$$

The proof is given in in Appendix E.1. Its main idea is that since the initialising session's state (in particular, ephemeral secret key $\tilde{sk}^*$) remains unrevealed throughout the game, at least message $\tilde{m}^*$ (that was randomly picked by $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*_{\mathrm{resp}})$ cannot be computed trivially. By patching encryption into the random oracle at the argument where the ephemeral messages go in, we ensure that the game makes no use of $\tilde{sk}^*$ any longer. Since PKE is DS (and hence, so is $\mathsf{T}[\mathsf{PKE},\mathsf{G}]$, see Lemma 1), we can decouple the test session's key from $\tilde{m}^*$ by replacing $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$, and changing the key accordingly. Given that PKE is ϵ_{dis}-disjoint, the probability that this fake ciphertext is a proper encryption can be upper bounded by ϵ_{dis}. Since the random oracle now comes with patched-in encryption, ϵ_{dis} also serves as an upper bound for the probability that a random oracle query actually hits the session key. Hence the key is indistinguishable from a random key with overwhelming probability.

Case ($\neg sk$). We claim that there exists an adversary $\mathsf{A}^{\neg sk}_{\mathsf{DS}}$ such that

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\ &\qquad \leq\ 2 \cdot SN \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg sk}_{\mathsf{DS}}) + 32N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \\ &\qquad + 2 \cdot SN \cdot \epsilon_{\mathrm{dis}} + S^2 \cdot N \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) \ . \end{aligned} \tag{4}$$

The proof of the upper bound is given in in Appendix E.2. Structurally, the proof is the same. It differs in the following way: while in case ($\neg$st), we made use of the fact that B does not obtain ephemeral secret key $\tilde{sk}^*$ and therefore, ciphertext $\tilde{c}$ was indistinguishable from a fake encryption, in case ($\neg sk$), we can replace ciphertext c_i (since $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]$ is not corrupted).

Collecting the probabilities, and folding $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}$ and $\mathsf{A}^{\neg sk}_{\mathsf{DS}}$ into one adversary A, we obtain

$$\begin{aligned}&|\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]| \\ &\quad \leq 2 \cdot S \cdot (S + N) \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}) + 32 \cdot (S + N) \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S + 1)^2 \cdot \delta \\ &\qquad + 4 \cdot S^2 \cdot \epsilon_{\mathrm{dis}} + S^2 \cdot (N + 1) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + 2 \cdot S^2 \cdot \mu(\mathsf{KG}) \ , \end{aligned}$$

the upper bound given in Lemma 2.

D.1 Case ($\neg$st) of the Proof of Lemma 2

Case ($\neg$st) (Initialising session's state was not revealed). Consider the sequence of games given in Figures 21, 22 and 24: First, we will enforce that indeed, we only need to consider the case where

$\neg$revState[sID$^*_{\text{init}}$]. Afterwards, we ensure that the game makes no use of ephemeral secret key $\tilde{sk}^*$ of sID$^*_{\text{init}}$ any longer by patching encryption into the random oracle (in games $G^{\neg\text{st}}_{2,b}$ to $G^{\neg\text{st}}_{10,b}$, see Figure 21 and22). Next, during execution of DER$_{\text{resp}}$(sID$^*_{\text{resp}}$), we replace $\tilde{c} = \mathsf{Enc}(\tilde{pk}^*, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G^{\neg\text{st}}_{11,b}$ to $G^{\neg\text{st}}_{12,b}$, Figure 24, see line 25). We show that after those changes, B's view does not change with overwhelming probability if we change TEST such that it always returns a random value (game $G^{\neg\text{st}}_{14,0}$, also Figure 24).

GAMES $G^{\neg\text{st}}_{2,b}$ - $G^{\neg\text{st}}_{6,b}$

01 cnt, sID* := 0
02 $s'_{\text{init}} \leftarrow_{\$} [S]$ // $G^{\neg\text{st}}_{4,b}$ - $G^{\neg\text{st}}_{6,b}$
03 **for** $n \in [N]$
04 $(pk_n, sk_n) \leftarrow \mathsf{KG}$
05 $(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \mathsf{KG}$ // $G^{\neg\text{st}}_{5,b}$ - $G^{\neg\text{st}}_{6,b}$
06 $b' \leftarrow \mathsf{B}^{\mathsf{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
07 **if** ATTACK(sID*)
08 **return** 0
09 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
10 **if** revState[sID$^*_{\text{init}}$] ABORT // $G^{\neg\text{st}}_{3,b}$ - $G^{\neg\text{st}}_{6,b}$
11 Pick sID$^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$ s. th.
role[sID$^*_{\text{init}}$] = "initiator" // $G^{\neg\text{st}}_{4,b}$ - $G^{\neg\text{st}}_{6,b}$
12 **if** sID$^*_{\text{init}} \neq s'_{\text{init}}$
13 **return** 0 // $G^{\neg\text{st}}_{4,b}$ - $G^{\neg\text{st}}_{6,b}$
14 **return** b'

INIT(sID)

15 **if** holder[sID] = $\perp$
or sent[sID] $\neq \perp$ **return** $\perp$
16 role[sID] := "initiator"
17 i := holder[sID]
18 j := peer[sID]
19 $m_j \leftarrow_{\$} \mathcal{M}$
20 $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
21 $(\tilde{pk}, \tilde{sk}) \leftarrow \mathsf{KG}$
22 **if** sID $\neq s'_{\text{init}}$ **and** $\tilde{pk} = \tilde{pk}^*$
23 ABORT // $G^{\neg\text{st}}_{6,b}$
24 **if** sID = s'_{init}
25 $(\tilde{pk}, \tilde{sk}) := (\tilde{pk}^*, \tilde{sk}^*)$ // $G^{\neg\text{st}}_{5,b}$ - $G^{\neg\text{st}}_{6,b}$
26 $M := (\tilde{pk}, c_j)$
27 state[sID] := $(\tilde{sk}, m_j, M)$
28 sent[sID] := M
29 **return** M

Figure 21: Games $G^{\neg\text{st}}_{2,b}$ - $G^{\neg\text{st}}_{6,b}$ for case ($\neg$st) of the proof of Lemma 2. Oracles DER$_{\text{resp}}$, DER$_{\text{init}}$ and TEST remain as in games $G^{\neg\text{st}}_{0,b}$ (see Figure 20, page 39), and helper procedure ATTACK and oracles EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 12 and Figure 13, pages 21 and 22).

GAMES $G^{\neg\text{st}}_{2,b}$. Since game $G^{\neg\text{st}}_{2,b}$ and $G_{2,b}$ are the same for both bits b,

$$|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\text{st}]| = |\Pr[G_{2,1}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G_{2,0}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1 \wedge \neg\text{st}]| \ .$$

GAMES $G^{\neg\text{st}}_{3,b}$. To enforce that we are in the correct case, games $G^{\neg\text{st}}_{3,b}$, abort in line 10 if revState[sID$^*_{\text{init}}$]. Since for both bits b it holds that $\Pr[G^{\mathsf{B}}_{3,b} \Rightarrow 1] = \Pr[G^{\mathsf{B}}_{2,b} \Rightarrow 1 \wedge \neg\text{st}]$,

$$|\Pr[G_{2,1}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G_{2,0}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1 \wedge \neg\text{st}]| = |\Pr[G_{3,1}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{3,0}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1]| \ .$$

As mentioned above, the first goal is not make use of the ephemeral secret key of sID$^*_{\text{init}}$ any longer. To this end, we first have to add a guess for sID$^*_{\text{init}}$.

GAMES $G^{\neg\text{st}}_{4,b}$. In both games $G^{\neg\text{st}}_{4,b}$, one of the sessions that get established during execution of B is picked at random in line 02, and the games return 0 in line 13 if any other session s'_{init} was picked than session sID$^*_{\text{init}}$. Since games $G^{\neg\text{st}}_{4,b}$ and $G^{\neg\text{st}}_{3,b}$ proceed identically for both bits b if $s'_{\text{init}} = \text{sID}^*_{\text{init}}$, and since games $G^{\neg\text{st}}_{4,b}$ output 0 if $s'_{\text{init}} \neq \text{sID}^*_{\text{init}}$,

$$\Pr[G_{3,b}{}^{\neg\text{st}\,\mathsf{B}} \Rightarrow 1] = S \cdot \Pr[G^{\neg\text{st}}_{4,b} \Rightarrow 1] \ .$$

GAMES $G^{\neg\text{st}}_{5,b}$. In both games $G^{\neg\text{st}}_{5,b}$, an ephemeral key pair $(\tilde{pk}^*, \tilde{sk}^*)$ gets drawn in line 05 and oracle INIT is changed in line 25 such that this key pair is used as the ephemeral key pair of sID$^*_{\text{init}}$. This change is only conceptual, hence

$$\Pr[G^{\neg\text{st}}_{4,b} \Rightarrow 1] = \Pr[G^{\neg\text{st}}_{5,b} \Rightarrow 1] \ .$$

GAMES $G^{\neg\mathrm{st}}_{6,b}$. Both games $G^{\neg\mathrm{st}}_{6,b}$, abort in line 23 if any of the initialised sessions apart from $\mathrm{sID}^*_{\mathrm{init}}$ comes up with the same ephemeral key $\tilde{pk}^*$.

$$|\Pr[G^{\neg\mathrm{st}}_{5,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{6,b} \Rightarrow 1]| \leq (S-1) \cdot \mu(\mathsf{KG}) \ .$$

So far, we established

$$\begin{aligned}|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \\ \leq S \cdot |\Pr[G^{\neg\mathrm{st}^{\mathsf{B}}}_{6,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}^{\mathsf{B}}}_{6,0} \Rightarrow 1]| + 2S^2 \cdot \mu(\mathsf{KG}) \ .\end{aligned}$$

Consider the sequence of games given in Figure 22. The goal is to change the game such that it can be simulated without usage of $\tilde{sk}^*$. As in the KEM proof, we will first modify random oracle G such that it renders PKE perfectly correct for key pair $(\tilde{pk}^*, \tilde{sk}^*)$.

GAME $G^{\neg\mathrm{st}}_{7,b}$. In game $G^{\neg\mathrm{st}}_{7,b}$, we enforce that no decryption failure with respect to key pair $(\tilde{pk}^*, \tilde{sk}^*)$ will occur: We replace random oracle G with $\mathsf{G}_{\tilde{pk}^*,\tilde{sk}^*}$ in line 08, where $\mathsf{G}_{\tilde{pk}^*,\tilde{sk}^*}(m)$ is defined in line 55 by

$$\mathsf{G}_{\tilde{pk}^*,\tilde{sk}^*}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(\tilde{pk}^*, \tilde{sk}^*, m); f(m)) \ ,$$

with $\mathcal{R}_{\mathrm{bad}}(pk, sk, m) := \{r \in \mathcal{R} \mid \mathsf{Dec}(sk, \mathsf{Enc}(pk, m; r)) \neq m\}$ denoting the set of "bad" randomness for any fixed key pair (pk, sk), and any message $m \in \mathcal{M}$. Further, let

$$\delta(pk, sk, m) := {}^{|\mathcal{R}_{\mathrm{bad}}(pk,sk,m)|}/_{|\mathcal{R}|} \tag{5}$$

denote 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 $|\Pr[G^{\neg\mathrm{st}}_{6,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{7,b} \Rightarrow 1]|$ for each bit b, we construct (unbounded, quantum) adversaries C^{b} against the generic distinguishing problem with bounded probabilities GDPB_{λ} (see Lemma 4) in Figure 23, issuing at most $q_{\mathsf{G}} + 3 \cdot S$ queries to $|\mathsf{F}\rangle$: Each C^{b} runs $(pk, sk) \leftarrow \mathsf{KG}$ and uses this key pair as $(\tilde{pk}^*, \tilde{sk}^*)$ when simulating game $G^{\neg\mathrm{st}}_{6,b}$. C^{b} 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 C^{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 distributed according to $B_{\lambda(m)} = B_{\delta(pk,sk,m)}$. By construction, the random variable $\mathsf{G}(m)$ defined in line 06 if $\mathsf{F}(m) = 0$ and in line 08 if $\mathsf{F}(m) = 1$ is uniformly distributed in $\mathcal{R}$. Therefore, G is a (quantum) random oracle, and C^{b} perfectly simulates game $G^{\neg\mathrm{st}}_{6,b}$ if executed in game $\mathsf{GDPB}_{\lambda,1}$. Since adversary C^{b} also perfectly simulates game $G^{\neg\mathrm{st}}_{7,b}$ if executed in game $\mathsf{GDPB}_{\lambda,0}$,

$$|\Pr[G^{\neg\mathrm{st}}_{6,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{7,b} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \ ,$$

and according to Lemma 4,

$$\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \leq 8 \cdot (q_{\mathsf{G}} + 3 \cdot S + 1)^2 \cdot \delta \ .$$

Recall that the goal is to simulate the game without knowledge of $\tilde{sk}^*$. To this end, we will first change $\mathrm{DER}_{\mathrm{init}}$ for s'_{init} as follows: If ciphertext c_i already induces de- or reencryption failure, we will not have to use $\tilde{sk}^*$ any more to check whether $\tilde{c}$ induces decryption failure as well.

GAMES $G^{\neg\mathrm{st}}_{8,b}$. In games $G^{\neg\mathrm{st}}_{8,b}$, oracle $\mathrm{DER}_{\mathrm{init}}$ is changed for our guess s'_{init} in line 45: Whenever decryption or reencryption fails with respect to c_i, the session key is defined as $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$, as opposed to letting $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', i, j, M, M')$ in the case that decryption or reencryption fails with respect to c_i, but $\tilde{c}$ de- and reencrypts properly.

We claim that this change does not affect B's view since the respective random value is decoupled from all other session keys: Let i^* and j^* denote holder and peer of s'_{init}. Recall that $\mathrm{DER}_{\mathrm{init}}(s'_{\mathrm{init}})$ uses ephemeral key pair $(\tilde{pk}^*, \tilde{sk}^*)$. Furthermore, let m^*_j be the message that was picked by $\mathrm{INIT}(s'_{\mathrm{init}})$, let $(c^*_i, \tilde{c}^*)$ denote the message that is received by $\mathrm{DER}_{\mathrm{init}}(s'_{\mathrm{init}})$, and let $\tilde{m}^* := \mathsf{Dec}(\tilde{sk}^*, \tilde{c}^*)$. To distinguish the games, both values $\mathsf{H}'_{\mathsf{L1}}(c^*_i, m^*_j, \tilde{c}^*, i^*, j^*, M, M')$ and $\mathsf{H}'_{\mathsf{L2}}(c^*_i, m^*_j, \tilde{m}^*, \tilde{pk}^*, i^*, j^*, M, M')$ must be obtained. But both $\mathsf{H}'_{\mathsf{L1}}$ and $\mathsf{H}'_{\mathsf{L2}}$ are internal random oracles that cannot be accessed directly

Figure 22: Games $G^{\neg st}_{6,b}$ - $G^{\neg st}_{10,b}$ for case ($\neg$st) of the proof of Lemma 2. Oracle Init remains as in games $G^{\neg st}_{4,b}$ (see Figure 21, page 42), (see Figure 12, page 21), and helper procedure ATTACK and oracles TEST, EST, REVEAL and REV-STATE remain as in the original IND-StAA games. f (lines 07 and 55) is an internal $2q$-wise independent hash function, where $q := q_{\mathsf{G}} + q_{\mathsf{H}} + S$, that cannot be accessed by B. $\mathsf{Sample}(Y)$ 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 $f(m)$.

by B. The only way to obtain oracle values of $\mathsf{H}'_{\mathsf{L1}}$ and $\mathsf{H}'_{\mathsf{L2}}$ is via calls to REVEAL after execution of DER_{init}, and possibly, via the additional call to TEST for the test session. (Note that the latter is only an option if the test session is an "initiator" session, and if either decryption or reencryption fails with respect to c^*_i. In this case, the test session and its match do not derive the same key.) Recall that the game trivially outputs 0 if B queries REVEAL on sID^*_{init} or if $\text{sID}^*_{init} \neq s'_{init}$. Therefore, to distinguish $\mathsf{H}'_{\mathsf{L1}}(c^*_i, m^*_j, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c^*_j), M')$ from $\mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}^*, i, j, (\tilde{pk}^*, c^*_j), M')$ without triggering the game to output 0, another session $s \neq s'_{init}$ would have to be established and initialized, and it would be necessary that the same ephemeral public key $\tilde{pk}^*$ was drawn by INIT(s). But recall that since game $G^{\neg st}_{6,b}$, it is enforced that $\tilde{pk}^*$ is not used as the ephemeral key of any other session than s'_{init} (see line 23 in Figure 21). Hence, B cannot obtain both values without losing trivially. Since both values are uniformly

$\underline{\mathsf{C}_1^{\mathsf{b}} = \mathsf{D}_1^{\mathsf{b}}}$
01 $(pk, sk) \leftarrow \mathsf{KG}$
02 **for** $m \in \mathcal{M}$
03 $\quad \lambda(m) := \delta(pk, sk, m)$
04 **return** $(\lambda(m))_{m\in\mathcal{M}}$

$\underline{\mathsf{G}(m)}$
05 **if** $\mathsf{F}(m) = 0$
06 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
07 **else**
08 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
09 **return** $\mathsf{G}(m)$

$\underline{\mathsf{C}_2^{\mathsf{b}|\mathsf{F}\rangle},\ \mathsf{D}_2^{\mathsf{b}|\mathsf{F}\rangle}}$
10 $\mathrm{cnt}, \mathrm{sID}^* := 0$
11 $s'_{\mathrm{init}} \leftarrow_{\$} [S]$
12 **for** $n \in [N]$
13 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
14 $(\tilde{pk}^*, \tilde{sk}^*) := (pk, sk)$
15 Pick $2q$-wise hash f
16 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
17 **if** ATTACK(sID^*)
18 $\quad$ **return** 0
19 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
20 **if** revState[$\mathrm{sID}^*_{\mathrm{init}}$] ABORT
21 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. role[$\mathrm{sID}^*_{\mathrm{init}}$] = "initiator"
22 **if** $\mathrm{sID}^*_{\mathrm{init}} \neq s'_{\mathrm{init}}$ **return** 0
23 **return** b'

Figure 23: Unbounded quantum adversaries $\mathsf{C}^{\mathsf{b}} = (\mathsf{C}_1^{\mathsf{b}}, \mathsf{C}_2^{\mathsf{b}})$ and $\mathsf{D}^{\mathsf{b}}_{=}(\mathsf{D}_1^{\mathsf{b}}, \mathsf{D}_2^{\mathsf{b}})$ for $b \in \mathbb{F}_2$, executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ with access to $|\mathsf{F}\rangle$, for case ($\neg$st) of the proof of Lemma 2. $\delta(pk, sk)$ is defined in Equation (5). The adversaries only differ in their definition of $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$ and H: For adversaries C^{b}, $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$ and H are defined as in game $G^{\neg\mathrm{st}}_{6,b}$, see Figure 22, while for adversaries D^{b}, $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$ and H are defined as in game $G^{\neg\mathrm{st}}_{9,b}$ (also Figure 22).

random, B's view does not change and

$$\Pr[G^{\neg\mathrm{st}}_{7,b} \Rightarrow 1] = \Pr[G^{\neg\mathrm{st}}_{8,b} \Rightarrow 1] \ .$$

We can now get rid of $\tilde{sk}^*$ altogether by changing $\mathrm{DER}_{\mathrm{init}}$ for s'_{init} such that $\tilde{sk}^*$ is not used any more even if ciphertext c_i decrypts correctly. This is achieved as follows: If ciphertext c_i decrypts correctly, we do note use the decryption of $\tilde{c}$, but $\tilde{c}$ itself. To this end, we will "plug in" encryption into random oracle H whenever ephemeral public key $\tilde{pk}^*$ is used. To maintain consistency, $\mathrm{DER}_{\mathrm{resp}}$ is changed accordingly.

Games $G^{\neg\mathrm{st}}_{9,b}$. In game $G^{\neg\mathrm{st}}_{9,b}$, random oracle H is changed as follows: Instead of picking H uniformly random, we pick two random oracles H_{q} and H' and define

$$\mathsf{H}(m_1, m_2, m_3, i, j, M = (\tilde{pk}, c_j), M') := \begin{cases} \mathsf{H}_{\mathsf{q}}(m_1, m_2, \mathsf{Enc}(\tilde{pk}, m_3; \mathsf{G}(m_3)), i, j, M, M') & \tilde{pk} = \tilde{pk}^* \\ \mathsf{H}'(m_1, m_2, m_3, i, j, M, M') & \text{o.w.} \end{cases},$$

see line 58. Note that since G only samples from good randomness, encryption is rendered perfectly correct and hence, injective. Since encryption is injective, H still is uniformly random.

We make the change of H explicit in the derivation oracles: We change $\mathrm{DER}_{\mathrm{init}}$ in line 52 such that for $\mathrm{sID} = s'_{\mathrm{init}}$, the session key is defined as $K := \mathsf{H}_{\mathsf{q}}(m'_i, m_j, \tilde{c}, i, j, M, M')$, given that c_i de- and reencrypts correctly. Likewise, make the change of H explicit in $\mathrm{DER}_{\mathrm{resp}}$: we change $\mathrm{DER}_{\mathrm{resp}}$ in line 30 such that if $\tilde{pk} = \tilde{pk}^*$, the session keys are defined as $K' := \mathsf{H}_{\mathsf{q}}(m_i, m'_j, \tilde{c}, i, j, M, M')$ whenever c_j decrypts correctly. The latter change is purely conceptual since $\tilde{c}$ is defined as $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$:

$$\begin{aligned} \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M = (\tilde{pk}^*, c_j), M') &= \mathsf{H}_{\mathsf{q}}(m_i, m'_j, \mathsf{Enc}(\tilde{pk}^*, \tilde{m}; \mathsf{G}(\tilde{m})), i, j, M, M') \\ &= \mathsf{H}_{\mathsf{q}}(m_i, m'_j, \tilde{c}, i, j, M, M') \ . \end{aligned}$$

It remains to show that the keys derived by $\mathrm{DER}_{\mathrm{init}}$ are still consistent. Since we enforced in game $G^{\neg\mathrm{st}}_{6,b}$ that no other session than s'_{init} could possibly use public key $\tilde{pk}^*$, this indeed is the only session where we have to modify the definition of K to keep it consistent with our redefinition of H. We will now argue that for $\mathrm{DER}_{\mathrm{init}}(s'_{\mathrm{init}})$, the change is only conceptual as well: Let $(c_i^*, \tilde{c}^*)$ denote the message on which $\mathrm{DER}_{\mathrm{init}}(s'_{\mathrm{init}})$ was called. Since there exists a matching "responder" session that generated c_i^* and $\tilde{c}^*$,

in particular there exists a message $\tilde{m}^*$ such that $\tilde{c}^* = \mathsf{Enc}(\tilde{pk}^*, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$. Since G only samples from good randomness, it holds that $\tilde{m}' := \mathsf{Dec}(\tilde{sk}^*, \tilde{c}^*) = \tilde{m}^*$, and reencryption works as well. Therefore, game $G^{\neg\mathrm{st}}_{9,b}$ should return $\mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M = (\tilde{pk}^*, c_j), M')$, as does $G^{\neg\mathrm{st}}_{8,b}$. Since $\tilde{m}' = \tilde{m}^*$ and hence $\mathsf{Enc}(\tilde{pk}^*, m'; \mathsf{G}(m')) = \tilde{c}^*$, this is indeed the case:

$$\begin{aligned}\mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M = (\tilde{pk}^*, c_j), M') &= \mathsf{H}_{\mathsf{q}}(m'_i, m_j, \mathsf{Enc}(\tilde{pk}^*, \tilde{m}'; \mathsf{G}(\tilde{m}')), i, j, M, M') \\ &= \mathsf{H}_{\mathsf{q}}(m'_i, m_j, \tilde{c}^*, i, j, M, M') \ .\end{aligned}$$

Hence, A's view is identical in both games and

$$\Pr[G^{\neg\mathrm{st}}_{8,b} \Rightarrow 1] = \Pr[G^{\neg\mathrm{st}}_{9,b} \Rightarrow 1] \ .$$

Games $G^{\neg\mathrm{st}}_{10,b}$. In games $G^{\neg\mathrm{st}}_{10,b}$, we switch back to using $\mathsf{G} \leftarrow_{\$} \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{\tilde{pk}^*, \tilde{sk}^*}$. With the same reasoning as for the gamehop from game $G^{\neg\mathrm{st}}_{6,b}$ to $G^{\neg\mathrm{st}}_{7,b}$,

$$\begin{aligned}|\Pr[G^{\neg\mathrm{st}}_{9,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{10,b} \Rightarrow 1]| &= |\Pr[\mathsf{GDPB}^{\mathsf{D}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{D}^{\mathsf{b}}}_{\lambda,0} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta \ ,\end{aligned}$$

where adversary D^{b} is also given in Figure 23.

So far, we established

$$\begin{aligned}|\Pr[G^{\neg\mathrm{st}}_{6,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{6,0} \Rightarrow 1]| \leq &|\Pr[G^{\neg\mathrm{st}}_{10,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{10,0} \Rightarrow 1]| \\ &+ 32 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta \ ,\end{aligned}$$

hence

$$\begin{aligned}|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \leq \ & S \cdot |\Pr[G^{\neg\mathrm{st}}_{10,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{10,0} \Rightarrow 1]| \\ &+ 32 \cdot S \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta + 2S^2 \cdot \mu(\mathsf{KG}) \ .\end{aligned}$$

We stress that from game $G^{\neg\mathrm{st}}_{10,b}$ on, none of the oracles use ephemeral secret key $\tilde{sk}^*$ any longer. To upper bound $|\Pr[G^{\neg\mathrm{st}}_{10,1}{}^{\mathsf{B}} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{10,0}{}^{\mathsf{B}} \Rightarrow 1]|$, consider the sequence of games given in Figure 24, where we replace $\mathrm{sID}^*_{\mathrm{resp}}$'s ciphertext $\tilde{c}$ with a fake encryption. To replace $\tilde{c}$, we first have to add a guess for $\mathrm{sID}^*_{\mathrm{resp}}$.

Games $G^{\neg\mathrm{st}}_{11,b}$. In game $G^{\neg\mathrm{st}}_{11,b}$, one of the sessions that get established during execution of B is picked at random in line 03, and the game returns 0 in line 16 if any other session s'_{resp} was picked than session $\mathrm{sID}^*_{\mathrm{resp}}$. Again,

$$\Pr[G^{\neg\mathrm{st}}_{10,b} \Rightarrow 1] = S \cdot \Pr[G^{\neg\mathrm{st}}_{11,b} \Rightarrow 1] \ .$$

Games $G^{\neg\mathrm{st}}_{12,b}$. In game $G^{\neg\mathrm{st}}_{12,b}$, $\mathrm{DER}_{\mathrm{resp}}$ is changed in line 25 such that for s'_{resp}, $\tilde{c}$ is no longer an encryption of a randomly drawn message $\tilde{m}$, but a fake encryption $\tilde{c} \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$. Consider the adversaries $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}$ against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ given in Figure 25. Each adversary $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}$ needs to generate ephemeral key pairs (at most S times), to (deterministically) encrypt or reencrypt (at most $3S$ times), to decrypt (at most $2S$ times), to evaluate the random oracles $\mathsf{H}^1_{\mathsf{q}}$ to $\mathsf{H}^3_{\mathsf{q}}$ and H' (at most $q_{\mathsf{H}} + S$ times) as well as G (at most $q_{\mathsf{G}} + q_{\mathsf{H}} + 3S$ times), and to lazy sample (at most S times). Hence the total running time is upper bounded as follows:

$$\begin{aligned}\mathrm{Time}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}) &\leq \mathrm{Time}(\mathsf{B}) + S \cdot (\mathrm{Time}(\mathsf{KG}) + 3 \cdot \mathrm{Time}(\mathsf{Enc}) + 2 \cdot \mathrm{Time}(\mathsf{Dec})) + q_{\mathsf{H}} + q_{\mathsf{G}} + 4S \\ &\approx \mathrm{Time}(\mathsf{B}) \ . \qquad (6)\end{aligned}$$

Since $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}$ perfectly simulates game $G^{\neg\mathrm{st}}_{11,b}$ if its input c^* was generated by randomly picking m and letting $c := \mathsf{Enc}(\tilde{pk}^*, m, \mathsf{G}(m))$, and game $G^{\neg\mathrm{st}}_{12,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$,

$$|\Pr[G^{\neg\mathrm{st}}_{11,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{12,b} \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}) \ .$$

GAMES $G^{\neg\mathrm{st}}_{10,b}$ - $G^{\neg\mathrm{st}}_{14,b}$

cnt, $\mathrm{sID}^* := 0$
$s'_{\mathrm{init}} \leftarrow_{\$} [S]$
$s'_{\mathrm{resp}} \leftarrow_{\$} [S]$ // $G^{\neg\mathrm{st}}_{11,b}$ - $G^{\neg\mathrm{st}}_{14,b}$
for $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \mathsf{KG}$
$b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
if ATTACK(sID^*)
  return 0
if $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
if revState[$\mathrm{sID}^*_{\mathrm{init}}$] ABORT
Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. role[$\mathrm{sID}^*_{\mathrm{init}}$] = "initiator"
if $\mathrm{sID}^*_{\mathrm{init}} \neq s'_{\mathrm{init}}$ **return** 0
Pick $\mathrm{sID}^*_{\mathrm{resp}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. role[$\mathrm{sID}^*_{\mathrm{resp}}$] = "responder" // $G^{\neg\mathrm{st}}_{11,b}$-$G^{\neg\mathrm{st}}_{14,b}$
if $\mathrm{sID}^*_{\mathrm{resp}} \neq s'_{\mathrm{resp}}$
  return 0 // $G^{\neg\mathrm{st}}_{11,b}$-$G^{\neg\mathrm{st}}_{14,b}$
return b'

$\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}, M = (\tilde{pk}, c_j))$
if holder[sID] $= \perp$ **or** sKey[sID] $\neq \perp$ **or** role[sID] = "initiator" **return** $\perp$
role[sID] := "responder"
$(j, i) :=$ (holder[sID], peer[sID])
$m_i, \tilde{m} \leftarrow_{\$} \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
if sID $= s'_{\mathrm{resp}}$
  $\tilde{c} \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$ // $G^{\neg\mathrm{st}}_{12,b}$-$G^{\neg\mathrm{st}}_{14,b}$
  if $\tilde{c} \in \mathsf{Enc}(\tilde{pk}, \mathcal{M}; \mathcal{R})$
    ABORT // $G^{\neg\mathrm{st}}_{13,b}$-$G^{\neg\mathrm{st}}_{14,b}$
$M' := (c_i, \tilde{c})$
$m'_j := \mathsf{Dec}(sk_j, c_j)$
if $m'_j = \perp$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
  $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
else
  $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
  if $\tilde{pk} = \tilde{pk}^*$
    $K' := \mathsf{H}_{\mathsf{q}}(m_i, m'_j, \tilde{c}, i, j, M, M')$
sKey[sID] := K'
(received[sID], sent[sID]) := (M, M')
return M'

TEST(sID) //only one query
$\mathrm{sID}^* :=$ sID
if sKey[sID^*] $= \perp$ **return** $\perp$
$K^*_0 :=$ sKey[sID^*] // $G^{\neg\mathrm{st}}_{10,b}$-$G^{\neg\mathrm{st}}_{12,b}$
$K^*_0 \leftarrow_{\$} \mathcal{K}$ // $G^{\neg\mathrm{st}}_{14,0}$
$K^*_1 \leftarrow_{\$} \mathcal{K}$
return K^*_b

Figure 24: Games $G^{\neg\mathrm{st}}_{10,b}$ - $G^{\neg\mathrm{st}}_{14,b}$ for case ($\neg$st) of the proof of Lemma 2. All oracles except for TEST and $\mathrm{DER}_{\mathrm{resp}}$ remain as in game $G^{\neg\mathrm{st}}_{10,b}$ (see Figure 22, page 43).

Folding adversaries $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},0}$ and $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},1}$ into one adversary $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}$ yields

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},0}) + \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},1}) = 2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}) \ .$$

Game $G^{\neg\mathrm{st}}_{13,b}$. In game $G^{\neg\mathrm{st}}_{13,b}$, we abort in line 27 if the fake ciphertext $\tilde{c}$ that was picked during execution of $\mathrm{DER}_{\mathrm{resp}}(s'_{\mathrm{resp}})$ lies within the range of encryption under $\tilde{pk}$, i.e., if $\tilde{c} \in \mathsf{Enc}(\tilde{pk}, \mathcal{M}; \mathcal{R})$. Since PKE is ϵ_{dis}-disjoint,

$$|\Pr[G^{\neg\mathrm{st}}_{12,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{13,b} \Rightarrow 1]| \leq \epsilon_{\mathrm{dis}} \ .$$

Game $G^{\neg\mathrm{st}}_{14,0}$. In game $G^{\neg\mathrm{st}}_{14,0}$, we change oracle TEST in line 42 such that it returns a random value instead of returning $\mathrm{sKey}[\mathrm{sID}^*]$. Since this change renders games $G^{\neg\mathrm{st}}_{14,0}$ and $G^{\neg\mathrm{st}}_{14,1}$ equal, and since game $G^{\neg\mathrm{st}}_{14,1}$ is equal to game $G^{\neg\mathrm{st}}_{13,1}$,

$$|\Pr[G^{\neg\mathrm{st}}_{12,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{12,0} \Rightarrow 1]| = |\Pr[G^{\neg\mathrm{st}}_{14,0} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{13,0} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G^{\neg\mathrm{st}}_{14,0} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{13,0} \Rightarrow 1]|$. B cannot distinguish the value $K^*_0 = \mathrm{sKey}[\mathrm{sID}^*]$ that is returned by $\mathrm{TEST}(\mathrm{sID}^*)$ from random in game $G^{\neg\mathrm{st}}_{13,0}$ unless it obtains K^*_0 (either classically or contained in a quantum answer) at some point other than during the calling of TEST. It's easy to verify that B can only obtain keys (and in particular, K^*_0) by queries to REVEAL or to H.

$\underline{\mathsf{A}^{\neg\mathrm{st}\ \langle\mathsf{H}'\rangle,\langle\mathsf{H}_\mathsf{q}\rangle,\langle\mathsf{G}\rangle}_{\mathsf{DS},\mathsf{b}}(\tilde{pk}^*, c^*)}$

01 $\mathrm{cnt}, \mathrm{sID}^* := 0$
02 $s'_{\mathrm{init}} \leftarrow_\$ [S], s'_{\mathrm{resp}} \leftarrow_\$ [S]$
03 **for** $n \in [N]$
04 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
05 $b' \leftarrow \mathsf{B}^{\mathsf{O},\langle\mathsf{G}\rangle,\langle\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
06 **if** ATTACK(sID^*) **return** 0
07 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
08 **if** $\mathrm{revState}[\mathrm{sID}^*_{\mathrm{init}}]$ ABORT
09 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] =$ "initiator"
10 **if** $\mathrm{sID}^*_{\mathrm{init}} \neq s'_{\mathrm{init}}$ **return** 0
11 Pick $\mathrm{sID}^*_{\mathrm{resp}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{resp}}] =$ "responder"
12 **if** $\mathrm{sID}^*_{\mathrm{resp}} \neq s'_{\mathrm{resp}}$ **return** 0
13 **return** b'

$\underline{\mathrm{REV\text{-}STATE}(\mathrm{sID} \neq s'_{\mathrm{init}})}$
14 **if** $\mathrm{state}[\mathrm{sID}] = \bot$ **return** $\bot$
15 $\mathrm{revState}[\mathrm{sID}] :=$ **true**
16 **return** $\mathrm{state}[\mathrm{sID}]$

$\underline{\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}, M = (\tilde{pk}, c_j))}$
17 **if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$ **or** $\mathrm{role}[\mathrm{sID}] =$ "initiator" **return** $\bot$
18 $\mathrm{role}[\mathrm{sID}] :=$ "responder"
19 $(j, i) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
20 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
21 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
22 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
23 **if** $\mathrm{sID} = s'_{\mathrm{resp}}$
24 $\quad \tilde{c} := c^*$
25 $M' := (c_i, \tilde{c})$
26 $m'_j := \mathsf{Dec}(sk_j, c_j)$
27 **if** $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
28 $\quad K' := \mathsf{H}'_\mathsf{R}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
29 **else**
30 $\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
31 $\quad$ **if** $\tilde{pk} = \tilde{pk}^*$
32 $\quad\quad K' := \mathsf{H}_\mathsf{q}(m_i, m'_j, \tilde{c}, i, j, M, M')$
33 $\mathrm{sKey}[\mathrm{sID}] := K'$
34 $(\mathrm{received}[\mathrm{sID}], \mathrm{sent}[\mathrm{sID}]) := (M, M')$
35 **return** M'

Figure 25: Adversaries $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS},\mathsf{b}}$ for case ($\neg$st) of the proof of Lemma 2, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H}_\mathsf{q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for $\mathrm{DER}_{\mathrm{resp}}$ and REV-STATE are defined as in game $G^{\neg\mathrm{st}}_{11,b}$ (see Figure 24, page 47).

We will first make explicit how the key is defined: Let i^* and j^* denote holder and peer of the "initiator" session. Recall that $\tilde{pk}^*$ denotes the ephemeral key that was chosen in the beginning of the game (see Figure 21, line 05) and used during execution of $\mathrm{INIT}(\mathrm{sID}^*_{\mathrm{init}})$ (line 25, also Figure 21). Let m^*_j denote the randomly chosen message with encryption $c^*_j := \mathsf{Enc}(pk_{j^*}, m^*_j; \mathsf{G}(m^*_j))$ that was sampled during execution of $\mathrm{INIT}(\mathrm{sID}^*_{\mathrm{init}})$, furthermore let $\tilde{c}^*$ denote the fake ciphertext that was sampled under $\tilde{pk}^*$ during execution of $\mathsf{Der}_{\mathsf{resp}}(\mathrm{sID}^*_{\mathrm{resp}})$ (Figure 24, line 25) and let m^*_i denote the randomly chosen message with encryption $c^*_i := \mathsf{Enc}(pk_{i^*}, m^*_i; \mathsf{G}(m^*_i))$ that was picked during execution of $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*_{\mathrm{resp}})$. We changed the key derivation such that since $\tilde{pk}^*$ is used, in the case that sID^* is an "initiator" session, we have

$$K^*_0 = \begin{cases} \mathsf{H}'_{\mathsf{L1}}(c^*_i, m^*_j, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & m'_i = \bot \text{ or } \mathsf{Enc}(pk_{i^*}, m'_i) \neq c^*_i \\ \mathsf{H}_\mathsf{q}(m'_i, m^*_j, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \text{o.w.} \end{cases},$$

where $m'_i := \mathsf{Dec}(sk_{i^*}, c^*_i)$. In the case that sID^* is a "responder" session, we have

$$K^*_0 = \begin{cases} \mathsf{H}'_\mathsf{R}(m^*_i, c^*_j, \tilde{m}^*, i^*, j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & m'_j = \bot \text{ or } \mathsf{Enc}(pk_{j^*}, m'_j) \neq c^*_j \\ \mathsf{H}_\mathsf{q}(m^*_i, m^*_j, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \text{o.w.} \end{cases},$$

where $m'_j := \mathsf{Dec}(sk_{j^*}, c^*_j)$.

First, we will argue that B could not possibly obtain K^*_0 by a query to REVEAL: Recall that B trivially loses if it revealed the test session or its match. Hence, B would have to create some session $\mathrm{sID} \notin \{\mathrm{sID}^*_{\mathrm{init}}, \mathrm{sID}^*_{\mathrm{resp}}\}$ that derives the same key as the test session. First, we argue that the key cannot be obtained via any session sID with $\mathrm{role}[\mathrm{sID}] \neq \mathrm{role}[\mathrm{sID}^*]$: Since both exchanged messages M and M' are hashed, the same key could only be derived if the respective session matches the test session. Since creation of an additional matching session would result in an abort, we can ignore this case, and only need to consider the case that $\mathrm{role}[\mathrm{sID}] = \mathrm{role}[\mathrm{sID}^*]$.

We first consider the case that sID^* is an "initiator" session: To obtain K^*_0 via another "initiator" session, B would have to establish and initialize another "initiator" session $\mathrm{sID} \neq \mathrm{sID}^*_{\mathrm{init}}$ with holder i^* and peer j^*. The subsequent call to $\mathrm{DER}_{\mathrm{init}}$ could only result in the same key if $\mathrm{INIT}(\mathrm{sID})$ had also picked ephemeral key $\tilde{pk}^*$, which is impossible since we enforced in game $G^{\neg\mathrm{st}}_{6,b}$ that no other session uses

$\tilde{pk}^*$. Now we consider the case that sID^* is a "responder" session. To obtain K_0^* via another "responder" session sID, B would have to call DER_{resp} for some session sID $\neq \text{sID}^*_{\text{resp}}$ with holder j^*, peer i^*, on the same message $M = (\tilde{pk}^*, c_j^*)$. Since the resulting key includes the message $(c_i, \tilde{c})$ that was computed by session sID, it can only be equal to the test session's key if $(c_i, \tilde{c}) = (c_i^*, \tilde{c}^*)$. Since $\tilde{c}$ is an encryption of some message, and $\tilde{c}^*$ does not lie within the range of $\mathsf{Enc}(\tilde{pk}, -; -)$, this is impossible. Either way, recreation of the test session's key is impossible.

To upper bound the probability that any of the quantum answers of $|\mathsf{H}\rangle$ could contain session key $K_0^* = \mathsf{H}_\mathsf{q}\ (m_i^*, m_j^*, \tilde{c}^*, i^*, j^*, M, M')$, recall that for any message $M = (\tilde{pk}^*, c_j)$, where c_j is arbitrary,

$$\mathsf{H}(m_1, m_2, m_3, i^*, j^*, M = (\tilde{pk}^*, c_j), M') = \mathsf{H}_\mathsf{q}(m_1, m_2, \mathsf{Enc}(\tilde{pk}^*, m_3; \mathsf{G}(m_3)), i^*, j^*, M, M') \ .$$

Hence, to trigger a query to $|\mathsf{H}_\mathsf{q}\rangle$ containing the classical query $(m_i^*, m_j^*, \tilde{c}^*, i^*, j^*, M, M')$, B would need to come up with a message m such that $\mathsf{Enc}(\tilde{pk}^*, m; \mathsf{G}(m)) = \tilde{c}^*$. Since $\tilde{c}^*$ does not lie in the range of $\mathsf{Enc}(\tilde{pk}, -; -)$, this is impossible with probability at most ϵ_{dis}.

In conclusion,

$$\begin{aligned}|\Pr[G^{\neg\text{st}}_{12,1} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{12,0} \Rightarrow 1]| &= |\Pr[G^{\neg\text{st}}_{14,0} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{13,0} \Rightarrow 1]| \\ &\leq \frac{S-2}{|\mathcal{M}|} \cdot \max\{\frac{1}{|\mathcal{M}|}, \epsilon_{\text{dis}}\} + \epsilon_{\text{dis}} \ \leq \frac{S}{|\mathcal{M}|} + \epsilon_{\text{dis}} \ ,\end{aligned}$$

hence

$$\begin{aligned}&|\Pr[G^{\neg\text{st}}_{10,1} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{10,0} \Rightarrow 1]| \\ &\quad\leq\ S \cdot \left(|\Pr[G^{\neg\text{st}}_{12,1} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{12,0} \Rightarrow 1]| + 2 \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\text{st}}_{\mathsf{DS}})\right) \\ &\quad\leq 2S \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\text{st}}_{\mathsf{DS}}) + 2S \cdot \epsilon_{\text{dis}} \ .\end{aligned}$$

Collecting the probabilities yields

$$\begin{aligned}|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\text{st}]| &\leq 2S^2 \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\text{st}}_{\mathsf{DS}}) + 32 \cdot S \cdot (q_\mathsf{G} + q_\mathsf{H} + 3S + 1)^2 \cdot \delta \\ &\quad 2S^2 \cdot \epsilon_{\text{dis}} + 2S^2 \cdot \mu(\mathsf{KG}) \ ,\end{aligned}$$

the upper bound we claimed in equation (3).

D.2 Case $(\neg sk)$ of the Proof of Lemma 2

Case $(\neg sk)$ (Initialising session's owner was not corrupted). Intuition is as follows: While B might have obtained both the secret key of peer[$\text{sID}^*_{\text{init}}$] and $\text{sID}^*_{\text{init}}$'s internal state, we can replace ciphertext c_i since holder[$\text{sID}^*_{\text{init}}$], henceforth called i^*, is not corrupted. To be able to replace c_i, we will patch in encryption at the first (and due to the need for symmetry, at the second) argument of the random oracle.

Consider the sequence of games given in Figures 26, 27 and 29: First, we will enforce that indeed, we only need to consider the case where ¬corrupted[holder[$\text{sID}^*_{\text{init}}$]]. Afterwards, we ensure that the game makes no use of sk_{i^*} any longer by patching encryption into the random oracle (in games $G^{\neg sk}_{2,b}$ to $G^{\neg sk}_{9,b}$, see Figure 27, line 53). This is the only part of the proof where we need to consider the adversary's capability to come up with encryptions that decrypt incorrectly. Next, during execution of $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}})$, we replace $c_i = \mathsf{Enc}(pk_{i^*}, m_i^*)$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G^{\neg sk}_{10,b}$ to $G^{\neg sk}_{13,b}$, see Figure 29). We show that after those changes, B's view does not change with overwhelming probability if we finally change TEST such that it always returns a random value (game $G^{\neg sk}_{13,b}$, also Figure 29).

Game $G^{\neg sk}_{2,b}$. Since games $G^{\neg sk}_{2,b}$ and $G_{2,b}$ are the same,

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\ &\qquad = |\Pr[G^{\neg sk^{\mathsf{B}}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\neg sk^{\mathsf{B}}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \ .\end{aligned}$$

GAMES $G^{\neg sk}_{2,b}$ - $G^{\neg sk}_{4,b}$

01 $\text{cnt}, \text{sID}^* := 0$
02 $i' \leftarrow_{\$} [N]$ // $G^{\neg sk}_{4,b}$
03 **for** $n \in [N]$
04 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
05 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
06 **if** ATTACK(sID^*)
07 $\quad$ **return** 0
08 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
09 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$
$\quad$ s. th. role[$\text{sID}^*_{\text{init}}$] = "initiator" // $G^{\neg sk}_{3,b}$-$G^{\neg sk}_{4,b}$
10 **if** corrupted[holder[$\text{sID}^*_{\text{init}}$]]
11 $\quad$ ABORT // $G^{\neg sk}_{3,b}$-$G^{\neg sk}_{7,b}$
12 **if** holder[$\text{sID}^*_{\text{init}}$] $\neq i'$
13 $\quad$ **return** 0 // $G^{\neg sk}_{4,b}$-$G^{\neg sk}_{4,b}$
14 **return** b'

Figure 26: Games $G^{\neg sk}_{2,b}$ - $G^{\neg sk}_{4,b}$ for case ($\neg sk$) of the proof of Lemma 2. Helper procedure ATTACK and oracles TEST, Init, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 12 and Figure 13, pages 21 and 22).

GAMES $G^{\neg sk}_{3,b}$. Both games $G^{\neg sk}_{3,b}$ abort in line 11 if corrupted[holder[$\text{sID}^*_{\text{init}}$]]. Since for both bits b it holds that $\Pr[G^{\neg sk\mathsf{B}}_{3,b} \Rightarrow 1] = \Pr[G^{\neg sk\mathsf{B}}_{2,b} \Rightarrow 1 \wedge \neg sk]$,

$$|\Pr[G^{\neg sk\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\neg sk\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| = |\Pr[G^{\neg sk}_{3,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{3,0} \Rightarrow 1]| \ .$$

Analogous to our proof of case ($\neg$st), the first goal is not to have to make use of the initialiser's s secret key any longer. Since initialiser $i^* = \text{holder}[\text{sID}^*_{\text{init}}]$ is not fixed until B issues the TEST query, we first add a guess i' for i^*. In the subsequent games, encryption will be plugged into random oracle H for the first two messages (since $pk_{i'}$ could be used in both slots) Afterwards, we will even out the difference in derivation for ciphertexts with de- or reencryption failure and ciphertexts without. As in the proof of case ($\neg$st), we will see that these changes do not affect B's view unless it is able to distinguish random oracle G from an oracle $\mathsf{G}_{pk_{i'},sk_{i'}}$ that only samples randomness under which decryption never fails, thereby allowing for a reduction to game GDPB.

GAMES $G^{\neg sk}_{4,b}$. In both games $G^{\neg sk}_{4,b}$, one of the parties is picked at random in line 02, and the games return 0 in line 13 if any other party i' was picked than the holder of $\text{sID}^*_{\text{init}}$. Since for both bits b it holds that games $G^{\neg sk}_{4,b}$ and $G^{\neg sk}_{3,b}$ proceed identically if $\text{holder}[\text{sID}^*_{\text{init}}] = i'$, and since games $G^{\neg sk}_{4,b}$ output 0 if $\text{holder}[\text{sID}^*_{\text{init}}] \neq i'$, we have that $\Pr[G^{\neg sk}_{3,b} \Rightarrow 1] = N \cdot \Pr[G^{\neg sk}_{4,b} \Rightarrow 1]$ and hence,

$$|\Pr[G^{\neg sk}_{3,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{3,0} \Rightarrow 1]| = N \cdot |\Pr[G^{\neg sk}_{4,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{4,0} \Rightarrow 1]| \ .$$

Figure 27: Games $G^{\neg sk}_{4,b}$ - $G^{\neg sk}_{9,b}$ for case $(\neg sk)$ of the proof of Lemma 2. Helper procedure ATTACK and oracles TEST, INIT, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 12 and Figure 13, pages 21 and 22).

To prepare getting rid of $sk_{i'}$, we will first modify random oracle G such that it renders PKE perfectly correct for key pair $(pk_{i'}, sk_{i'})$.

GAME $G^{\neg sk}_{5,b}$. In game $G^{\neg sk}_{5,b}$, we enforce that no decryption failure with respect to key pair $(pk_{i'}, pk_{i'})$ will occur: We replace random oracle G with $\mathsf{G}_{pk_{i'}, sk_{i'}}$ in line 06, where $\mathsf{G}_{pk_{i'}, pk_{i'}}$ is defined in line 71 by

$$\mathsf{G}_{pk_{i'}, sk_{i'}}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk_{i'}, sk_{i'}, m); f(m)) \ ,$$

with $\mathcal{R}_{\mathrm{bad}}(pk, sk, m) := \{r \in \mathcal{R} \mid \mathsf{Dec}(sk, \mathsf{Enc}(pk, m; r)) \neq m\}$ denoting the set of bad randomness for any fixed key pair (pk, sk), and any message $m \in \mathcal{M}$,

$$\delta(pk, sk, m) := {}^{|\mathcal{R}_{\mathrm{bad}}(pk,sk,m)|}/_{|\mathcal{R}|} \tag{7}$$

denoting the fraction of bad randomness, and $\delta(pk, sk) := \max_{m \in \mathcal{M}} \delta(pk, sk, m)$. (As in the proof of (¬st).) Recall that 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 $|\Pr[G^{\neg sk}_{4,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{5,b} \Rightarrow 1]|$ for each bit b, we construct (unbounded, quantum) adversaries C^{b} against the generic distinguishing problem with bounded probabilities GDPB_{λ} in Figure 28, issuing at most $q_{\mathsf{G}} + 3S$ queries to $|\mathsf{F}\rangle$. With the same analysis as in our proof for case (¬st) (see page 44),

$$|\Pr[G^{\neg sk}_{4,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{5,b} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \ ,$$

and according to Lemma 4,

$$\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \leq 8 \cdot (q_{\mathsf{G}} + 3S + 1)^2 \cdot \delta \ .$$

$\underline{\mathsf{C}^{\mathsf{b}}_1 = \mathsf{D}^{\mathsf{b}}_1}$
01 $(pk, sk) \leftarrow \mathsf{KG}$
02 **for** $m \in \mathcal{M}$
03 $\quad \lambda(m) := \delta(pk, sk, m)$
04 **return** $(\lambda(m))_{m \in \mathcal{M}}$

$\underline{\mathsf{G}(m)}$
05 **if** $\mathsf{F}(m) = 0$
06 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
07 **else**
08 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R}_{\mathrm{bad}}(pk, sk, m); f(m))$
09 **return** $\mathsf{G}(m)$

$\underline{\mathsf{C}^{\mathsf{b}|\mathsf{F}\rangle}_2, \mathsf{D}^{\mathsf{b}|\mathsf{F}\rangle}_2}$
10 $\mathrm{cnt}, \mathrm{sID}^* := 0$
11 $i' \leftarrow_{\$} [N]$
12 **for** $n \in [N] \setminus \{i'\}$
13 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
14 $(pk_{i'}, sk_{i'}) := (pk, sk)$
15 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
16 **if** ATTACK(sID^*)
17 $\quad$ **return** 0
18 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
19 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$
$\quad$ s. th. role[$\mathrm{sID}^*_{\mathrm{init}}$] = "initiator"
20 **if** corrupted[holder[$\mathrm{sID}^*_{\mathrm{init}}$]] ABORT
21 **if** holder[$\mathrm{sID}^*_{\mathrm{init}}$] $\neq i'$
22 $\quad$ **return** 0
23 **return** b'

Figure 28: Unbounded quantum adversaries C^{b} and D^{b}, executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$, for case $(\neg sk)$ of the proof of Lemma 2. The adversaries only differ in their definition of $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$ and H: For adversaries C^{b}, $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$ and H are defined as in game $G^{\neg sk}_{4,b}$, see Figure 27, while for adversaries D^{b}, $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$ and H are defined as in game $G^{\neg sk}_{8,b}$ (also Figure 27).

Recall that the goal is to simulate the game without knowledge of $sk_{i'}$. To this end, we will first change the key derivation procedure $\mathrm{DER}_{\mathrm{init}}$ for holder i' as follows: If ciphertext $\tilde{c}$ already induces decryption failure, we will not have to use $sk_{i'}$ any more to check whether c_i induces de- or reencryption failure as well.

Games $G^{\neg sk}_{6,b}$. In both games $G^{\neg sk}_{6,b}$, we change oracle $\mathrm{DER}_{\mathrm{init}}$ in line 35 for session holder i' as follows: Whenever $\tilde{c}$ does not decrypt to a message $\tilde{m}'$ s. th. $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}', \mathsf{G}(\tilde{m}'))$, the session key is defined as $K := \mathsf{H}'_{\mathsf{L}1}(c_i, m_j, \tilde{c}, i, j, M, M')$. (Before this change we let $K := \mathsf{H}'_{\mathsf{L}3}(m'_i, m_j, \tilde{c}, i, j, M, M')$ in the case that $\tilde{c}$ fails to decrypt, but c_i decrypts correctly.)

We claim that this change does not affect B's view since the cases are logically distinct: Since both $\mathsf{H}'_{\mathsf{L}1}$ and $\mathsf{H}'_{\mathsf{L}2}$ are internal random oracles that cannot be accessed directly by B, the only way to obtain oracle values of $\mathsf{H}'_{\mathsf{L}1}$ and $\mathsf{H}'_{\mathsf{L}2}$ is via calls to REVEAL and TEST after execution of $\mathrm{DER}_{\mathrm{init}}$. Intuitively, B could only tell the games apart by establishing and completing two distinct "initiator" sessions s and s' with holder i' such that they derive different keys in game $G^{\neg sk}_{5,b}$, but the same key in game $G^{\neg sk}_{6,b}$. In more detail, the following requirements would have to be met:

- Both sesions have the same peer j, and algorithms INIT(s) and INIT(s') pick the same key pair $(\tilde{pk}, \tilde{sk})$ as well as the same message m_j
- Calling $\mathrm{DER}_{\mathrm{init}}(s)$ on some message $(c_i, \tilde{c})$ results in session key $\mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i', j, M, M')$ in both games
- Calling $\mathrm{DER}_{\mathrm{init}}(s')$ on some message $(c'_i, \tilde{c}')$ in game $G^{\neg sk}_{5,b}$ results in session key $\mathsf{H}'_{\mathsf{L3}}(\mathsf{Dec}(sk_i, c'_i), m_j, \tilde{c}', i, j, M, M')$ and in game $G^{\neg sk}_{6,b}$, it derives the same key as $\mathrm{DER}_{\mathrm{init}}(s)$, i.e., its key is now $\mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i', j, M, M')$.

To achieve key equality in game $G^{\neg sk}_{6,b}$ it is required that both $c'_i = c_i$ and $\tilde{c}' = \tilde{c}$, i.e., both $\mathrm{DER}_{\mathrm{init}}(s)$ and $\mathrm{DER}_{\mathrm{init}}(s')$ were called on the same message $(c_i, \tilde{c})$. Requiring that $\mathrm{DER}_{\mathrm{init}}(s)$ computes its key as $\mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i', j, M, M')$ in game $G^{\neg sk}_{5,b}$ implies that ciphertext c_i is problematic. But additionally requiring that $\mathrm{DER}_{\mathrm{init}}(s)$ computes its key as $\mathsf{H}'_{\mathsf{L3}}(\mathsf{Dec}(sk_i, c'_i), m_j, \tilde{c}', i, j, M, M')$ in game $G^{\neg sk}_{5,b}$ implies that ciphertext $c_i = c'_i$ is non-problematic at the same time. Since this is impossible, B's view does not change and

$$\Pr[G^{\neg sk}_{5,b} \Rightarrow 1] = \Pr[G^{\neg sk}_{6,b} \Rightarrow 1] \ .$$

In the next game, we change key definition of $\mathrm{DER}_{\mathrm{init}}$ if both ciphertexts de- and reencrypt correctly, and key definition of $\mathrm{DER}_{\mathrm{resp}}$ if c_j de-and reencrypts correctly. In these cases, we do note use the decryptions under $sk_{i'}$, but the ciphertexts themself. Similar to case ($\neg$st), we "plug in" encryption into random oracle H whenever i' appears as one of the involved parties. Since $sk_{i'}$ might be used for the first as well as the second message, depending on the session's role, we have to plug encryption into either of the first two arguments of the random oracle, accordingly.

Games $G^{\neg sk}_{7,b}$. In games $G^{\neg sk}_{7,b}$, the random oracle is changed as follows: Instead of picking H uniformly random, we pick four random oracles $\mathsf{H}^1_{\mathsf{q}}$ to $\mathsf{H}^3_{\mathsf{q}}$, and H', and define

$$\mathsf{H}(m_1, m_2, m_3, i, j, M, M') := \begin{cases} \mathsf{H}^1_{\mathsf{q}}(\mathsf{Enc}(pk_{i'}, m_1; \mathsf{G}(m_1)), m_2, m_3, i', j, M, M') & i = i' \wedge j \neq i' \\ \mathsf{H}^2_{\mathsf{q}}(m_1, \mathsf{Enc}(pk_{i'}, m_2; \mathsf{G}(m_2)), m_3, i, i', M, M') & i \neq i' \wedge j = i' \\ \mathsf{H}^3_{\mathsf{q}}(\mathsf{Enc}(pk_{i'}, m_1; \mathsf{G}(m_1)), \mathsf{Enc}(pk_{i'}; \mathsf{G}(m_2)), m_2, m_3, i', j, M, M') & i = j = i' \\ \mathsf{H}'(m_1, m_2, m_3, i, j, M, M') & \text{o.w.} \end{cases} ,$$

see lines 73 to 79. Note that since G only samples from good randomness, encryption under public key $pk_{i'}$ is rendered perfectly correct and hence, injective. Since encryption under public key $pk_{i'}$ is injective, H still is uniformly random.

We make the change of H explicit in the derivation oracles: We have to change $\mathrm{DER}_{\mathrm{resp}}$ for the case that c_j was unproblematic, because this is the only case in which we use H, and (at least) one of the involved parties is i', because this is the only case in which we do not just use random oracle H'. This is done in lines 63 to 67: If i' is holder of the respective session, we define the session key not relative to decrypted message m'_j, but relative to received ciphertext c_j. If i' is the peer of the respective session, we keep using decrypted message m'_j, but we do not use m_i. Instead, we use its encryption c_i. If i' is even both holder and peer of the respective session, we use received ciphertext c_j and encryption c_i. These changes are purely conceptual since m_i encrypts to c_i, and we are in the case that m'_j reencrypts to c_j: In this setting, we have

$$\begin{aligned} \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M') &= \mathsf{H}^2_{\mathsf{q}}(m_i, \mathsf{Enc}(pk_{i'}, m'_j; \mathsf{G}(m'_j)), \tilde{m}, i, i', M, M') \\ &= \mathsf{H}^2_{\mathsf{q}}(m_i, c_j, \tilde{m}, i, i', M, M') = K' \end{aligned}$$

if the holder j is i', but the peer i is not, and we have

$$\begin{aligned} \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M') &= \mathsf{H}^1_{\mathsf{q}}(\mathsf{Enc}(pk_{i'}, m_i; \mathsf{G}(m_i)), m'_j, \tilde{m}, i', j, M, M') \\ &= \mathsf{H}^1_{\mathsf{q}}(c_i, m'_j, \tilde{m}, i', j, M, M') = K' \end{aligned}$$

if the peer i is i', but the holder j is not, and

$$\begin{aligned} \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M') &= \mathsf{H}^3_{\mathsf{q}}(\mathsf{Enc}(pk_{i'}, m_i; \mathsf{G}(m_i)), \mathsf{Enc}(pk_{i'}, m'_j; \mathsf{G}(m'_j)), \tilde{m}, i', i', M, M') \\ &= \mathsf{H}^3_{\mathsf{q}}(c_i, c_j, \tilde{m}, i', i', M, M') = K' \end{aligned}$$

if both holder and peer are i'.

Likewise, make the change of H explicit in $\mathrm{DER}_{\mathrm{init}}$: We have to change $\mathrm{DER}_{\mathrm{resp}}$ for the case that $\tilde{c}$ did not decrypt to $\perp$ and c_i was unproblematic, because this is the only case in which we use H, and the subcase that (at least) one of the involved parties is i', because this is the only case in which we do not just use random oracle H'. This is done in lines 39 to 43: If i' is holder of the respective session, we define the session key not relative to decrypted message m_i', but relative to received ciphertext c_i. If i' is the peer of the respective session, we keep using decrypted message m_i', but we do not use m_j. Instead, we use its encryption c_j. If i' is even both holder and peer of the respective session, we use received ciphertext c_i and encryption c_j. These changes are purely conceptual since m_j encrypts to c_j, and we are in the case that m_i' reencrypts to c_i: In this setting, we have

$$\begin{aligned}\mathsf{H}(m_i', m_j, \tilde{m}', i, j, M, M') &= \mathsf{H}_{\mathsf{q}}^1(\mathsf{Enc}(pk_{i'}, m_i'; \mathsf{G}(m_i')), m_j, \tilde{m}', i', j, M, M') \\ &= \mathsf{H}_{\mathsf{q}}^1(c_i, m_j, \tilde{m}', i', j, M, M') = K'\end{aligned}$$

if the holder i is i', but the peer j is not, and we have

$$\begin{aligned}\mathsf{H}(m_i', m_j, \tilde{m}', i, j, M, M') &= \mathsf{H}_{\mathsf{q}}^2(m_i', \mathsf{Enc}(pk_{i'}, m_j; \mathsf{G}(m_j)), \tilde{m}', i, i', M, M') \\ &= \mathsf{H}_{\mathsf{q}}^2(m_i', c_j, \tilde{m}', i, i', M, M') = K'\end{aligned}$$

if the holder i is not i', but the peer j is i', and

$$\begin{aligned}\mathsf{H}(m_i', m_j, \tilde{m}', i, j, M, M') &= \mathsf{H}_{\mathsf{q}}^3(\mathsf{Enc}(pk_{i'}, m_i'; \mathsf{G}(m_i')), \mathsf{Enc}(pk_{i'}, m_j; \mathsf{G}(m_j)), \tilde{m}', i', i', M, M') \\ &= \mathsf{H}_{\mathsf{q}}^3(c_i, c_j, \tilde{m}', i', i', M, M') = K'\end{aligned}$$

if both holder and peer are i'. Since key consistency is kept, all changes are purely conceptual and

$$\Pr[G_{6,b}^{\neg sk} \Rightarrow 1] = \Pr[G_{7,b}^{\neg sk} \Rightarrow 1] \ .$$

The final step to get rid of $sk_{i'}$ is to even out the key derivation ciphertexts that are problematic with respect to secret key $sk_{i'}$: To this end, we also use $\mathsf{H}_{\mathsf{q}}^1$ to $\mathsf{H}_{\mathsf{q}}^3$ if a ciphertext fails to de-or reencrypt under $sk_{i'}$, instead of using the implicit reject oracles.

Games $G_{8,b}^{\neg sk}$. In games $G_{8,b}^{\neg sk}$, we change $\mathrm{DER}_{\mathrm{resp}}$ in lines 57 to 59 such that if c_j fails to de- or reencrypt and i' is the session's holder, the session key is defined relative to the random oracles $\mathsf{H}_{\mathsf{q}}^2$ or $\mathsf{H}_{\mathsf{q}}^3$ instead of rejecting implicitly, just as if c_j had reencrypted correctly. Likewise, we change $\mathrm{DER}_{\mathrm{init}}$ in lines 28 to 30 such that if c_i fails to de- or reencrypt and i' is the session's holder, but ciphertext $\tilde{c}$ does not decrypt to $\perp$, the session key is defined relative to the random oracles $\mathsf{H}_{\mathsf{q}}^1$ or $\mathsf{H}_{\mathsf{q}}^3$ instead of rejecting implicitly, just as if c_i had reencrypted correctly.

We now argue that this change could not possibly affect B's view: B could only tell the games apart by either

- establishing and revealing two matching sessions such that the keys mismatched in game $G_{7,b}^{\neg sk}$ due to an implicit reject, while in game $G_{8,b}^{\neg sk}$, this difference is evened out by the changes described above, or by
- establishing a session such that its key resulted from an implicit reject in game $G_{7,b}^{\neg sk}$, while the key can be linked to random oracle H in game $G_{8,b}^{\neg sk}$.

It is easy to verify that the former only happens if there exists a completed "initiator" session s with holder i and peer j, and also a completed "responder" session s', with holder j and peer i, such that one of the following conditions hold:

- $i = j = i'$ and at least one of the ciphertexts c_i or c_j is problematic,
- $i = i'$, and $j \neq i'$, and ciphertext c_i is problematic,
- $i \neq i'$, while $j = i'$, and ciphertext c_j is problematic.

Here, c_j is the encryption that INIT(s) returned, and c_j is the encryption that DER_{resp} returned. The conditions above are unsatisfiable since $\mathsf{G}_{pk_{i'},sk_{i'}}$ only samples good randomness: We have $c_i = \mathsf{Enc}(pk_i, m_i; \mathsf{G}_{pk_{i'},sk_{i'}}(m_i))$ and $c_j = \mathsf{Enc}(pk_j, m_j; \mathsf{G}_{pk_{i'},sk_{i'}}(m_j))$. Both $\mathsf{G}_{pk_{i'},sk_{i'}}(m_i)$ and $\mathsf{G}_{pk_{i'},sk_{i'}}(m_j)$ are good randomness for message m_i or m_j, respectively. If $i = i'$, we can conclude that c_i decrypts to m_i (and hence, reencryption also works), therefore c_i can not be problematic. If $j = i'$, we can conclude that c_j decrypts to m_j (and hence, reencryption also works), therefore c_j can not be problematic. Either way, the conditions can not be satisfied. Since the keys' mismatch will be kept in game $G^{\neg sk}_{8,b}$ in any other case, a loss of mismatching keys is impossible.

The latter, i.e., linking a key to H that should not have been linked, is impossible as well: Assume that the session is an "initiator" session. We only have to consider the case that it is a session with holder i'. Let c_i denote the ciphertext received by $\text{DER}_{\text{init}}(s)$. First we examine the case that there exists some message m_i such that $c_i = \mathsf{Enc}(pk_i, m; \mathsf{G}_{pk_{i'},sk_{i'}}(m_i))$: In this case, an implicit reject can not happen since c_i can not be problematic due to the reasons given above. Now we examine the case that there exists no message m_i such that $c_i = \mathsf{Enc}(pk_i, m_i; \mathsf{G}_{pk_{i'},sk_{i'}}(m_i))$: In this case, while the derived key would be defined as $\mathsf{H}^1_{\mathsf{q}}(c_i, m_j, \tilde{m}', i', j, M, M')$ or $\mathsf{H}^3_{\mathsf{q}}(c_i, c_j, \tilde{m}', i', i', M, M')$, respectively, it could not possibly correlate to any random oracle query to $|\mathsf{H}\rangle$: Since $i = i'$, $|\mathsf{H}\rangle$ plugs encryption into the first argument, and we only consider the case that c_i does not lie in the encryption's range. Hence, the respective key is still uniformly random and not linked to H. The argument is completely symmetrical, hence a link of keys can also not happen if the session is a "responder" session.

In conclusion, B's view is identical in both games and

$$\Pr[G^{\neg sk}_{7,b} \Rightarrow 1] = \Pr[G^{\neg sk}_{8,b} \Rightarrow 1] \ .$$

Game $G^{\neg sk}_{9,b}$. In game $G^{\neg sk}_{9,b}$, we switch back to using $\mathsf{G} \leftarrow\!\!\$\ \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{pk_{i'},sk_{i'}}$. With the same reasoning as for the gamehop from game $G^{\neg sk}_{4,b}$ to $G^{\neg sk}_{5,b}$,

$$\begin{aligned} |\Pr[G^{\neg sk}_{8,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{9,b} \Rightarrow 1]| &= |\Pr[\mathsf{GDPB}^{\mathsf{D}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{D}^{\mathsf{b}}}_{\lambda,0} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta \ , \end{aligned}$$

where adversary D^{b} also is given in Figure 28.

So far, we established

$$\begin{aligned} &|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\ &\qquad \leq N \cdot |\Pr[G^{\neg sk}_{9,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{9,0} \Rightarrow 1]| + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta \ . \end{aligned}$$

To upper bound $|\Pr[G^{\neg sk}_{9,b} \Rightarrow 1] - 1/2|$, consider the sequence of games given in Figure 29.

Games $G^{\neg sk}_{10,b}$. Games $G^{\neg sk}_{10,b}$ do not differ from games $G^{\neg sk}_{9,b}$. We only changed the structure of the case distinctions in DER_{resp} and DER_{init} to achieve more readability. It is easy to verify that all cases are still treated exactly the same as in game $G^{\neg sk}_{9,b}$.

$$\Pr[G^{\neg sk}_{9,b} \Rightarrow 1] = \Pr[G^{\neg sk}_{10,b} \Rightarrow 1] \ .$$

We stress that from game $G^{\neg sk}_{9,b}$ on, none of the oracles uses $sk_{i'}$ any longer: DER_{resp} and DER_{init} were changed accordingly, and we only consider the case that B did not query CORRUPT on i'. Since we want to replace $\text{sID}^*_{\text{resp}}$'s ciphertext c_i with a fake encryption, we first have to add a guess for $\text{sID}^*_{\text{resp}}$, like in case ($\neg$st).

Games $G^{\neg sk}_{11,b}$. In games $G^{\neg sk}_{11,b}$, one of the sessions that get established during execution of B is picked at random in line 03, and the games return 0 in line 16 if any other session s'_{resp} was picked than session $\text{sID}^*_{\text{resp}}$. Since for both bits b it holds that both games $G^{\neg sk}_{11,b}$ and $G^{\neg sk}_{10,b}$ proceed identically unless $s'_{\text{resp}} \neq \text{sID}^*_{\text{resp}}$, and since games $G^{\neg sk}_{11,b}$ output 0 if $s'_{\text{resp}} \neq \text{sID}^*_{\text{resp}}$,

$$\Pr[G^{\neg sk}_{10,b} \Rightarrow 1] = S \cdot \Pr[G^{\neg sk}_{11,b} \Rightarrow 1] \ .$$

Games $G^{\neg sk}_{12,b}$. In games $G^{\neg sk}_{12,b}$, oracle DER_{resp} is changed in line 53 such that for $\text{sID}^*_{\text{resp}}$, c_i is no longer a ciphertext of the form $c_i := \mathsf{Enc}(pk_{i'}, m_i; \mathsf{G}(m_i))$ for some randomly drawn message m_i, but a fake

GAMES $G^{\neg sk}_{10,b}$-$G^{\neg sk}_{13,b}$

cnt, $\text{sID}^* := 0$
$i' \leftarrow_\$ [N]$
$s'_{\text{resp}} \leftarrow_\$ [S]$ // $G^{\neg sk}_{11,b}$-$G^{\neg sk}_{13,b}$
for $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
if ATTACK(sID^*)
  return 0
if $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. $\text{role}[\text{sID}^*_{\text{init}}] = \text{"initiator"}$
if corrupted[holder[$\text{sID}^*_{\text{init}}$]] ABORT
Pick $\text{sID}^*_{\text{resp}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. $\text{role}[\text{sID}^*_{\text{resp}}] = \text{"responder"}$ // $G^{\neg sk}_{11,b}$-$G^{\neg sk}_{13,b}$
if $\text{holder}[\text{sID}^*_{\text{init}}] \neq i'$
  return 0
if $\text{sID}^*_{\text{resp}} \neq s'_{\text{resp}}$
  return 0 // $G^{\neg sk}_{11,b}$-$G^{\neg sk}_{13,b}$
return b'

$\text{DER}_{\text{init}}(\text{sID}, M' = (c_i, \tilde{c}))$

if $\text{holder}[\text{sID}] = \bot$ **or** $\text{state}[\text{sID}] = \bot$ **or** $\text{sKey}[\text{sID}] \neq \bot$ **return** $\bot$
$(i, j) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
$(\tilde{sk}, m_j, M := (\tilde{pk}, c_j)) := \text{state}[\text{sID}]$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
if $i = i'$
  if $\tilde{m}' = \bot$
    $K := \mathsf{H}'_{\mathsf{L}1}(c_i, m_j, \tilde{c}, i, j, M, M')$
  else
    if $j = i'$
      $K' := \mathsf{H}^3_{\mathsf{q}}(c_i, c_j, \tilde{m}', i', i', M, M')$
    else
      $K' := \mathsf{H}^1_{\mathsf{q}}(c_i, m_j, \tilde{m}', i', j, M, M')$
else
  $m'_i := \mathsf{Dec}(sk_i, c_i)$
  if $m'_i = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
    if $\tilde{m}' = \bot$
      $K := \mathsf{H}'_{\mathsf{L}1}(c_i, m_j, \tilde{c}, i, j, M, M')$
    else
      $K := \mathsf{H}'_{\mathsf{L}2}(c_i, m_j, \tilde{m}', i, j, M, M')$
  else
    if $\tilde{m}' = \bot$
      $K := \mathsf{H}'_{\mathsf{L}3}(m'_i, m_j, \tilde{c}, i, j, M, M')$
    else
      if $j = i'$
        $K' := \mathsf{H}^2_{\mathsf{q}}(m'_i, c_j, \tilde{m}', i, i', M, M')$
      else
        $K := \mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
$\text{sKey}[\text{sID}] := K$
$\text{received}[\text{sID}] := M'$

$\text{DER}_{\text{resp}}(\text{sID}, M = (\tilde{pk}, c_j))$

if $\text{holder}[\text{sID}] = \bot$ **or** $\text{sKey}[\text{sID}] \neq \bot$ **or** $\text{role}[\text{sID}] = \text{"initiator"}$ **return** $\bot$
$\text{role}[\text{sID}] := \text{"responder"}$
$(j, i) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
if $\text{sID} = s'_{\text{resp}}$
  $c_i \leftarrow \overline{\mathsf{Enc}}(pk_{i'})$ // $G^{\neg sk}_{12,b}$-$G^{\neg sk}_{13,b}$
  if $c_i \in \mathsf{Enc}(pk_{i'}, \mathcal{M}; \mathcal{R})$
    ABORT // $G^{\neg sk}_{13,b}$-$G^{\neg sk}_{13,b}$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
if $j = i'$
  if $i = i'$
    $K' := \mathsf{H}^3_{\mathsf{q}}(c_i, c_j, \tilde{m}, i', i', M, M')$
  else
    $K' := \mathsf{H}^2_{\mathsf{q}}(m_i, c_j, \tilde{m}, i, i', M, M')$
else
  $m'_j := \mathsf{Dec}(sk_j, c_j)$
  if $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
    $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
  else
    if $i' = i$
      $K' := \mathsf{H}^1_{\mathsf{q}}(c_i, m'_j, \tilde{m}, i', j, M, M')$
    else
      $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
$\text{sKey}[\text{sID}] := K'$
$(\text{received}[\text{sID}], \text{sent}[\text{sID}]) := (M, M')$
return M'

$\text{TEST}(\text{sID})$ //only one query

$\text{sID}^* := \text{sID}$
if $\text{sKey}[\text{sID}^*] = \bot$
  return $\bot$
$K^*_0 := \text{sKey}[\text{sID}^*]$ // $G^{\neg sk}_{10,b}$-$G^{\neg sk}_{13,b}$
$K^*_0 \leftarrow_\$ \mathcal{K}$ // $G^{\neg sk}_{13,0}$
$K^*_1 \leftarrow_\$ \mathcal{K}$
return K^*_b

Figure 29: Games $G^{\neg sk}_{10,b}$ - $G^{\neg sk}_{13,b}$ for case ($\neg sk$) of the proof of Lemma 2. Random oracles G and H remain as in game $G^{\neg sk}_{10,b}$.

encryption $c_i \leftarrow \overline{\mathsf{Enc}}(pk_{i'})$. Consider the adversaries $\mathsf{A}^{\neg sk}_{\mathsf{DS},\mathsf{b}}$ given in Figure 30. The running times are the

same as in case ($\neg$st), see Equation (6), page 48:

$$\begin{aligned}\mathrm{Time}(\mathsf{A}^{\neg sk}_{\mathsf{DS},\mathsf{b}}) &\leq \mathrm{Time}(\mathsf{B}) + S \cdot (\mathrm{Time}(\mathsf{KG}) + 3 \cdot \mathrm{Time}(\mathsf{Enc}) + 2 \cdot \mathrm{Time}(\mathsf{Dec})) + q_{\mathsf{H}} + q_{\mathsf{G}} + 4S \\ &\approx \mathrm{Time}(\mathsf{B}) \ .\end{aligned}$$

Since each adversary $\mathsf{A}^{\neg sk}_{\mathsf{DS},\mathsf{b}}$ perfectly simulates game $G^{\neg sk}_{12,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(pk)$, and game $G^{\neg sk}_{11,b}$ if its input c was generated by $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$ for some randomly picked message m,

$$|\Pr[G^{\neg sk\,\mathsf{B}}_{11,b} \Rightarrow 1] - \Pr[G^{\neg sk\,\mathsf{B}}_{12,b} \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg sk}_{\mathsf{DS},\mathsf{b}})$$

for both bits b. Folding $\mathsf{A}^{\neg sk}_{\mathsf{DS},0}$ and $\mathsf{A}^{\neg sk}_{\mathsf{DS},1}$ into one adversary $\mathsf{A}^{\neg sk}_{\mathsf{DS}}$ yields

$$\begin{aligned}|\Pr[G^{\neg sk}_{11,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{11,0} \Rightarrow 1]| \leq{}& |\Pr[G^{\neg sk}_{12,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| \\ &+ 2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg sk}_{\mathsf{DS}}) \ .\end{aligned}$$

$\mathsf{A}^{\neg sk\ |\mathsf{H}'\rangle,|\mathsf{H}_\mathsf{q}\rangle,|\mathsf{G}\rangle}_{\mathsf{DS},\mathsf{b}}(pk, c^*)$

01 $\mathrm{cnt}, \mathrm{sID}^* := 0$
02 $i' \leftarrow_\$ [N]$
03 $s'_{\mathrm{resp}} \leftarrow_\$ [S]$
04 **for** $n \in [N] \setminus \{i'\}$
05 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
06 $(pk_{i'}, sk_{i'}) := (pk, \bot)$
07 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
08 **if** ATTACK(sID^*)
09 $\quad$ **return** 0
10 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
11 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] =$ "initiator"
12 **if** corrupted[holder[$\mathrm{sID}^*_{\mathrm{init}}$]] ABORT
13 Pick $\mathrm{sID}^*_{\mathrm{resp}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{resp}}] =$ "responder"
14 **if** holder[$\mathrm{sID}^*_{\mathrm{init}}$] $\neq i'$
15 $\quad$ **return** 0
16 **if** $\mathrm{sID}^*_{\mathrm{resp}} \neq s'_{\mathrm{resp}}$
17 $\quad$ **return** 0
18 **return** b'

$\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}, M = (\tilde{pk}, c_j))$

19 **if** holder[sID] $= \bot$ **or** sKey[sID] $\neq \bot$ **or** role[sID] = "initiator"
20 $\quad$ **return** $\bot$
21 role[sID] := "responder"
22 $(j, i) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
23 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
24 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
25 **if** sID $= s'_{\mathrm{resp}}$
26 $\quad c_i := c^*$
27 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
28 $M' := (c_i, \tilde{c})$
29 **if** $j = i'$
30 $\quad$ **if** $i = i'$
31 $\quad\quad K' := \mathsf{H}^3_\mathsf{q}(c_i, c_j, \tilde{m}, i', i', M, M')$
32 $\quad$ **else**
33 $\quad\quad K' := \mathsf{H}^2_\mathsf{q}(m_i, c_j, \tilde{m}, i, i', M, M')$
34 **else**
35 $\quad m'_j := \mathsf{Dec}(sk_j, c_j)$
36 $\quad$ **if** $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
37 $\quad\quad K' := \mathsf{H}'_\mathsf{R}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
38 $\quad$ **else**
39 $\quad\quad$ **if** $i' = i$
40 $\quad\quad\quad K' := \mathsf{H}^1_\mathsf{q}(c_i, m'_j, \tilde{m}, i', j, M, M')$
41 $\quad\quad$ **else**
42 $\quad\quad\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
43 sKey[sID] := K'
44 $(\mathrm{received}[\mathrm{sID}], \mathrm{sent}[\mathrm{sID}]) := (M, M')$
45 **return** M'

Figure 30: Adversaries $\mathsf{A}^{\neg sk}_{\mathsf{DS},\mathsf{b}}$ for case ($\neg sk$) of the proof of Lemma 2, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H}_\mathsf{q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for $\mathrm{DER}_{\mathrm{resp}}$ and CORRUPT are defined as in game $G^{\neg sk}_{11,b}$ (see Figure 29).

GAME $G^{\neg sk}_{13,0}$. In game $G^{\neg sk}_{13,0}$, we abort in line 55 if the fake ciphertext c_i that was picked during execution of $\mathrm{DER}_{\mathrm{resp}}(s'_{\mathrm{resp}})$ lies within the range of encryption under $pk_{i'}$, i.e., if $c_i \in \mathsf{Enc}(pk_{i'}, \mathcal{M}; \mathcal{R})$. Since PKE is ϵ_{dis}-disjoint,

$$|\Pr[G^{\neg sk}_{12,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| \leq |\Pr[G^{\neg sk}_{13,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{13,0} \Rightarrow 1]| + 2 \cdot \epsilon_{\mathrm{dis}} \ .$$

GAME $G^{\neg sk}_{13,0}$. In game $G^{\neg sk}_{13,0}$, we change oracle TEST in line 79 such that it returns a random value instead of returning sKey[sID*]. Since games $G^{\neg sk}_{12,1}$ and $G^{\neg sk}_{13,0}$ are equal,

$$|\Pr[G^{\neg sk}_{13,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{13,0} \Rightarrow 1]| = |\Pr[G^{\neg sk}_{13,0} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G^{\neg sk}_{13,0} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]|$. B cannot distinguish the value $K^*_0 =$ sKey[sID*] that is returned by TEST(sID*) from random in game $G^{\neg sk}_{12,0}$ unless it obtains K^*_0 (either classically or contained in a quantum answer) at some point other than during the calling of TEST. This means obtaining K^*_0 by queries to REVEAL or to H.

We will first make explicit how the key is defined: Let j^* denote the peer of $\text{sID}^*_{\text{init}}$. Let m^*_j denote the randomly chosen message with encryption $c^*_j := \mathsf{Enc}(pk_{j^*}, m^*_j; \mathsf{G}(m^*_j))$ that was sampled during execution of INIT($\text{sID}^*_{\text{init}}$). Let $(\tilde{pk}^*, \tilde{sk}^*)$ denote the key pair that was sampled during execution of INIT($\text{sID}^*_{\text{init}}$). Furthermore, let c^*_i denote the fake ciphertext that was sampled under $pk_{i'}$ during execution of $\mathsf{Der}_{\mathsf{resp}}(\text{sID}^*_{\text{resp}})$ (Figure 29, line 53) and let $\tilde{m}^*$ denote the randomly chosen message with encryption $\tilde{c}^* := \mathsf{Enc}(\tilde{pk}, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$ that was picked during execution of $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}})$.

In the case that sID* is a "responder" session,

$$K^*_0 = \begin{cases} \mathsf{H}^3_{\mathsf{q}}(c^*_i, c^*_j, \tilde{m}^*, i', i', (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & j^* = i' \\ \mathsf{H}'_{\mathsf{R}}(m^*_i, c^*_j, \tilde{m}^*, i', j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & j^* \neq i' \text{ and } (m'_j = \bot \text{ or } \mathsf{Enc}(pk_{j^*}, m'_j) \neq c^*_j) \\ \mathsf{H}^1_{\mathsf{q}}(c^*_i, m'_j, \tilde{m}^*, i', j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \text{o.w.} \end{cases} ,$$

where $m'_j := \mathsf{Dec}(sk_{j^*}, c^*_j)$. In the case that sID* is an "initiator" session, we have

$$K^*_0 = \begin{cases} \mathsf{H}'_{\mathsf{L1}}(c^*_i, m^*_j, \tilde{c}^*, \tilde{pk}^*, i', j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \tilde{m}' = \bot \text{ or } \tilde{c}^* \neq \mathsf{Enc}(\tilde{pk}, \tilde{m}'; \mathsf{G}(\tilde{m}')) \\ \mathsf{H}^3_{\mathsf{q}}(c^*_i, c^*_j, \tilde{m}', i', i', (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \tilde{c}^* = \mathsf{Enc}(\tilde{pk}, \tilde{m}'; \mathsf{G}(\tilde{m}')) \text{ and } j^* = i' \\ \mathsf{H}^1_{\mathsf{q}}(c^*_i, m^*_j, \tilde{m}', i', j^*, (\tilde{pk}^*, c^*_j), (c^*_i, \tilde{c}^*)) & \text{o.w.} \end{cases} ,$$

where $m'_i := \mathsf{Dec}(sk_{i'}, c^*_i)$ and $\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c}^*)$.

With an argument similar to case ($\neg$st), we can show that none of the quantum answers of $|\mathsf{H}\rangle$ could contain the session key: In any of the cases, to trigger a query to $|\mathsf{H}_{\mathsf{q}}\rangle$ such that its answer contains K_0, B would need to come up with a message m such that $\mathsf{Enc}(pk_{i'}, m; \mathsf{G}(m)) = c^*_i$. Since we abort if c^*_i lies in the range of $\mathsf{Enc}(pk_{i'}, -; -)$, this is impossible.

Next, we will argue that B obtains K^*_0 by a query to REVEAL with negligible probability, no matter if sID* is an "initiator" session, or if sID* is a "responder" session: B would have to derive the same session key by recreating the test session. (Recall that recreating the key on the other side would result in creation of an additional matching session, and hence, in an abort.)

We first consider the case that sID* is an "initiator" session: To obtain K^*_0 via recreation, B would have to establish and initialize another "initiator" session $s \neq \text{sID}^*$ with holder i^* and peer j^*. The final call to DER_{init} could only result in the same key if INIT(s) had computed the same message M as sID*. This means that it picked the same ephemeral key $\tilde{pk}^*$ as INIT($\text{sID}^*_{\text{init}}$), and additionally, some message m_j such that m_j encrypts to c^*_j, happening with probability at most $(S-2) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc})$. Now assume that sID* is a "responder" session. To obtain K^*_0 via recreation, B would have to establish and derive another "responder" session $s \neq \text{sID}^*$ with holder j^* and peer i'. $\text{DER}_{\text{resp}}(s)$ will only derive the same key as $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}})$ if $\text{DER}_{\text{resp}}(s)$computed the same message M' as sID*. This means that in particular, it picked some message m_i such that m_i encrypts to c^*_i. Since we abort if c^*_i lies in the range of $\mathsf{Enc}(pk_{i'}, -; -)$, this is impossible. Hence, we can upper bound the probability of recreation, and therefore, the game distance, by

$$|\Pr[G^{\neg sk}_{13,0} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| \leq (S-2) \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) \ .$$

Collecting the probabilities, we obtain

$$\begin{aligned} &|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\ &\qquad \leq 2 \cdot SN \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg sk}_{\mathsf{DS}}) + 32N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \\ &\qquad\quad + 2 \cdot SN \cdot \epsilon_{\text{dis}} + S^2 \cdot N \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) \ , \end{aligned}$$

the upper bound we claimed in equation (4).

E Proof of Lemma 3

TAMPERING WITH THE PROTOCOL ($\mathfrak{M}(\text{sID}^*) = \varnothing$). Recall that we are proving an upper bound for $|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing]|$. Therefore, we will first enforce that indeed, we only need to consider the case where $\mathfrak{M}(\text{sID}^*) = \varnothing$. Consider the sequence of games given in Figure 31.

GAMES $G_{0,b}$ - $G_{1,b}$

01 $\text{cnt}, \text{sID}^* := 0$
02 **for** $n \in [N]$
03 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
04 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
05 **if** ATTACK(sID^*)
06 $\quad$ **return** 0
07 **if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT // $G_{1,b}$
08 **return** b'

INIT(sID)

09 **if** holder[sID] $= \perp$ **or** sent[sID] $\neq \perp$ **return** $\perp$
10 role[sID] := "initiator"
11 $i :=$ holder[sID]
12 $j :=$ peer[sID]
13 $m_j \leftarrow_{\$} \mathcal{M}$
14 $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
15 $(\tilde{pk}, \tilde{sk}) \leftarrow \mathsf{KG}$
16 $M := (\tilde{pk}, c_j)$
17 state[sID] $:= (\tilde{sk}, m_j, M)$
18 sent[sID] $:= M$
19 **return** M

$\text{DER}_{\text{resp}}(\text{sID}, M = (\tilde{pk}, c_j))$

20 **if** holder[sID] $= \perp$ **or** sKey[sID] $\neq \perp$ **or** role[sID] = "initiator" **return** $\perp$
21 role[sID] := "responder"
22 $(j, i) :=$ (holder[sID], peer[sID]
23 $m_i, \tilde{m} \leftarrow_{\$} \mathcal{M}$
24 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
25 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
26 $M' := (c_i, \tilde{c})$
27 $m'_j := \mathsf{Dec}(sk_j, c_j)$
28 **if** $m'_j = \perp$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
29 $\quad K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
30 **else** $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
31 sKey[sID] $:= K'$
32 (received[sID], sent[sID]) $:= (M, M')$
33 **return** M'

$\text{DER}_{\text{init}}(\text{sID}, M' = (c_i, \tilde{c}))$

34 **if** holder[sID] $= \perp$ **or** state[sID] $= \perp$ **or** sKey[sID] $\neq \perp$ **return** $\perp$
35 $(i, j) :=$ (holder[sID], peer[sID])
36 $(\tilde{sk}, m_j, M := (\tilde{pk}, c_j)) :=$ state[sID]
37 $m'_i := \mathsf{Dec}(sk_i, c_i)$
38 $\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
39 **if** $m'_i = \perp$ **or** $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
40 $\quad$ **if** $\tilde{m}' = \perp$
41 $\quad\quad K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$
42 $\quad$ **else**
43 $\quad\quad K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', i, j, M, M')$
44 **else if** $\tilde{m}' = \perp$
45 $\quad\quad K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, i, j, M, M')$
46 **else** $K := \mathsf{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
47 sKey[sID] $:= K$
48 received[sID] $:= M'$

Figure 31: Games $G_{0,b}$ - $G_{1,b}$ for case two of the proof of Theorem 3. Helper procedure ATTACK and oracles TEST, EST, CORRUPT, REVEAL and REV-STATE remains as in the original IND-StAA game (see Figures 12 and 13).

GAMES $G_{0,b}$. Since for both bits b, game $G_{0,b}$ is the original game $\mathsf{IND\text{-}StAA}_b$,

$$\begin{aligned}&|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing]|\\ &= |\Pr[G_{0,1}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing] - \Pr[G_{0,0}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing]| \ .\end{aligned}$$

GAMES $G_{1,b}$. Both games $G_{1,b}$ abort in line 07 if $\mathfrak{M}(\text{sID}^*) \neq \varnothing$. Since for both bits b it holds that $\Pr[G_{1,b}^{\mathsf{B}} \Rightarrow 1] = \Pr[G_{0,b}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing]$,

$$\begin{aligned}&|\Pr[G_{0,1}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing] - \Pr[G_{0,0}^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) = \varnothing]|\\ &= |\Pr[G_{1,1}^{\mathsf{B}} \Rightarrow 1] - \Pr[G_{1,0}^{\mathsf{B}} \Rightarrow 1]| \ .\end{aligned}$$

To upper bound $|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1]|$, we will examine both the case that role[sID*] = "initiator", called case (init), and the case that role[sID*] = "responder", called case (resp). Since cases (init) and (resp) are mutually exclusive,

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1]| \\
&\leq |\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \text{role[sID}^*\text{]} = \text{"initiator"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \text{role[sID}^*\text{]} = \text{"initiator"}]| \\
&+ |\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \text{role[sID}^*\text{]} = \text{"responder"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \text{role[sID}^*\text{]} = \text{"responder"}]| \ .
\end{aligned}$$

As discussed below Definition 8, B's bit only counts in game $\mathsf{IND\text{-}StAA}$ (and hence, in game $G_{1,b}$) if no attack was executed that we ruled out by method ATTACK: Since we examine the case that no matching session exists, ATTACK returns **true** if B obtained the test session's internal state or the secret key of its peer.

Case (init). Intuition is as follows: While B interacts with "initiator" session sID* and could pick message $(c_i, \tilde{c})$ on its own (thereby being able to control both corresponding messages m'_j and $\tilde{m}'$), the test session's peer (henceforth called j^*) remains uncorrupted throughout the game, and also the test session's internal state remains unrevealed. Therefore, at least message m^*_j that was randomly picked by INIT(sID*) can not be obtained trivially, and therefore, ciphertext c^*_j can be faked.

Consider the sequence of games given in Figures 32 and 34: First, we will enforce that indeed, we are in the case where sID* is an "initiator" session. Afterwards, we ensure that the game makes no use of the peer's secret key any longer by plugging encryption into the random oracle (in games $G^{\text{init}}_{2,b}$ to $G^{\text{init}}_{8,b}$, see Figure 32). Again, this is the only part of the proof where the correctness error comes into play. Next, during execution of INIT(sID*), we replace ciphertext c^*_j with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G^{\text{init}}_{9,b}$ to $G^{\text{init}}_{10,b}$, see Figure 34, line 26). We show that after those changes, B's view does not change with overwhelming probability if we finally change TEST such that it always returns a random value (game $G^{\text{init}}_{12,b}$, also Figure 34).

Figure 32: Games $G^{\text{init}}_{1,b}$ - $G^{\text{init}}_{8,b}$ for case (init) of the proof of Lemma 3. Helper procedure ATTACK and oracles INIT, TEST, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 12 and Figure 13, pages 21 and 22).

GAME $G^{\text{init}}_{1,b}$. Since game $G^{\text{init}}_{1,b}$ is equal to game $G_{1,b}$,

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}]|\\ &= |\Pr[G^{\text{init}\,\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}] - \Pr[G^{\text{init}\,\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}]| \ .\end{aligned}$$

GAMES $G^{\text{init}}_{2,b}$. Both games $G^{\text{init}}_{2,b}$ abort in line 12 if $\text{role}[\text{sID}^*] = $ "responder". Since for both bits b it holds

that $\Pr[G^{\text{init}\,\mathsf{B}}_{1,b} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}] = \Pr[G^{\text{init}\,\mathsf{B}}_{2,b} \Rightarrow 1]$,

$$\begin{aligned}|\Pr[G^{\text{init}\,\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}] - \Pr[G^{\text{init}\,\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \text{role}[\text{sID}^*] = \text{"initiator"}]| \\ = |\Pr[G^{\text{init}}_{2,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{2,0} \Rightarrow 1]| \ .\end{aligned}$$

The first goal is not to have to make use of the test session's peer's secret key any longer. Since peer $j^* = \text{peer}[\text{sID}^*]$ is not fixed until B issues the TEST query, we first add a guess j'. Afterwards, we patch encryption into H for the first two messages, and even out derivation for ciphertexts with decryption failure and for ciphertexts without. Like in case $(\neg sk)$, these changes do not affect B's view unless it is able to distinguish random oracle G from an oracle $\mathsf{G}_{pk,sk}$ that only samples randomness under which decryption never fails, allowing for a reduction to game GDPB.

Games $G^{\text{init}}_{3,b}$. In games $G^{\text{init}}_{3,b}$, one of the parties is picked at random in line 03, and the games return 0 in line 14 if any other party j' was picked than the test session's peer.

$$\Pr[G^{\text{init}\,\mathsf{B}}_{2,b} \Rightarrow 1] = N \cdot \Pr[G^{\text{init}\,\mathsf{B}}_{3,b} \Rightarrow 1] \ .$$

To prepare getting rid of $sk_{j'}$, we will first modify random oracle G such that it renders PKE perfectly correct for key pair $(pk_{j'}, sk_{j'})$.

Game $G^{\text{init}}_{4,b}$. In game $G^{\text{init}}_{4,b}$, we enforce that no decryption failure with respect to key pair $(pk_{j'}, pk_{j'})$ will occur: We replace random oracle G with $\mathsf{G}_{pk_{j'},sk_{j'}}$ in line 06, where $\mathsf{G}_{pk_{j'},pk_{j'}}$ is defined in line 70. To upper bound $|\Pr[G^{\text{init}\,\mathsf{B}}_{3,b} \Rightarrow 1] - \Pr[G^{\text{init}\,\mathsf{B}}_{4,b} \Rightarrow 1]|$ for each bit b, we construct (unbounded, quantum) adversaries C^{b} against the generic distinguishing problem with bounded probabilities GDPB_{λ} in Figure 33, issuing at most $q_{\mathsf{G}} + 3S$ queries to $|\mathsf{F}\rangle$. With the same analysis as in our proofs for cases $(\neg\text{st})$ and $(\neg sk)$(see pages 44 and 54),

$$|\Pr[G^{\text{init}\,\mathsf{B}}_{3,b} \Rightarrow 1] - \Pr[G^{\text{init}\,\mathsf{B}}_{4,b} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \ ,$$

and according to Lemma 4,

$$\Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}^{\mathsf{b}}}_{\lambda,0} = 1]| \leq 8 \cdot (q_{\mathsf{G}} + 3S + 1)^2 \cdot \delta \ .$$

$\underline{\mathsf{C}^{\mathsf{b}}_1 = \mathsf{D}^{\mathsf{b}}_1}$
01 $(pk, sk) \leftarrow \mathsf{KG}$
02 **for** $m \in \mathcal{M}$
03 $\quad \lambda(m) := \delta(pk, sk, m)$
04 **return** $(\lambda(m))_{m \in \mathcal{M}}$

$\underline{\mathsf{G}(m)}$
05 **if** $\mathsf{F}(m) = 0$
06 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
07 **else**
08 $\quad \mathsf{G}(m) := \mathsf{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
09 **return** $\mathsf{G}(m)$

$\underline{\mathsf{C}^{\mathsf{b}|\mathsf{F}\rangle}_2, \mathsf{D}^{\mathsf{b}|\mathsf{F}\rangle}_2}$
10 $\text{cnt}, \text{sID}^* := 0$
11 $j' \leftarrow_{\$} [N]$
12 **for** $n \in [N]$
13 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
14 $b' \leftarrow \mathsf{B}^{\mathsf{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
15 **if** ATTACK(sID^*)
16 $\quad$ **return** 0
17 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
18 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$
$\quad$ s. th. $\text{role}[\text{sID}^*_{\text{init}}] = \text{"initiator"}$
19 **if** $\text{corrupted}[\text{holder}[\text{sID}^*_{\text{init}}]]$ ABORT
20 **if** $\text{holder}[\text{sID}^*_{\text{init}}] \neq j'$
21 $\quad$ **return** 0
22 **return** b'

Figure 33: Unbounded quantum adversaries C^{b} and D^{b}, executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$, for case $(\neg sk)$ of the proof of Lemma 2. The adversaries only differ in their definition of DER_{resp}, DER_{init} and H: For adversaries C^{b}, DER_{resp}, DER_{init} and H are defined as in game $G^{\text{init}}_{3,b}$, see Figure 32, while for adversaries D^{b}, DER_{resp} and DER_{init} and H are defined as in game $G^{\text{init}}_{7,b}$ (also Figure 32).

Recall that the goal is to simulate the game without knowledge of $sk_{j'}$. To this end, we will first change the key derivation procedure DER_{init} for holder j' as follows: If ciphertext $\tilde{c}$ already induces

decryption failure, we will not have to use $sk_{j'}$ any more to check whether c_j induces de- or reencryption failure as well.

GAMES $G^{\text{init}}_{5,b}$. In both games $G^{\text{init}}_{5,b}$, we change oracle DER_{init} in line 34 for session holder j' as follows: Whenever $\tilde{c}$ does not decrypt to a message $\tilde{m}'$ s. th. $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}', \mathsf{G}(\tilde{m}'))$, the session key is defined as $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$. (Before this change we let $K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, i, j, M, M')$ in the case that $\tilde{c}$ fails to decrypt, but c_i decrypts correctly.) Due to the same argument as for games $G^{\neg sk}_{6,b}$ (that the cases are logically distinct, see page 54), B's view does not change and

$$\Pr[G^{\text{init}}{}^{\mathsf{B}}_{4,b} \Rightarrow 1] = \Pr[G^{\text{init}}{}^{\mathsf{B}}_{5,b} \Rightarrow 1] \ .$$

In the next game, we change key definition of DER_{init} if both ciphertexts de- and reencrypt correctly, and key definition of DER_{resp} if c_j de-and reencrypts correctly. In these cases, we do note use the decryptions under $sk_{j'}$, but the ciphertexts themself. Similar to games $G^{\neg sk}_{7,b}$ (see page 55), we "plug in" encryption into random oracle H whenever j' appears as one of the involved parties.

GAMES $G^{\text{init}}_{6,b}$. In games $G^{\text{init}}_{6,b}$, the random oracle is changed as follows: Instead of picking H uniformly random, we pick four random oracles $\mathsf{H}^1_{\mathsf{q}}$ to $\mathsf{H}^3_{\mathsf{q}}$, and H', and define

$$\mathsf{H}(m_1,m_2, m_3, i, j, M, M') := \begin{cases} \mathsf{H}^1_{\mathsf{q}}(\mathsf{Enc}(pk_{j'}, m_1; \mathsf{G}(m_1)), m_2, m_3, j', j, M, M') & i = j' \wedge j \neq j' \\ \mathsf{H}^2_{\mathsf{q}}(m_1, \mathsf{Enc}(pk_{j'}, m_2; \mathsf{G}(m_2)), m_3, i, j', M, M') & i \neq j' \wedge j = j' \\ \mathsf{H}^3_{\mathsf{q}}(\mathsf{Enc}(pk_{j'}, m_1; \mathsf{G}(m_1)), \mathsf{Enc}(pk_{j'}; \mathsf{G}(m_2)), m_2, m_3, j', j, M, M') & i = j = j' \\ \mathsf{H}(m_1, m_2, m_3, i, j, M, M') & \text{o.w.} \end{cases} ,$$

see lines 72 to 78. Again, since G only samples from good randomness, encryption under public key $pk_{j'}$ is rendered perfectly correct and hence, injective. Since encryption under public key $pk_{j'}$ is injective, H still is uniformly random.

Like in games $G^{\neg sk}_{7,b}$, we also make the change of H explicit in the derivation oracles (see lines 62 to 66 and lines 38 to 42). Due to the same argument as for games $G^{\neg sk}_{7,b}$ (since m_i encrypts to c_i, or respectively, m_j encrypts to c_i, and we are in the cases where m'_j reencrypts to c_j, or respectively, m'_i reencrypts to c_i, see page 55), all changes are purely conceptual and

$$\Pr[G^{\text{init}}{}^{\mathsf{B}}_{5,b} \Rightarrow 1] = \Pr[G^{\text{init}}{}^{\mathsf{B}}_{6,b} \Rightarrow 1] \ .$$

The final step to get rid of $sk_{j'}$ is to even out the key derivation ciphertexts that are problematic with respect to secret key $sk_{j'}$: Similar to games $G^{\neg sk}_{8,b}$ (see page 57), we also use $\mathsf{H}^1_{\mathsf{q}}$ to $\mathsf{H}^3_{\mathsf{q}}$ if a ciphertext fails to de-or reencrypt under $sk_{j'}$, instead of using the implicit reject oracles.

GAMES $G^{\text{init}}_{7,b}$. In games $G^{\text{init}}_{7,b}$, we change DER_{resp} in lines 56 to 58 such that if c_j fails to de- or reencrypt and j' is the session's holder, the session key is defined relative to the random oracles $\mathsf{H}^2_{\mathsf{q}}$ or $\mathsf{H}^3_{\mathsf{q}}$ instead of rejecting implicitly, just as if c_j had reencrypted correctly. Likewise, we change DER_{init} in lines 27 to 29 such that if c_i fails to de- or reencrypt and j' is the session's holder, but ciphertext $\tilde{c}$ does not decrypt to $\perp$, the session key is defined relative to the random oracles $\mathsf{H}^1_{\mathsf{q}}$ or $\mathsf{H}^3_{\mathsf{q}}$ instead of rejecting implicitly, just as if c_i had reencrypted correctly. Again, B could only tell the games apart by either

- establishing and revealing two matching sessions such that the keys mismatched in the previous due to an implicit reject, while in game $G^{\text{init}}_{7,b}$, this difference is evened out by the changes described above, or by
- establishing a session such that its key resulted from an implicit reject in the previous game, while the key can be linked to random oracle H in game $G^{\text{init}}_{7,b}$.

Due to the same argument as for games $G^{\neg sk}_{8,b}$ ($\mathsf{G}_{pk_{j'}, sk_{j'}}$ only samples good randomness, hence all mismatching keys are kept and not linked to random oracle values), see page 57, B's view is identical in both games and

$$\Pr[G^{\text{init}}{}^{\mathsf{B}}_{6,b} \Rightarrow 1] = \Pr[G^{\text{init}}{}^{\mathsf{B}}_{7,b} \Rightarrow 1] \ .$$

GAME $G^{\text{init}}_{8,b}$. In game $G^{\text{init}}_{8,b}$, we switch back to using $\mathsf{G} \leftarrow_{\$} \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{pk_{j'},sk_{j'}}$. With the same reasoning as for the gamehop from game $\Pr[G^{\text{init}\,\mathsf{B}}_{3,b} \Rightarrow 1]$ to $\Pr[G^{\text{init}\,\mathsf{B}}_{4,b} \Rightarrow 1]$,

$$\begin{aligned}|\Pr[G^{\text{init}\,\mathsf{B}}_{7,b} \Rightarrow 1] - \Pr[G^{\text{init}\,\mathsf{B}}_{8,b} \Rightarrow 1]| &= |\Pr[\mathsf{GDPB}^{\mathsf{D}'}_{\lambda,0} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{D}'}_{\lambda,1} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S)^2 \cdot \delta \ ,\end{aligned}$$

where adversaries D^{b} also are given in Figure 33.

To upper bound $|\Pr[G^{\text{init}}_{8,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{8,0} \Rightarrow 1]|$, consider the sequence of games given in Figure 34.

GAMES $G^{\text{init}}_{8,b}$-$G^{\text{init}}_{12,b}$

```
01 cnt, sID* := 0
02 j' ←$ [N]
03 for n ∈ [N]
04    (pk_n, sk_n) ← KG
05 s' ←$ [S]                    // G^init_{9,b}-G^init_{12,b}
06 b' ← B^{O,|G⟩,|H⟩}((pk_n)_{n∈[N]})
07 if ATTACK(sID*)
08    return 0
09 if M(sID*) ≠ ∅ ABORT
10 if role[sID*] = "responder"
11    ABORT
12 if peer[sID*] ≠ j'
13    return 0
14 if peer[sID*] ≠ j'
15    return 0
16 if sID* ≠ s'
17    return 0                  // G^init_{9,b}-G^init_{12,b}
18 return b'
```

INIT(sID)

```
19 if holder[sID] = ⊥ or sent[sID] ≠ ⊥
20    return ⊥
21 role[sID] := "initiator"
22 i := holder[sID], j := peer[sID]
23 m_j ←$ M
24 c_j := Enc(pk_j, m_j; G(m_j))
25 if sID = s'
26    c_j ← Enc(pk_{j'})        // G^init_{10,b}-G^init_{12,b}
27    if c_j ∈ Enc(pk_{j'}, M; R)
28       ABORT                  // G^init_{11,b}-G^init_{12,b}
29 (s̃k, p̃k) ← KG
30 M := (p̃k, c_j)
31 state[sID] := (s̃k, m_j, M)
32 sent[sID] := M
33 return M
```

TEST(sID) //only one query

```
34 sID* := sID
35 if sKey[sID*] = ⊥ return ⊥
36 K*_0 := sKey[sID*]          // G^init_{8,b}-G^init_{10,b}
37 K*_0 ←$ K                   // G^init_{12,0}
38 K*_1 ←$ K
39 return K*_b
```

Figure 34: Games $G^{\text{init}}_{8,b}$ - $G^{\text{init}}_{12,b}$ for case (init) of the proof of Lemma 3. All oracles except for INIT and TEST remain as in game $G^{\text{init}}_{8,b}$ (see Figure 32).

We stress that from game $G^{\text{init}}_{8,b}$ on, none of the oracles uses $sk_{j'}$ any longer: DER_{resp} and DER_{init} were changed accordingly, B would trivially lose if it ever queried CORRUPT on j'. Since we want to replace the test session's ciphertext c_j with a fake encryption, we first have to add a guess for sID^*.

GAMES $G^{\text{init}}_{9,b}$. In games $G^{\text{init}}_{9,b}$, one of the sessions that gets established during execution of B is picked at random in line 05, and the game returns 0 in line 17 if any other session s' was picked than test session sID^*.

$$\Pr[G^{\text{init}\,\mathsf{B}}_{8,b} \Rightarrow 1] = S \cdot \Pr[G^{\text{init}\,\mathsf{B}}_{9,b} \Rightarrow 1] \ .$$

GAMES $G^{\text{init}}_{10,b}$. In games $G^{\text{init}}_{10,b}$, oracle INIT is changed in line 26 such that for s', c_j is no longer a ciphertext of the form $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$ for some randomly drawn message m_j, but a fake encryption $c_j \leftarrow \overline{\mathsf{Enc}}(pk_{j'})$. Consider the adversaries $\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}$ given in Figure 35. The running time is the same as in case ($\neg$st), see Equation (6):

$$\begin{aligned}\text{Time}(\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}) &\leq \text{Time}(\mathsf{B}) + S \cdot (\text{Time}(\mathsf{KG}) + 3 \cdot \text{Time}(\mathsf{Enc}) + 2 \cdot \text{Time}(\mathsf{Dec})) + q_{\mathsf{H}} + q_{\mathsf{G}} + 4S \\ &\approx \text{Time}(\mathsf{B}) \ .\end{aligned}$$

Since each adversary $\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}$ perfectly simulates game $G^{\text{init}}_{10,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(pk)$, and game $G^{\text{init}}_{9,b}$ if its input c was generated by $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$ for some randomly picked message m,

$$|\Pr[G^{\text{init}}_{9,b} \Rightarrow 1] - \Pr[G^{\text{init}}_{10,b} \Rightarrow 1]| = \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}) \ ,$$

for both bits b. Folding $\mathsf{A}^{\text{init}}_{\mathsf{DS},0}$ and $\mathsf{A}^{\text{init}}_{\mathsf{DS},1}$ into one adversary $\mathsf{A}^{\text{init}}_{\mathsf{DS}}$ yields

$$|\Pr[G^{\text{init}}_{9,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{9,0} \Rightarrow 1]| \leq |\Pr[G^{\neg sk}_{10,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{10,0} \Rightarrow 1]| + 2 \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg sk}_{\mathsf{DS}}) \ .$$

$\underline{\mathsf{A}^{\text{init}\ |\mathsf{H}'\rangle,|\mathsf{H}_\mathsf{q}\rangle,|\mathsf{G}\rangle}_{\mathsf{DS},\mathsf{b}}(pk, c)}$
01 $\text{cnt}, \text{sID}^* := 0$
02 $j' \leftarrow_\$ [N]$
03 $s' \leftarrow_\$ [S]$
04 **for** $n \in [N] \setminus \{j'\}$
05 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
06 $(pk_{j'}, sk_{j'}) := (pk, \bot)$
07 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{RO}\rangle}(pk_1, \cdots, pk_N)$
08 **if** ATTACK(sID^*)
09 $\quad$ **return** 0
10 **if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT
11 **if** $\text{role}[\text{sID}^*] =$ "responder"
12 $\quad$ ABORT
13 **if** $\text{peer}[\text{sID}^*] \neq j'$ **return** 0
14 **if** $\text{peer}[\text{sID}^*] \neq j'$ **return** 0
15 **if** $\text{sID}^* \neq s'$ **return** 0
16 **return** b'

$\underline{\text{CORRUPT}(i \in [N] \setminus \{j'\})}$
17 **if** $\text{corrupted}[i]$ **return** $\bot$
18 $\text{corrupted}[i] :=$ **true**
19 **return** sk_i

$\underline{\text{INIT}(\text{sID})}$
20 **if** $\text{holder}[\text{sID}] = \bot$
21 $\quad$ **return** $\bot$
22 **if** $\text{sent}[\text{sID}] \neq \bot$
23 $\quad$ **return** $\bot$
24 $\text{role}[\text{sID}] :=$ "initiator"
25 $i := \text{holder}[\text{sID}]$
26 $j := \text{peer}[\text{sID}]$
27 $m_j \leftarrow_\$ \mathcal{M}$
28 $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
29 **if** $\text{sID} = s'$
30 $\quad c_j := c$
31 $(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{KG}$
32 $M := (\tilde{pk}, c_j)$
33 $\text{state}[\text{sID}] := (\tilde{sk}, m_j, M)$
34 $\text{sent}[\text{sID}] := M$
35 **return** M

Figure 35: Adversaries $\mathsf{A}^{\text{init}}_{\mathsf{DS},\mathsf{b}}$ for case (init) of the proof of Lemma 3, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H}_\mathsf{q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for INIT and CORRUPT are defined as in game $G^{\text{init}}_{9,b}$ (see Figure 34).

Game $G^{\text{init}}_{11,b}$. In game $G^{\text{init}}_{11,b}$, we abort in line 28 if the fake ciphertext c_j that was picked during execution of $\text{INIT}(s'_{\text{init}})$ lies within the range of encryption under $pk_{j'}$, i.e., if $c_j \in \mathsf{Enc}(pk_{j'}, \mathcal{M}; \mathcal{R})$. Since PKE is ϵ_{dis}-disjoint,

$$|\Pr[G^{\text{init}}_{10,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{10,0} \Rightarrow 1]| \leq |\Pr[G^{\text{init}}_{11,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{11,0} \Rightarrow 1]| + 2 \cdot \epsilon_{\text{dis}} \ .$$

Game $G^{\text{init}}_{12,b}0$. In game $G^{\text{init}}_{12,0}$, we change oracle TEST in line 37 such that it returns a random value instead of $\text{sKey}[\text{sID}^*]$. Again,

$$|\Pr[G^{\text{init}}_{10,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{10,0} \Rightarrow 1]| = |\Pr[G^{\text{init}\ \mathsf{B}}_{12,0} \Rightarrow 1] - \Pr[G^{\text{init}\ \mathsf{B}}_{11,0} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G^{\text{init}\ \mathsf{B}}_{12,0} \Rightarrow 1] - \Pr[G^{\text{init}\ \mathsf{B}}_{11,0} \Rightarrow 1]|$, which means upper bounding the probability that B obtained the test session's key in game $G^{\text{init}}_{11,0}$ by queries to REVEAL or to H.

Note that similar to previous cases, since both M and M' are hashed, B could not create a "responder" session that derives the same key without creating a match and hence, triggering an abort. Furthermore, to create an "initiator" session that derives the same key, its initialisation would have to output the same message $M = (\tilde{pk}^*, c^*_j)$ that was returned by the initialisation of sID^*. But since c^*_j is a fake encryption and we abort if c^*_j lies within the range of $\mathsf{Enc}(pk_{j'}, -; -)$, this is impossible.

With an argument similar to the previous cases, we can show that none of the quantum answers of $|\mathsf{H}\rangle$ could contain the session key: Since the test session's peer is j', encryption is plugged in for the second argument of H. To trigger a query to $|\mathsf{H}_\mathsf{q}\rangle$ such that its answer contains the key, B would need to come up with a message m such that $\mathsf{Enc}(pk_{j'}, m; \mathsf{G}(m)) = c^*_j$, which is impossible.

$$\Pr[G^{\text{init}\ \mathsf{B}}_{12,0} \Rightarrow 1] = \Pr[G^{\text{init}\ \mathsf{B}}_{11,0} \Rightarrow 1] \ .$$

Collecting the probabilities, we obtain

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}]| \\ &\quad \leq 2 \cdot SN \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\mathrm{init}}_{\mathsf{DS}}) + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta + 2 \cdot SN \cdot \epsilon_{\mathrm{dis}} \ .\end{aligned}$$

Case (resp). Intuition is as follows: While B could pick message $(c_j, \tilde{pk})$ on its own (thereby being able to control both m_j and $\tilde{m}$), the test session's peer remains uncorrupted throughout the game. Therefore, at least message m_i (that was randomly picked by $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*, (c_j, \tilde{pk}))$) cannot be computed trivially. The proof differs from case (init) only in the following way: instead of changing $\mathrm{INIT}(\mathrm{sID}^*)$ such that it outputs a fake encryption c_j, we change $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*, m)$ such that it outputs a fake encryption c_i. In the last game, c_i does not lie in the range of $\mathsf{Enc}(pk_{i'}, -; -))$ any more. Therefore, it is impossible to recreate the test session, hence the key can neither be revealed nor hit by a random oracle query. We obtain the same upper bound: there exists an adversary $\mathsf{A}^{\mathrm{resp}}_{\mathsf{DS}}$ such that

$$\begin{aligned}&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}]| \\ &\quad \leq 2 \cdot SN \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\mathrm{resp}}_{\mathsf{DS}}) + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta + 2 \cdot SN \cdot \epsilon_{\mathrm{dis}} \ .\end{aligned}$$

Collecting the probabilities, and folding $\mathsf{A}^{\mathrm{init}}_{\mathsf{DS}}$ and $\mathsf{A}^{\mathrm{resp}}_{\mathsf{DS}}$ into one adversary A', we obtain

$$\begin{aligned}&|\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing]| \\ &\quad \leq 4 \cdot SN \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}') + 64 \cdot N \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S)^2 \cdot \delta + 4 \cdot SN\epsilon_{\mathrm{dis}} \ ,\end{aligned}$$

the upper bound bound given in Lemma 3.

F IND-StAA AKE without disjoint simulatability.

Recall that transformation Punc punctures the message space at one message and samples encryptions of this message as fake encryptions, see Figure 4, and that plugging transformation Punc into transformation T achieves DS and CPA security from CPA security (see Appendix D):

Corollary F.1 (DS security of TPunc). *For all adversaries A issuing at most q_{G} queries to $|\mathsf{G}\rangle$, there exist two adversaries B_1 and B_2 such that*

$$\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0,\hat{m}],\mathsf{G}]}(\mathsf{A}) \leq \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2 \cdot \sqrt{q_{\mathsf{G}} \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4q_{\mathsf{G}}^2}{|\mathcal{M}| - 1}} \ ,$$

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

F.1 Proof of Theorem 4

The following theorem establishes that $\mathsf{FO}_{\mathsf{AKE}} \circ \mathsf{Punc}$ turns $\mathsf{IND\text{-}CPA}$ security into $\mathsf{IND\text{-}StAA}$ security, in the quantum random oracle model, as long as PKE is γ-spread.

Theorem (CCA security of $\mathsf{FO}^{\not\perp}_{\hat{m}} \circ \mathsf{Punc.}$). Assume PKE_0 to be δ-correct and γ-spread, and let $\hat{m} \in \mathcal{M}$. Let $\mathsf{AKE} := \mathsf{FO}_{\mathsf{AKE}}[\mathsf{Punc}[\mathsf{PKE}, \hat{m}], \mathsf{G}, \mathsf{H}]$. Then, for any $\mathsf{IND\text{-}StAA}$ 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 exist adversaries B_1 and B_2 such that

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{B}) \leq\ & 2S \cdot (S + 3 \cdot N) \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2)}\right) \\ &+ (S + 3N) \cdot \left(8q^2 \cdot (S + 4) + S\right) \cdot \delta + S \cdot (S + 3N) \cdot 2^{-\gamma} \\ &+ \frac{S(8q \cdot (S + 3N) + S^2)}{\sqrt{|\mathcal{M}| - 1}} + S \cdot (3S^2 + 2) \cdot \mu(\mathsf{KG}) \ ,\end{aligned}$$

and the running time of B_1 and B_2 is about that of B.

Proof. Similar to our KEM proof given in D, we can proceed in any of the cases until just before the last gamehop, and achieve the upper bounds

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \\
&\qquad \leq S^2 \cdot |\Pr[G^{\neg\mathrm{st}}_{12,0} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{12,1} \Rightarrow 1]| + 2S^2 \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\neg\mathrm{st}}) \\
&\qquad + 4S^2 \cdot \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 1) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\neg\mathrm{st}}) + \frac{4(q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 1)^2}{|\mathcal{M}| - 1}} \\
&\qquad + 32 \cdot S \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 1)^2 \cdot \delta + 2S^2 \cdot \mu(\mathsf{KG}) \ ,
\end{aligned}$$

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\
&\qquad \leq SN \cdot |\Pr[G^{\neg sk}_{12,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2 \cdot \delta \\
&\qquad + 2 \cdot SN \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\neg sk}) \\
&\qquad\quad + 4SN \cdot \sqrt{(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\neg sk}) + \frac{4(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2}{|\mathcal{M}| - 1}}) \ ,
\end{aligned}$$

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}]| \\
&\qquad \leq SN \cdot |\Pr[G^{\mathrm{init}}_{10,1} \Rightarrow 1] - \Pr[G^{\mathrm{init}}_{10,0} \Rightarrow 1]| + 2 \cdot SN \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\mathrm{init}}) \\
&\qquad\quad + 4SN \cdot \sqrt{(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\mathrm{init}}) + \frac{4(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2}{|\mathcal{M}| - 1}} \\
&\qquad + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ ,
\end{aligned}$$

and

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}]| \\
&\qquad \leq SN \cdot |\Pr[G^{\mathrm{resp}}_{10,1} \Rightarrow 1] - \Pr[G^{\mathrm{resp}}_{10,0} \Rightarrow 1]| + 2 \cdot SN \cdot (\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) \\
&\qquad\quad + 4SN \cdot \sqrt{(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2) + \frac{4(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S + 1)^2}{|\mathcal{M}| - 1}} \\
&\qquad + 32 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ .
\end{aligned}$$

Since we do not want to make use of ϵ_{dis}-disjointness, we will use PKE is γ-spread and hence, the probability that $\hat{m}$ deterministically encrypts to c^* is negligible. Switching G to its correctness-inducing version renders c^* being hit by any query to H impossible unless c^* decrypts incorrectly, which happens with probability δ. We obtain

$$|\Pr[G^{\neg\mathrm{st}}_{12,0} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{12,1} \Rightarrow 1]| \leq 2^{-\gamma} + (8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 2)^2 + 1) \cdot \delta + \frac{S}{|\mathcal{M}| - 1} \ ,$$

$$\begin{aligned}
|\Pr[G^{\neg sk}_{12,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{12,0} \Rightarrow 1]| &\leq 2^{-\gamma} + (8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 2)^2 + 1) \cdot \delta \\
&\quad + S\mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) \ ,
\end{aligned}$$

and

$$\begin{aligned}
&|\Pr[G^{\mathrm{init}}_{10,1} \Rightarrow 1] - \Pr[G^{\mathrm{init}}_{10,0} \Rightarrow 1]|, |\Pr[G^{\mathrm{resp}}_{10,1} \Rightarrow 1] - \Pr[G^{\mathrm{resp}}_{10,0} \Rightarrow 1]| \\
&\qquad\qquad \leq 2^{-\gamma} + (8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 2)^2 + 1) \cdot \delta \ .
\end{aligned}$$

Collecting the probabilities and letting $q := q_{\mathsf{G}} + q_{\mathsf{H}} + 3S + 2$, we obtain

$$\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \\
&\qquad \leq 2S^2 \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\neg\mathrm{st}}) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\neg\mathrm{st}})} \right) \\
&\qquad + S \cdot (8q^2 \cdot (S + 4) + S) \cdot \delta + S^2 \cdot (2^{-\gamma} + 2 \cdot \mu(\mathsf{KG})) + \frac{S^2 \cdot (S + 8q)}{\sqrt{|\mathcal{M}| - 1}} \ ,
\end{aligned}$$

$$
\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| \\
&\qquad \leq 2SN \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\neg sk}) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\neg sk})} \right) \\
&\qquad + N \cdot \big(8q^2 \cdot (S+4) + S\big) \cdot \delta + SN \cdot \left(2^{-\gamma} + S \cdot \mu(\mathsf{KG}) \cdot \mu(\mathsf{Enc}) + \frac{8 \cdot q}{\sqrt{|\mathcal{M}| - 1}} \right) \quad ,
\end{aligned}
$$

$$
\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"initiator"}]| \\
&\qquad \leq 2SN \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\mathrm{init}}) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\mathrm{init}})} \right) \\
&\qquad + N \cdot \big(8q^2 \cdot (S+4) + S\big) \cdot \delta + SN \cdot \left(2^{-\gamma} + \frac{8 \cdot q}{\sqrt{|\mathcal{M}| - 1}} \right) \quad ,
\end{aligned}
$$

and

$$
\begin{aligned}
&|\Pr[G^{\mathsf{B}}_{1,1} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}] - \Pr[G^{\mathsf{B}}_{1,0} \Rightarrow 1 \wedge \mathrm{role}[\mathrm{sID}^*] = \text{"responder"}]| \\
&\qquad \leq 2SN \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1^{\mathrm{resp}}) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2^{\mathrm{resp}})} \right) \\
&\qquad + N \cdot \big(8q^2 \cdot (S+4) + S\big) \cdot \delta + SN \cdot \left(2^{-\gamma} + \frac{8 \cdot q}{\sqrt{|\mathcal{M}| - 1}} \right) \quad .
\end{aligned}
$$

Folding the adversaries, we obtain

$$
\begin{aligned}
&|\Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_1 \Rightarrow 1] - \Pr[\mathsf{IND\text{-}StAA}^{\mathsf{B}}_0 \Rightarrow 1]| \\
&\qquad \leq 2S \cdot (S + 3 \cdot N) \cdot \left(\mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_1) + 2\sqrt{q \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}_0}(\mathsf{B}_2)} \right) \\
&\qquad + (S + 3N) \cdot \big(8q^2 \cdot (S+4) + S\big) \cdot \delta + S \cdot (S + 3N) \cdot 2^{-\gamma} \\
&\qquad\quad + \frac{S(8q \cdot (S + 3N) + S^2)}{\sqrt{|\mathcal{M}| - 1}} + S \cdot (3S^2 + 2) \cdot \mu(\mathsf{KG}) \quad .
\end{aligned}
$$