

Generic Authenticated Key Exchange in the Quantum Random Oracle Model (full version)

Kathrin Hövelmanns¹ Eike Kiltz¹ Sven Schäge¹ Dominique Unruh²

January 29, 2020

¹ Ruhr-Universität Bochum

[kathrin.Hoevermanns](mailto:kathrin.Hoevermanns@rub.de), [eike.kiltz](mailto:eike.kiltz@rub.de), [sven.schaege](mailto:sven.schaege@rub.de)@rub.de

² University of Tartu

unruh@ut.ee

Abstract

We propose FO_{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.

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

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

1 Introduction

AUTHENTICATED KEY EXCHANGE. Besides public key encryption (PKE) and digital signatures, authenticated key exchange (AKE) is arguably one of the most important cryptographic building blocks in modern security systems. In the last two decades, research on AKE protocols has made tremendous progress in developing more solid theoretical foundations [11, 20, 39, 32] as well as increasingly efficient designs of AKE protocols [38, 48, 45]. 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, 18]. Even rarer are constructions that only rely on PKE, discarding Diffie-Hellman key exchanges entirely. Notable recent exceptions are [24, 25] and the protocol in [2], the latter of which has been criticised for having a flawed security proof and a weak security model [47, 40].

THE NIST POST-QUANTUM COMPETITION. Recently, some of the above mentioned designs have gathered renewed interest in the quest of finding AKE protocols that are secure against quantum adversaries, i.e., adversaries equipped with a quantum computer. In particular, the National Institute of Standards and Technology (NIST) announced a competition with the goal to standardise new PKE and signature

algorithms [42] with security against quantum adversaries. With the understanding that an AKE protocol can be constructed from low level primitives such as quantum-secure PKE and signature schemes, the NIST did not require the submissions to describe a concrete AKE protocol. Many PKE and signature candidates base their security on the hardness of certain problems over lattices and codes, which are generally believed to resist quantum adversaries.

THE QUANTUM ROM. Quantum computers may execute all “offline primitives” such as hash functions on arbitrary superpositions, which motivated the introduction of the quantum (accessible) random oracle model (QROM) [15]. 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 PUBLIC-KEY PRIMITIVES. The generic AKE protocol of Fujioaka et al. [24] (itself based on [18]) transforms a passively secure PKE scheme PKE and an actively (i.e., IND-CCA) secure PKE scheme PKE_{cca} into an AKE protocol. We will refer to this transformation as $\text{FSXY}[\text{PKE}, \text{PKE}_{\text{cca}}]$. Since the FSXY transformation is in the standard model, it is likely to be secure with the same proof in the post-quantum setting and thus also in the QROM. The standard way to obtain actively secure encryption from passively secure ones is the Fujisaki-Okamoto transformation $\text{PKE}_{\text{cca}} = \text{FO}[\text{PKE}, \text{G}, \text{H}]$ [26, 27]. In its “implicit rejection” variant [29], it comes with a recently discovered security proof [44] that models the hash functions G and H as quantum random oracles. Indeed, the *combined AKE transformation* $\text{FSXY}[\text{PKE}, \text{FO}[\text{PKE}, \text{G}, \text{H}]]$ 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 [44] requires the underlying scheme to be perfectly correct, and a generalisation of the proof for schemes with non-perfect correctness is not straightforward. Hence, it is unclear whether $\text{FSXY}[\text{PKE}, \text{FO}[\text{PKE}, \text{G}, \text{H}]]$ can be instantiated with lattice- or code-based encryption schemes.
- **Lack of simplicity.** The Fujisaki-Okamoto transformation already involves hashing the key using hash function H , and FSXY involves even more (potentially redundant) hashing of the (already

¹Clearly, 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 immunise against decryption errors (e.g., [23]). 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, 30]), 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.

hashed) session key. Overall, the combined transformation seems overly complicated and hence impractical.

In [25], a transformation was given that started from oneway-secure KEMs, but its security proof was given in the ROM, and its generalisation to the QROM was explicitly left as an open problem. Furthermore, it involves more hashing, similar to transformation FSXY.

Hence, it seems desirable to provide a simplified transformation that gets rid of unnecessary hashing steps, and that can be proven secure in the QROM even if the underlying scheme does not satisfy perfect correctness. As a motivating example, note that the Kyber AKE protocol [17] 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, $\text{FO}_{\text{AKE}}[\text{PKE}, \text{G}, \text{H}]$ (“Fujisaki-Okamoto for AKE”) that converts any passively secure encryption scheme into an actively secure AKE protocol, with provable security in the quantum random oracle model. It can deal with non-perfect correctness and does not use digital signatures. Our transformation FO_{AKE} can be viewed as a modification of the transformation given in [25]. 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 [33].

1.1.1 FO transformation for KEMs.

To simplify the presentation of FO_{AKE} , we first give some background on the Fujisaki-Okamoto transformation for KEMs. In its original form [26, 27], FO yields an encryption scheme that is IND-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 [22] provided FO-like IND-CCA secure KEMs. (Recall that any IND-CCA secure Key Encapsulation Mechanism can be combined with any (one-time) chosen-ciphertext secure symmetric encryption scheme to obtain a IND-CCA secure PKE scheme [21].) 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, [29] gave several modularisations of FO-like transformations and proved them robust against correctness errors. The key observation was that FO-like transformations essentially consists of two separate steps and can be dissected into two transformations, as sketched in the introduction of [29]:

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

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

where $\text{G}(m)$ is used as the random coins for Enc , rendering Enc' deterministic. $\text{Dec}'(sk, c)$ first decrypts c into m' and rejects if $\text{Enc}(pk, m'; \text{G}(m')) \neq c$ (“re-encryption”).

- Transformation U_m^\times : “Hashing”. Starting from an encryption scheme PKE' and a hash function H, key encapsulation mechanism $\text{KEM}_m^\times = \text{U}_m^\times[\text{PKE}', \text{H}]$ with “implicit rejection” is defined by

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

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

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

where $m := \text{Dec}(sk, c)$ 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 [43, Sec. 5.3].

Transformation T was proven secure both in the (classical) ROM and the QROM, and U_m^χ was proven secure in the ROM. To achieve QROM security, [29] gave a modification of U_m^χ , called QU_m^χ , but its security proof in the QROM suffered from a quartic³ loss in tightness, and furthermore, most real-world proposals are designed such that they fit the framework of $FO_m^\chi = U_m^\chi \circ T$, not $QU_m^\chi \circ T$.

A slightly different modularisation was introduced in [44]: they gave transformations TPunc ("Puncturing and Encrypt-with-Hash") and SXY ("Hashing with implicit reject and reencryption"). SXY differs from U_m^χ in that it reencrypts during decryption. Hence, it can only be applied to deterministic schemes. Even in the QROM, its CCA security tightly reduces to an intermediate notion called Disjoint Simulatability (DS) of ciphertexts. Intuitively, disjoint simulatability means that we can efficiently sample "fake ciphertexts" that are computationally indistinguishable from real PKE ciphertexts ("simulatability"), while the set of possible fake ciphertexts is required to be (almost) disjoint from the set of real ciphertexts. DS is naturally satisfied by many code/lattice-based encryption schemes. Additionally, it can be achieved using transformation Punc, i.e., by puncturing the underlying schemes' message space at one point and using this message to sample fake encryptions. Deterministic DS can be achieved by using transformation TPunc, albeit non-tightly: the reduction suffers from quadratic loss in security and an additional factor of q , the number of the adversary's hash queries.

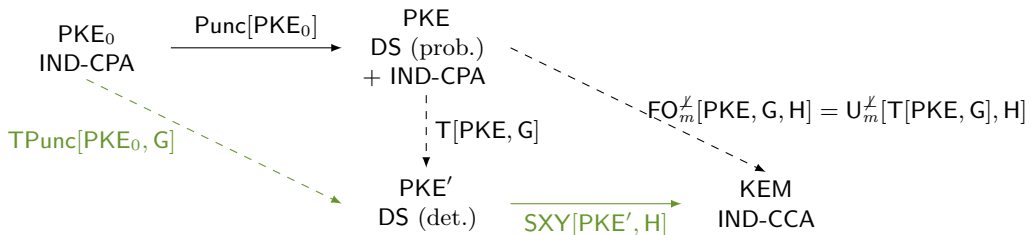


Figure 1: Comparison of [44]'s modular transformation (green) with ours. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions.

However, the reduction that is given in [44] requires the underlying encryption scheme to be perfectly correct. Later, [33] gave non-modular security proofs for the transformations FO_m^χ and FO^χ as well as a security proof for SXY⁴ for schemes with correctness errors, which still suffered from quadratic loss in security and an additional factor of q , the latter of which this work improves to \sqrt{q} .

Our transformation FO_m^χ can be applied to any PKE scheme that is both IND-CPA and DS secure. The reduction is tighter than the one that results from combining those of TPunc and SXY in [44], and also than the reduction given in [34]. 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 [34]) due to a better bound for the generic distinguishing problem. In cases where PKE is not already DS, this requirement can be waived with negligible loss of efficiency: To rely on IND-CPA alone, all that has to be done is to plug in transformation Punc. A visualisation is given in Figure 1.

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 (IND-AA) and the weaker notion of indistinguishability against active attacks without state reveal in the test session (IND-StAA). IND-AA captures the classical notion of key indistinguishability

³not just quadratic, but indeed quartic

⁴Note that nomenclature of [34] is a bit misleading: while the respective KEM is called U_m^χ , it is actually transformation SXY (it reencrypts during decryption, which U_m^χ does not).

(as introduced by Bellare and Rogaway [11]) as well as security against reflection attacks, key compromise impersonation (KCI) attacks, and weak forward secrecy (wFS) [38]. 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 [18], this makes our model incomparable to the eCK model [39] but strictly stronger than the CK model. Essentially, the IND-AA model states that the session key remains indistinguishable from a random one even if

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

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

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

We remark that IND-StAA security is essentially the same notion that was achieved by the FSXY transformation [24].⁵ In Section 5 we provide a more general perspective on how our model compares to existing ones.

1.1.3 Our Authenticated Key-Exchange Protocol.

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

Transformation $\text{FO}_{\text{AKE}}[\text{PKE}, \text{G}, \text{H}]$ is described in Figure 2 and uses transform $\text{PKE}' = \text{T}[\text{PKE}, \text{G}]$ as a building block. (The full construction is given in Figure 19, see Section 6.) Our main security result (Theorem 4) states that $\text{FO}_{\text{AKE}}[\text{PKE}, \text{G}, \text{H}]$ is an IND-StAA-secure AKE if the underlying probabilistic PKE is DS as well as IND-CPA secure and has negligible correctness error, and furthermore G and H are modeled as quantum random oracles.

The proof essentially is the AKE analogue to the security proof of FO_m^\times we give in Section 3.2: By definition of our security model, it always holds that at least one of the messages m_i , m_j and \tilde{m} is hidden from the adversary (unless it loses trivially) since it may not reveal a party’s secret key and its session state at the same time. Adapting the simulation technique in [44], we can simulate the session keys even if we do not know the corresponding secret key sk_i (sk_j , 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 $\text{FO}_{\text{AKE}}[\text{PKE}, \text{G}, \text{H}]$ in the stronger sense of IND-AA and actually, it is not secure against state attacks. Secondly, note that our security statement involves the probabilistic scheme PKE rather than PKE' . Unfortunately, we were not able to provide a modular proof of AKE solely based on reasonable security properties of $\text{PKE}' = \text{T}[\text{PKE}, \text{G}]$. The reason for this is indeed the non-perfect correctness of PKE. This difficulty corresponds to the difficulty to generalise [44]’s result for deterministic encryption schemes with correctness errors discussed above.

CONCRETE APPLICATIONS. Our transformation can be applied to any scheme that is IND-CPA secure with post-quantum security, e.g., Frodo [41], Kyber [17], and Lizard [5]. Recall that the additional requirement of DS can be achieved with negligible loss of efficiency. However, in many applications even this negligible loss is inexistent since most of the aforementioned schemes can already be proven DS under the same assumption that their IND-CPA security is based upon.

⁵The difference is that the model from [24] 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.

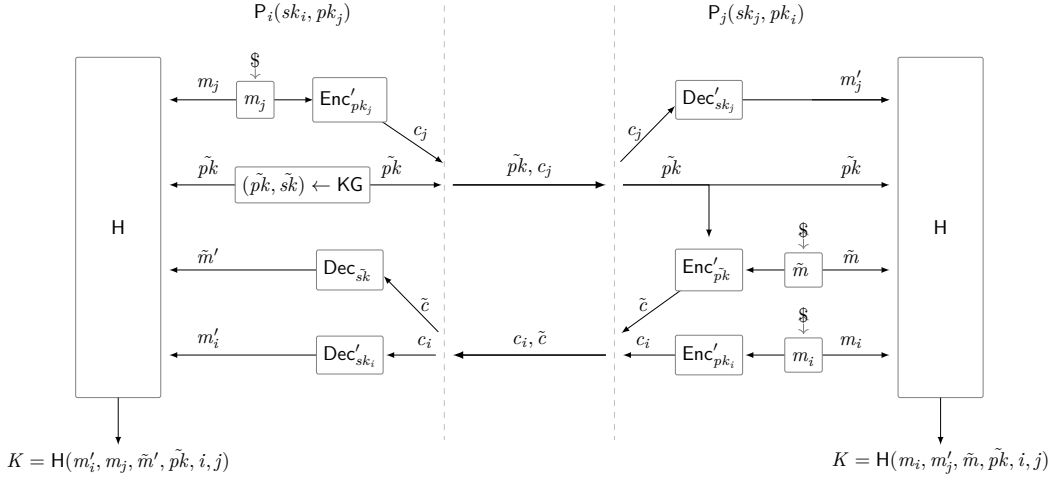


Figure 2: A visualisation of our authenticated key-exchange protocol FO_{AKE} . We make the convention that, in case any of the Dec' algorithms returns \perp , the session key K is derived deterministically and pseudorandomly from the player’s state (“implicit rejection”).

1.1.4 Subsequent work.

Since this paper was published on eprint, there has been more work on CCA security of FO in the QROM ([36, 14]), essentially achieving the same level of tightness as this work. [14] achieves more modularity, and covers a class of schemes that is both less and more restrictive at the same time: They only require schemes to be oneway-secure (instead of CPA, as required in this work), but the schemes have to meet an additional injectivity requirement (specified below).

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

Essentially, [36] is an update of [33] that makes use of the improved Oneway-to-Hiding bounds given in [3], thereby improving [33]’s bound $q\sqrt{\epsilon}$ to $\sqrt{q}\sqrt{\epsilon}$, with the security requirement switching from onewayness to IND-CPA. The result seems to differ from this work solely in its (nonmodular) proof structure.

In [14], a new modular proof for FO was given by starting from probabilistic onewayness and choosing deterministic oneway-secure as their intermediate⁶ notion, opposed to our (strictly stronger) intermediate notion of deterministic DS. This approach matches the observation that if one can start from a scheme that already is deterministically oneway-secure (like [13]), derandomisation step T is superfluous. In this case, only transformation U has to be applied, which is proven secure q -tightly. The weaker intermediate notion, however, shifts the root-nontightness to second transformation U. Therefore, the result still is heavily non-tight, even if derandomising via T is skipped. Furthermore, no tightness improvements whatsoever are achieved if the underlying scheme is not already deterministic, and thus has to be derandomised using T first.

MODULARITY. The modular proof of [14] is achieved by introducing an additional notion for the intermediate scheme that deals with correctness errors. Unfortunately, the possibility of correctness errors complicate modular attempts on analysing FO: For underlying probabilistic schemes, [14] requires more than this work since its approach only is applicable if the “intermediate” scheme is injective with overwhelming probability. It is very likely that the modular approach of [14] could be generalised to an

⁶ By “intermediate”, we mean the deterministic scheme that is to be plugged into one of the U-transforms. In most cases, it is derived by starting from a probabilistic scheme and first applying derandomisation transformation T.

AKE proof that similarly is modular and hence, conceptually nicer. But this gain in modularity would come at a cost: The approach only is applicable if the derandomised scheme is essentially injective. We would, therefore, add an unnecessary restriction on the class of schemes that AKE can be based upon.

1.1.5 Open Problems.

In the literature, one can find several Diffie-Hellman based protocols that achieve IND-AA security, for example HMQV [38]. However, none of them provides security against quantum computers. We leave as an interesting open problem to design a generic and efficient two-message AKE protocol in our stronger IND-AA model, preferably with a security proof in the QROM to guarantee its security even in the presence of quantum adversaries.

While [14] gave a proof of CCA security that is conceptually cleaner, it still is heavily non-tight due to its root-nontightness, with the root-nontightness stemming from its usage of a standard Oneway-to-Hiding strategy. Recent work [35] proved that for reductions using this standard approach, suffering from quadratic security loss is inevitable. We would like to point out, however, that we do not view this result as an impossibility result⁷. It rather proves impossibility of root-tightness *for a certain type of reduction*, and thereby informs us how to adapt possible future proof strategies: A root-tight proof of CCA security still might be achievable, but the respective reduction would have to be more sophisticated than extracting oneway solutions for the underlying scheme by simply applying Oneway-to-Hiding.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$. 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 $x \leftarrow_{\mathfrak{s}} S$, while we denote the sampling according to some distribution \mathfrak{D} by $x \leftarrow \mathfrak{D}$. By $\llbracket B \rrbracket$ 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 [46, 12], we use code-based games. We implicitly assume boolean flags to be initialised to false, numerical types to 0, sets to \emptyset , 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 $\text{PKE} = (\text{KG}, \text{Enc}, \text{Dec})$ consists of three algorithms, and a finite message space \mathcal{M} which we assume to be efficiently recognisable. The key generation algorithm KG outputs a key pair (pk, sk) , where pk also defines a finite randomness space $\mathcal{R} = \mathcal{R}(pk)$ as well as a ciphertext space \mathcal{C} . The encryption algorithm Enc , on input pk and a message $m \in \mathcal{M}$, outputs an encryption $c \leftarrow \text{Enc}(pk, m)$ of m under the public key pk . If necessary, we make the used randomness of encryption explicit by writing $c := \text{Enc}(pk, m; r)$, where $r \leftarrow_{\mathfrak{s}} \mathcal{R}$. The decryption algorithm Dec , on input sk and a ciphertext c , outputs either a message $m = \text{Dec}(sk, c) \in \mathcal{M}$ 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

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

Definition 2.2 (Collision probability of ciphertexts.). We define

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

⁷ A strict impossibility result would have to consist of a concrete scheme as well as a concrete attack, with the latter matching the given upper bound.

Definition 2.3 (γ -Spreadness). [26] We say that PKE is γ -spread iff for all key pairs $(pk, sk) \in \text{supp}(\text{KG})$ and all messages $m \in \mathcal{M}$ it holds that

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

Definition 2.4 (Correctness). [29] We define $\delta := \mathbf{E}[\max_{m \in \mathcal{M}} \Pr[c \leftarrow \text{Enc}(pk, m) : \text{Dec}(sk, c) \neq m]]$, where the expectation is taken over $(pk, sk) \leftarrow \text{KG}$.

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

Definition 2.5 (IND-CPA). Let $\text{PKE} = (\text{KG}, \text{Enc}, \text{Dec})$ be a public-key encryption scheme. We define game IND-CPA as in Figure 3, and the IND-CPA advantage function of a quantum adversary $A = (A_1, A_2)$ against PKE (such that A_2 has binary output) as

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

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

<u>GAME IND-CPA_b</u>	<u>GAME IND-CCA</u>	<u>DECAPS($c \neq c^*$)</u>
01 $(pk, sk) \leftarrow \text{KG}$	06 $(pk, sk) \leftarrow \text{KG}$	12 $K := \text{Decaps}(sk, c)$
02 $(m_0^*, m_1^*, st) \leftarrow A_1(pk)$	07 $b \leftarrow_{\S} \mathbb{F}_2$	13 return K
03 $c^* \leftarrow \text{Enc}(pk, m_b^*)$	08 $(K_0^*, c^*) \leftarrow \text{Encaps}(pk)$	
04 $b' \leftarrow A_2(pk, c^*, st)$	09 $K_1^* \leftarrow_{\S} \mathcal{K}$	
05 return b'	10 $b' \leftarrow A^{\text{DECAPS}}(pk, c^*, K_b^*)$	
	11 return $\llbracket b' = b \rrbracket$	

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

DISJOINT SIMULATABILITY. Following [44], 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) Let $\text{PKE} = (\text{KG}, \text{Enc}, \text{Dec})$ be a PKE scheme with message space \mathcal{M} and ciphertext space \mathcal{C} , coming with an additional PPT algorithm $\overline{\text{Enc}}$. For quantum adversaries A , we define the *advantage against PKE's disjoint simulatability* as

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

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

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

2.2 Key Encapsulation

SYNTAX. A key encapsulation mechanism $\text{KEM} = (\text{KG}, \text{Encaps}, \text{Decaps})$ 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 $K := \text{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.

We call KEM δ -correct if

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

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: Indistinguishability 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

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

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 $\text{MEASURE}(\cdot)$, where the outcome of $\text{MEASURE}(|\varphi\rangle)$ 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 [15, 6], we view a quantum oracle $|O\rangle$ as a mapping

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

where $O : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, and model quantum adversaries A with access to O by a sequence $U_1, |O\rangle, U_2, \dots, |O\rangle, U_N$ of unitary transformations. We write $\text{A}^{(O)}$ 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 [19], and unlike in [16]. In the IND-CCA game, the adversary only has access to a classical decryption oracle, unlike in [28] and [1].

Zhandry [49] proved that no quantum algorithm $\text{A}^{(O)}$, issuing at most q quantum queries to $|O\rangle$, can distinguish between a random function $O : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ and a $2q$ -wise independent function f_{2q} . For concreteness, we view $f_{2q} : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ as a random polynomial of degree $2q$ over the finite field \mathbb{F}_{2^n} . The running time to evaluate f_{2q} is linear in q . In this article, we will use this observation in the context of security reductions, where quantum adversary B simulates quantum adversary $\text{A}^{(O)}$ issuing at most q queries to $|O\rangle$. Hence, the running time of B is $\text{Time}(\text{B}) = \text{Time}(\text{A}) + q \cdot \text{Time}(O)$, where $\text{Time}(O)$ denotes the time it takes to simulate $|O\rangle$. Using the observation above, B can use a $2q$ -wise independent function in order to (information-theoretically) simulate $|O\rangle$, and we obtain that the running time of B is $\text{Time}(\text{B}) = \text{Time}(\text{A}) + q \cdot \text{Time}(f_{2q})$, and the time $\text{Time}(f_{2q})$ to evaluate f_{2q} is linear in q . Following [44] and [37], 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 $|O\rangle$. Assuming evaluating the random oracle takes one time unit, we write $\text{Time}(\text{B}) = \text{Time}(\text{A}) + q$, which is approximately $\text{Time}(\text{A})$.

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 O_S^{SC} : Algorithm O_S^{SC} , when queried on $|\psi, 0\rangle$, measures with respect to the projectors M_1 and M_0 , where $M_1 := \sum_{x \in S} |x\rangle\langle x|$ and $M_0 := \sum_{x \notin S} |x\rangle\langle x|$. The oracle then initialises the second register to $|b\rangle$ for the measured bit b . This means that $|\psi, 0\rangle$ collapses to either a state $|\psi', 0\rangle$ such that $|\psi'\rangle$ only contains elements of $X \setminus S$ or to a state $|\psi', 1\rangle$ such that $|\psi'\rangle$ only contains elements of S . Let FIND denote the event that the latter ever is the case, i.e., that O_S^{SC} ever answers with $|\psi', 1\rangle$ for some ψ' . To a quantum-accessible oracle G and a subset $S \subset X$, Ambainis et al. associate the following punctured oracle $G \setminus S$ that removes S from the domain of G unless FIND occurs.

$G \setminus S \psi, \phi\rangle$
01 $ \psi', b\rangle := O_S^{\text{SC}} \psi, 0\rangle$
02 return $U_G \psi', \phi\rangle$

Figure 4: Punctured oracle $G \setminus S$ for O2H.

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

Theorem 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,*

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

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

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

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], [31, Lem. 3]) is to distinguish quantum access to an oracle $F : X \rightarrow \mathbb{F}_2$, 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 [31, Lem. 3].

Lemma 2.9 (Generic Distinguishing Problem with Bounded Probabilities). *[Generic Distinguishing Problem with Bounded Probabilities] Let X be a finite set, and let $\lambda \in [0, 1]$. Then, for any (unbounded, quantum) algorithm A issuing at most q quantum queries,*

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

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

GAME $\text{GDPB}_{\lambda,b}$ 01 $(\lambda_x)_{x \in X} \leftarrow \mathbf{A}_1$ 02 if $\exists x \in X$ s.t. $\lambda_x > \lambda$ return 0 03 if $b = 0$ 04 $F := 0$ 05 else for all $x \in X$ 06 $F(x) \leftarrow B_{\lambda_x}$ 07 $b' \leftarrow \mathbf{A}_2^{(F)}$ 08 return b'

Proof. In this proof, let CGDPB_λ denote the game GDPB_λ as defined in [4] and [31], 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 \mathbf{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 \mathbf{B} against game CGDPB_λ , given in Figure 5.

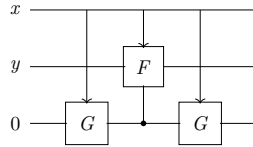
B₁ 01 $(\lambda_x)_{x \in X} \leftarrow \mathbf{A}_1$ 02 $\lambda := \max_{x \in X} \lambda_x$ 03 for all $x \in X$ 04 $\mu_x := \frac{\lambda_x}{\lambda}$ 05 $G(x) \leftarrow B_{\mu_x}$ 06 return λ	B₂^(F) 07 $b' \leftarrow \mathbf{A}_2^{(F \cdot G)}$ 08 return b'
--	--

Figure 5: Adversary \mathbf{B} for the proof of Lemma 1.

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

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

We now argue that \mathbf{B} can realise \mathbf{A} 's oracle access to $|F \cdot G\rangle$ in a way such that any query to $|F \cdot G\rangle$ by \mathbf{A} triggers at most one query to $|F\rangle$. 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 \mathbf{B} can answer queries to $|F \cdot G\rangle$ by querying $|F\rangle$ just once. Since \mathbf{B} issues at most q queries to $|F\rangle$, we can apply [31, Lem. 3] and obtain

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

□

3 The FO Transformation: QROM security with correctness errors

In Section 3.1, we modularise transformation TPunc that was given in [44] and that turns any public key encryption scheme that is IND-CPA secure into a deterministic one that is DS. Transformation TPunc essentially consists of first puncturing the message space at one point (transformation Punc, to achieve probabilistic DS), and then applying transformation T. Next, in Section 3.2, we show that transformation $U_m^{\mathcal{L}}$, when applied to T, transforms any encryption scheme that is DS as well as IND-CPA into a KEM that is IND-CCA secure. We believe that many lattice-based schemes fulfill DS in a natural way,⁸ but for the sake of completeness, we will show in Section 3.3 how transformation Punc can be used to waive the requirement of DS with negligible loss of efficiency.

3.1 Modularisation of TPunc

We modularise transformation TPunc ("Puncturing and Encrypt-with-Hash") that was given in [44], and that turns any IND-CPA secure PKE scheme into a deterministic one that is DS. Note that apart from reencryption, TPunc[PKE₀, G] given in [44] and our modularisation T[Punc[PKE₀], G] are equal. We first give transformation Punc that turns any IND-CPA secure scheme into a scheme that is both DS and IND-CPA. In Section 3.1, we show that transformation T turns any scheme that is DS as well as IND-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 IND-CPA secure public-key encryption scheme into a DS secure one by puncturing the message space at one message and sampling encryptions of this message as fake encryptions.

THE CONSTRUCTION. To a public-key encryption scheme PKE₀ = (KG₀, Enc₀, Dec₀) with message space \mathcal{M}_0 , we associate PKE := Punc[PKE₀, \hat{m}] := (KG := KG₀, Enc, Dec := Dec₀) 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 6. Note that transformation Punc will only be used as a helper transformation to achieve DS, generically.

Enc(pk, m ∈ \mathcal{M})	$\overline{\text{Enc}}(pk)$
01 $c \leftarrow \text{Enc}_0(pk, m)$	03 $c \leftarrow \text{Enc}_0(pk, \hat{m})$
02 return c	04 return c

Figure 6: Encryption and fake encryption sampling of PKE = Punc[PKE₀].

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 IND-CPA secure. Our security proof is tighter than the proof given for TPunc (see [44, Theorem 3.3]) due to our use of the semi-classical O2H theorem.

THE CONSTRUCTION. Take an encryption scheme PKE = (KG, Enc, Dec) with message space \mathcal{M} and randomness space \mathcal{R} . Assume PKE to be additionally endowed with a sampling algorithm $\overline{\text{Enc}}$ (see Definition 6). To PKE and random oracle $G : \mathcal{M} \rightarrow \mathcal{R}$, we associate PKE' = T[PKE, G], where the algorithms of PKE' = (KG' := KG, Enc', Dec', $\overline{\text{Enc}}' := \overline{\text{Enc}}$) are defined in Figure 7. Note that Enc' deterministically computes the ciphertext as $c := \text{Enc}(pk, m; G(m))$.

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

⁸Fake 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.

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

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

Lemma 3.1 (DS security of PKE'). *If PKE is ϵ -disjoint, so is PKE' . For all adversaries \mathbf{A} issuing at most $q_{\mathbf{G}}$ (quantum) queries to \mathbf{G} , there exist an adversary \mathbf{B}_{IND} and an adversary \mathbf{B}_{DS} such that*

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

and the running time of each adversary is about that of \mathbf{B} .

Proof. It is straightforward to prove disjointness since $\text{Enc}'(pk, \mathcal{M})$ is subset of $\text{Enc}(pk, \mathcal{M}; \mathcal{R})$. Let \mathbf{A} be a DS adversary against PKE' . Consider the sequence of games given in Figure 8. Per definition,

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

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

Figure 8: Games $G_0 - G_5$ for the proof of Lemma 2.

To upper bound $|\Pr[G_0^{\mathbf{A}} \Rightarrow 1] - \Pr[G_3^{\mathbf{A}} \Rightarrow 1]|$, consider adversary \mathbf{B}_{DS} against the disjoint simulatability of the underlying scheme PKE , given in Figure 9. \mathbf{B}_{DS} runs in the time that is required to run \mathbf{A} and to simulate \mathbf{G} for $q_{\mathbf{G}}$ queries. Since \mathbf{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 $c \leftarrow \mathbf{Enc}(pk, m^*)$,

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

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

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

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

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

$\frac{\mathbf{B}_{\text{DS}}(pk, c)}{01 \ b' \leftarrow \mathbf{A}^{ \mathcal{G} }(pk, c)} \\ 02 \ \mathbf{return} \ b'$	$\frac{\mathbf{B}_{\text{IND},1}(pk)}{03 \ m^* \leftarrow_{\$} \mathcal{M}} \\ 04 \ \mathbf{return} \ (0, m^*, \text{st} := m^*)$	$\frac{\mathcal{G} \setminus \{m^*\} \psi, \phi}{08 \ \psi', b\rangle := \mathbf{O}_{\{m^*\}}^{\text{SC}} \psi, 0\rangle} \\ 09 \ \mathbf{if} \ b = 1 \\ 10 \ \quad \mathbf{FIND} := \mathbf{true} \\ 11 \ \mathbf{return} \ U_{\mathcal{G}} \psi', \phi\rangle$
$\frac{\mathbf{B}_{\text{IND},2}(pk, c^*, \text{st} := m^*)}{05 \ \mathbf{FIND} := \mathbf{false}} \\ 06 \ b' \leftarrow \mathbf{A}^{ \mathcal{G} \setminus \{m^*\} }(pk, c^*) \\ 07 \ \mathbf{return} \ \mathbf{FIND}$		

Figure 9: Adversaries \mathbf{B}_{DS} and \mathbf{B}_{IND} for the proof of Lemma 2.

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

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

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

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

To upper bound $|\Pr[G_4^{\mathbf{A}} \Rightarrow 1] - \Pr[G_5^{\mathbf{A}} \Rightarrow 1]|$, consider adversary \mathbf{B}_{IND} against the IND-CPA security of PKE, also given in Figure 9. \mathbf{B}_{IND} runs in the time that is required to run \mathbf{A} and to simulate the measured version of oracle \mathcal{G} for $q_{\mathcal{G}}$ queries. \mathbf{B}_{IND} perfectly simulates game G_4 if run in game IND-CPA_0 and game G_5 if run in game IND-CPA_1 , therefore,

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

Collecting the probabilities yields

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

□

3.2 Transformation FO_m^{\neq} and correctness errors

Transformation SXY [44] got rid of the additional hash (sometimes called key confirmation) that was included in [29]'s quantum transformation QU_m^{\neq} . SXY is essentially the (classical) transformation U_m^{\neq} that was also given in [29], and apart from doing without the additional hash, it comes with a tight security reduction in the QROM. SXY differs from the (classical) transformation U_m^{\neq} only in the regard that it reencrypts during decapsulation. (In [29], reencryption is done during decryption of T .)

The security proof given in [44] 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 $\text{FO}_m^\perp = \text{U}_m^\perp[\text{T}[-, \text{G}], \text{H}]$ 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 [44] due to our tighter security proof for T .

THE CONSTRUCTION. To $\text{PKE} = (\text{KG}, \text{Enc}, \text{Dec})$ with message space \mathcal{M} and randomness space \mathcal{R} , and random oracles $\text{H} : \mathcal{M} \rightarrow \mathcal{K}$, $\text{G} : \mathcal{M} \rightarrow \mathcal{R}$, and an additional internal random oracle $\text{H}_r : \mathcal{C} \rightarrow \mathcal{K}$ that can not be directly accessed, we associate $\text{KEM} = \text{FO}_m^\perp[\text{PKE}, \text{G}, \text{H}] := \text{U}_m^\perp[\text{T}[\text{PKE}, \text{G}], \text{H}]$, where the algorithms of $\text{KEM} = (\text{KG}, \text{Encaps}, \text{Decaps})$ are given in Figure 10.

<u>Encaps(pk)</u>	<u>Decaps(sk, c)</u>
01 $m \leftarrow_{\$} \mathcal{M}$	05 $m' := \text{Dec}(sk, c)$
02 $c := \text{Enc}(pk, m; \text{G}(m))$	06 if $m' = \perp$ or $\text{Enc}(pk, m'; \text{G}(m')) \neq c$
03 $K := \text{H}(m)$	07 return $K := \text{H}_r(c)$
04 return (K, c)	08 else return $K := \text{H}(m')$

Figure 10: Key encapsulation mechanism $\text{KEM} = \text{FO}_m^\perp[\text{PKE}, \text{G}, \text{H}] = \text{U}_m^\perp[\text{T}[\text{PKE}, \text{G}], \text{H}]$. Oracle H_r is used to generate random values whenever reencryption fails. This strategy is called implicit reject. Amongst others, it is used in [29], [44], and [33]. 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 OF KEM. The following theorem (whose proof is essentially the same as in [44] 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 $\xrightarrow{\text{QROM}}$ KEM IND-CCA). *Assume PKE to be δ -correct, and to come with a fake sampling algorithm $\bar{\text{Enc}}$ 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 H , and at most q_G quantum queries to G , there exist (quantum) adversaries B_{DS} and B_{IND} such that*

$$\begin{aligned} \text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(\text{A}) &\leq 8 \cdot (2 \cdot q_G + q_H + q_D + 4)^2 \cdot \delta + \text{Adv}_{\text{PKE}}^{\text{DS}}(\text{B}_{\text{DS}}) \\ &\quad + 2 \cdot \sqrt{(q_G + q_H) \cdot \text{Adv}_{\text{PKE}}^{\text{IND-CPA}}(\text{B}_{\text{IND}}) + \frac{4(q_G + q_H)^2}{|\mathcal{M}|}} + \epsilon_{dis} \end{aligned}$$

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 H , and at most q_G queries to the quantum random oracle G . Consider the sequence of games given in Figure 11.

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

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

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

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

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

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

GAMES $G_0 - G_6$	$\text{DECAPS}(c \neq c^*)$	$\parallel G_0 - G_2$
01 $(pk, sk) \leftarrow \text{KG}$	19 $m' := \text{Dec}(sk, c)$	
02 $H_r \leftarrow_{\S} \mathcal{K}^C$	20 if $m' = \perp$	
03 $G \leftarrow_{\S} \mathcal{R}^{\mathcal{M}}$ $\parallel G_0, G_4 - G_6$	or $\text{Enc}(pk, m'; G(m')) \neq c$	
04 Pick $2q$ -wise hash f $\parallel G_1 - G_3$	21 return $K := H_r(c)$	
05 $G := G_{pk, sk}$ $\parallel G_1 - G_3$	22 else	
06 $H \leftarrow_{\S} \mathcal{K}^{\mathcal{M}}$ $\parallel G_0 - G_1$	23 return $K := H(m')$ $\parallel G_0 - G_1$	
07 $H_q \leftarrow_{\S} \mathcal{K}^C$ $\parallel G_2 - G_6$	24 return $K := H_q(c)$ $\parallel G_2 - G_6$	
08 $H := H_q(\text{Enc}(pk, -; G(-)))$ $\parallel G_2 - G_6$		
09 $b \leftarrow_{\S} \mathbb{F}_2$	$\text{DECAPS}(c \neq c^*)$ $\parallel G_3 - G_6$	
10 $m^* \leftarrow \mathcal{M}$	25 return $K := H_q(c)$	
11 $c^* := \text{Enc}(pk, m^*; G(m^*))$ $\parallel G_0 - G_4$		
12 $c^* \leftarrow \overline{\text{Enc}}(pk)$ $\parallel G_5 - G_6$	$G_{pk, sk}(m)$	
13 $K_0^* := H(m^*)$ $\parallel G_0 - G_1$	26 $r := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$	
14 $K_0^* := H_q(c^*)$ $\parallel G_2 - G_5$	27 return r	
15 $K_0^* \leftarrow_{\S} \mathcal{K}$ $\parallel G_6$		
16 $K_1^* \leftarrow_{\S} \mathcal{K}$		
17 $b' \leftarrow A^{\text{DECAPS}, H , G }(pk, c^*, K_b^*)$		
18 return $\llbracket b' = b \rrbracket$		

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

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

To upper bound $|\Pr[G_0^A = 1] - \Pr[G_1^A = 1]|$, we construct an (unbounded, quantum) adversary B against the generic distinguishing problem with bounded probabilities GDPB (see Lemma 1) in Figure 12, issuing $q_G + q_D + 1$ queries to F . B draws a key pair $(pk, sk) \leftarrow \text{KG}$ 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 $\text{GDPB}_{\lambda, 1}$, the random variable $F(m)$ is bernoulli-distributed according to $B_{\lambda(m)} = B_{\delta(pk, sk, m)}$. By construction, the random variable $G(m)$ defined in line 19 if $F(m) = 0$ and in line 21 if $F(m) = 1$ is uniformly distributed in \mathcal{R} . Therefore, G is a (quantum-accessible) random oracle, and $B^{|F|}$ perfectly simulates game G_0 if executed in game $\text{GDPB}_{\lambda, 1}$. Since $B^{|F|}$ also perfectly simulates game G_1 if executed in game $\text{GDPB}_{\lambda, 0}$,

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

and according to Lemma 1,

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

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 $H \leftarrow_{\S} \mathcal{K}^{\mathcal{M}}$, we draw $H_q \leftarrow_{\S} \mathcal{K}^C$ in line 07 and define $H := H_q(\text{Enc}(pk, -; G(-)))$ in line 08. For consistency, we also change key K_0^* in line 14 from letting $K_0^* := H(m^*)$ to letting $K_0^* := H_q(c^*)$, which is a purely conceptual change since $c^* = \text{Enc}(pk, m^*; G(m^*))$. Additionally, we make the change of H explicit in oracle DECAPS , i.e., we change oracle DECAPS in line 24 such that it returns $K := H_q(c)$ whenever $\text{Enc}(pk, m'; G(m')) = c$. 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 $c = \text{Enc}(pk, m'; G(m'))$, we maintain consistency of H and DECAPS . In conclusion, A 's view is identical in both games and

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

GAME G_3 . In game G_3 , we change oracle DECAPS such that it always returns $K := H_q(c)$, as opposed to returning $K := H_r(c)$ as in game G_2 whenever decryption or reencryption fails (see line 21). We argue

with deterministic encryption $c^* := \text{Enc}(pk, m^*; \mathbf{G}(m^*))$ of a random message m^* , and game G_5 if run with a fake ciphertext,

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

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

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

$\text{C}_{\text{DS}}^{ \mathbf{G} , \mathbf{H}_r , \mathbf{H}_q }(pk, c^*)$	$\text{DECAPS}(c \neq c^*)$
01 $b \leftarrow_{\S} \mathbb{F}_2$	06 return $K := \mathbf{H}_q(c)$
02 $K_0^* := \mathbf{H}_q(c^*)$	
03 $K_1^* \leftarrow_{\S} \mathcal{K}$	
04 $b' \leftarrow \mathbf{A}^{\text{DECAPS}, \mathbf{H} , \mathbf{G} }(pk, c^*, K_b^*)$	
05 return $\llbracket b' = b \rrbracket$	

Figure 13: Adversary C_{DS} (with access to additional oracles \mathbf{H}_r and \mathbf{H}_q) against the disjoint simulatability of $\mathbb{T}[\text{PKE}, \mathbf{G}]$ for the proof of Theorem 2.

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 \mathbf{A} in game G_6 ,

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

It remains to upper bound $|\Pr[G_5^A = 1] - \Pr[G_6^A = 1]|$. To this end, it is sufficient to upper bound the probability that any of the queries to \mathbf{H}_q could possibly contain c^* . Each query to \mathbf{H}_q is either a classical query, triggered by \mathbf{A} querying DECAPS on some ciphertext c , or a query in superposition, triggered by \mathbf{A} querying \mathbf{H} . Since queries to DECAPS on c^* are explicitly forbidden, the only possibility would be one of \mathbf{A} 's queries to \mathbf{H} . \mathbf{A} 's queries to \mathbf{H} trigger queries to \mathbf{H}_q that are of the form $\sum_m \alpha_m |\text{Enc}(pk, m; \mathbf{G}(m))\rangle$. They cannot contain c^* unless there exists some message m such that $\text{Enc}(pk, m; \mathbf{G}(m)) = c^*$. Since we assume PKE to be ϵ_{dis} -disjoint,

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

3.3 CCA security without disjoint simulatability.

In this section, we show that the requirement of disjoint simulatability can be waived with negligible loss of efficiency. Recall that transformation Punc punctures the message space at one message and samples encryptions of this message as fake encryptions, see Figure 6. We will now argue that Punc both achieves (probabilistic) DS and maintains IND-CPA security. Subsequent application of transformation \mathbb{T} (resulting in transformation TPunc) therefore achieves deterministic DS. Hence, TPunc can be plugged into transformation \mathbf{U}_m^{\neq} to achieve CCA security.

The following lemma states that IND-CPA security of PKE_0 implies DS security of PKE. Note that we do not specify 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 will bypass this issue during our proof of CCA security (Theorem 3), in which we will achieve disjointness by switching \mathbf{G} to a function that renders the encryption scheme perfectly correct and hence, injective.

Lemma 3.3 (DS security of Punc). *If PKE_0 is δ -correct, so is PKE. For all adversaries \mathbf{A} , there exists an IND-CPA adversary \mathbf{B} such that*

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

Proof. Let A be a DS adversary against PKE. Consider the games given in Figure 14.

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

Game G	$B_1(pk)$
01 $pk \leftarrow \text{KG}_0$	08 $m \leftarrow_{\S} \mathcal{M}_0 \setminus \{\hat{m}\}$
02 $m \leftarrow_{\S} \mathcal{M}_0 \setminus \{\hat{m}\}$	09 return (m, \hat{m})
03 $b \leftarrow_{\S} \mathbb{F}_2$	
04 $c_0 \leftarrow \text{Enc}_0(pk, m)$	$B_2(c)$
05 $c_1 \leftarrow \text{Enc}_0(pk, \hat{m})$	10 $b' \leftarrow A(pk, c)$
06 $b' \leftarrow A(pk, c_b)$	11 return b'
07 return $\llbracket b' = b \rrbracket$	

Figure 14: Game G and IND-CPA adversary $B = (B_1, B_2)$ for the proof of Lemma 3.

Consider the IND-CPA adversary $B := (B_1, B_2)$ also given in Figure 14. Since B perfectly simulates game G ,

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

The following lemma states that IND-CPA security of PKE_0 translates to IND-CPA security of PKE. Its proof is straightforward.

Lemma 3.4 (IND-CPA security of Punc). *For all IND-CPA adversaries A there exists an adversary B such that*

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

Applying Lemma 2, we obtain the following corollary. It states that combining T with Punc turns IND-CPA security into (deterministic) DS security.

Corollary 3.5 (DS security of TPunc). *For all adversaries A issuing at most q_G queries to $|G\rangle$, there exist two adversaries B_1 and B_2 such that*

$$\text{Adv}_{T[\text{Punc}[\text{PKE}_0, \hat{m}], G]}^{\text{DS}}(A) \leq \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_1) + 2 \cdot \sqrt{q_G \cdot \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_2) + \frac{4q_G^2}{|\mathcal{M}| - 1}} ,$$

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

The following theorem establishes that plugging in transformation Punc (before using FO_m^{\neq}) achieves IND-CCA security from IND-CPA security alone, as long as PKE is γ -spread (see Definition 3).

Theorem 3.6 (CCA security of $\text{FO}_m^{\neq} \circ \text{Punc}$). *Assume PKE_0 to be δ -correct and γ -spread, and let $\hat{m} \in \mathcal{M}$. Let $\text{KEM} := \text{FO}_m^{\neq}[\text{Punc}[\text{PKE}, \hat{m}], G, H]$. 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 H , and at most q_G quantum queries to G , there exist (quantum) CPA adversaries B_1 and B_2 against PKE_0 such that*

$$\begin{aligned} \text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(A) &\leq (8 \cdot (3 \cdot q_G + 2 \cdot q_H + q_D + 6)^2 + 1) \cdot \delta + \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_1) \\ &\quad + 2 \cdot \sqrt{(q_G + q_H) \cdot \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_2) + \frac{4(q_G + q_H)^2}{|\mathcal{M}| - 1}} + 2^{-\gamma} , \end{aligned}$$

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 random oracle H , and at most q_G queries to random oracle G . It is easy to verify that we can apply the first 5 game-hops of our proof of Theorem 2 to $\text{FO}_m^{\neq}[\text{Punc}[\text{PKE}, \hat{m}], G, H]$: We first

enforce that no decryption failure will occur by replacing G with oracle $G_{pk,sk}$ that only samples from good randomness. Note that since PKE_0 is δ -correct, so is $\text{Punc}[\text{PKE}, \hat{m}]$. Afterwards, we plug encryption into the random oracle, and then change oracle DECAPS such that it always returns $K := H_q(c)$, as opposed to implicitly rejecting whenever decryption or reencryption fails. Both changes are not recognisable by A . After evening out the decapsulation oracle, we switch back to using random oracle G and obtain the upper bound

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

Since for $\text{PKE} = \text{Punc}[\text{PKE}, \hat{m}]$, 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 $T[\text{Punc}[\text{PKE}_0, \hat{m}], G]$ such that

$$|\Pr[G_4^A = 1] - \Pr[G_5^A = 1]| = \text{Adv}_{T[\text{Punc}[\text{PKE}_0, \hat{m}], G]}^{\text{DS}}(C_{DS}),$$

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

$$\begin{aligned} & \text{Adv}_{T[\text{Punc}[\text{PKE}_0, \hat{m}], G]}^{\text{DS}}(C_{DS}) \\ & \leq \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_1) + 2 \cdot \sqrt{(q_G + q_H) \cdot \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_2) + \frac{4(q_G + q_H)^2}{|\mathcal{M}| - 1}} . \end{aligned}$$

GAMES $G_5 - G_8$	$\text{DECAPS}(c \neq c^*)$
01 $(pk, sk) \leftarrow \text{KG}$	15 return $K := H_q(c)$
02 Pick $2q$ -wise hash $f \quad // G_7 - G_8$	
03 $G := G_{pk,sk} \quad // G_7 - G_8$	$G_{pk,sk}(m)$
04 $H := H_q(\text{Enc}_0(pk, -, G(-)))$	16 $r := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
05 $b \leftarrow_{\S} \mathbb{F}_2$	17 return r
06 $m^* \leftarrow \mathcal{M} \setminus \{\hat{m}\}$	
07 $c^* \leftarrow \text{Enc}_0(pk, \hat{m})$	
08 if $\text{Enc}_0(pk, \hat{m}; G(\hat{m})) = c^*$	
09 ABORT $// G_6 - G_8$	
10 $K_0^* := H_q(c^*)$ $// G_5 - G_7$	
11 $K_0^* \leftarrow_{\S} \mathcal{K}$ $// G_8$	
12 $K_1^* \leftarrow_{\S} \mathcal{K}$	
13 $b' \leftarrow A^{\text{DECAPS}, H , G }(pk, c^*, K_b^*)$	
14 return $[[b' = b]]$	

Figure 15: Games $G_5 - G_8$ for the proof of Theorem 3.

To justify that we can replace the real key with random, we give a sequence of games in Figure 15.

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

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

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

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

where C is given in Figure 16, and according to Lemma 1,

$$|\Pr[\text{GDPB}_{\lambda,1}^C = 1] - \Pr[\text{GDPB}_{\lambda,0}^C = 1]| \leq 8 \cdot (q_G + q_H + 2)^2 \cdot \delta .$$

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 ,

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

$\underline{C_1}$ 01 $(pk, sk) \leftarrow \text{KG}$ 02 for $m \in \mathcal{M}$ 03 $\lambda(m) := \delta(pk, sk, m)$ 04 return $(\lambda(m))_{m \in \mathcal{M}}$ $\underline{C_2}^{ \mathbb{H}_q , \mathbb{F} }$ 05 $(pk, sk) \leftarrow \text{KG}$ 06 Pick $2q$ -wise hash f 07 $\text{H} := \text{H}_q(\text{Enc}_0(pk, -; \text{G}(-)))$ 08 $b \leftarrow_{\S} \mathbb{F}_2$ 09 $m^* \leftarrow \mathcal{M} \setminus \{\hat{m}\}$ 10 $c^* \leftarrow \text{Enc}_0(pk, \hat{m})$ 11 if $\text{Enc}_0(pk, \hat{m}; \text{G}(\hat{m})) = c^*$ 12 ABORT 13 $K_0^* := \text{H}_q(c^*)$ 14 $K_1^* \leftarrow_{\S} \mathcal{K}$ 15 $b' \leftarrow \text{A}^{\text{DECAPS}, \mathbb{H} , \mathbb{G} }(pk, c^*, K_b^*)$ 16 return $\llbracket b' = b \rrbracket$	$\underline{\text{DECAPS}}(c \neq c^*)$ 17 return $K := \text{H}_q(c)$ $\underline{\text{G}}(m)$ 18 if $\text{F}(m) = 0$ 19 $\text{G}(m) := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 20 else 21 $\text{G}(m) := \text{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 22 return $\text{G}(m)$
--	---

Figure 16: Adversary C executed in game $\text{GDPB}_{\delta(pk, sk)}$ for the proof of Theorem 3.

It remains to upper bound $|\Pr[G_7^A \Rightarrow 1] - \Pr[G_8^A \Rightarrow 1]|$. To this end, it is sufficient to upper bound the probability that any of the queries to H_q could possibly contain c^* . Each query to H_q is either a classical query, triggered by A querying DECAPS on some ciphertext c , or a query in superposition, triggered by A querying H . Since queries to DECAPS on c^* are explicitly forbidden, the only possibility would be one of A's queries to $|\mathbb{H}\rangle$. A's queries to H trigger queries to H_q that are of the form $\sum_m \alpha_m |\text{Enc}_0(pk, m; \text{G}(m))\rangle$. They cannot contain c^* unless there exists some message m such that $\text{Enc}_0(pk, m; \text{G}(m)) = c^*$, which we claim is impossible. First we consider the case that $m = \hat{m}$: It would be required that $\text{Enc}_0(pk, \hat{m}; \text{G}(\hat{m})) = c^*$, 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 $\text{Enc}_0(pk, m; \text{G}(m)) = c^*$. Since all sampled randomness is good, it is implied that $\hat{m} \neq m = \text{Dec}_0(sk, c^*)$. Since c^* was a random encryption of \hat{m} , the probability of c^* inducing decryption failure can be upper bounded by δ , hence

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

Collecting the probabilities yields the theorem's upper bound.

4 Two-Message Authenticated Key Exchange

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

RUNNING A KEY EXCHANGE PROTOCOL BETWEEN TWO PARTIES. To run a two-message key exchange protocol, the algorithms KG , Init , Der_{init} , and Der_{resp} are executed in an interactive manner between two parties P_i and P_j with key pairs $(sk_i, pk_i), (sk_j, pk_j) \leftarrow \text{KG}$. To execute the protocol, the parties call the algorithms in the following way:

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

$$\begin{array}{ccc}
\underline{\text{Party } P_i} (pk_i, sk_i) & & \underline{\text{Party } P_j} (pk_j, sk_j) \\
(M, st) \leftarrow \text{Init}(sk_i, pk_j) & \xrightarrow{M} & (M', K') \leftarrow \text{Der}_{\text{resp}}(sk_j, pk_i, M) \\
K := \text{Der}_{\text{init}}(sk_i, pk_j, M', st) & \xleftarrow{M'} &
\end{array}$$

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 P_1, \dots, 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 Der_{init} or Der_{resp} .
7. The session key K .

In our security model, the adversary A is given black-box access to the set of processes Init , Der_{resp} and Der_{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 DER_{init} or DER_{resp} , each at most once per session (see Figure 17, page 28). Since both derivation processes can be called on arbitrary input, A may relay their input faithfully as well as modify the data on transit. Moreover, the attacker is additionally granted queries to reveal both secret process data, namely using oracles REVEAL , REV-STATE and CORRUPT (see Figure 18, page 29). Oracles REVEAL and REV-STATE both can be queried on an arbitrary session ID, with oracle REVEAL revealing the respective session's key (if already defined), and oracle REV-STATE revealing the respective session's internal state. Oracle CORRUPT can be queried on an arbitrary number $i \in [N]$ to reveal the respective party's long-term key material. Usage of this oracle allows the attacker to corrupt the test session's holder, the oracle therefore models the possibility of KCI attacks. Combined usage of oracles REV-STATE and CORRUPT allows the attacker to obtain the state as well as the long-term secret key on both sides of the session, the oracles therefore model the possibility of MEX attacks. After choosing a test session, either the session's key or a uniformly random key is returned. The attacker's task is to distinguish these two cases, to this end it outputs a bit.

Definition 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 17 and Figure 18.

We define the IND-AA advantage function of an adversary A against AKE as

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

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

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

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

<u>GAME IND-AA_b</u>		<u>GAME IND-StAA_b</u>	
01 cnt := 0	//session counter	23 cnt := 0	//session counter
02 sID* := 0	//test session's id	24 sID* := 0	//test session's id
03 for n ∈ [N]		25 for n ∈ [N]	
04 (pk _n , sk _n) ← KG		26 (pk _n , sk _n) ← KG	
05 b' ← A ^O (pk ₁ , ..., pk _N)		27 b' ← A ^O (pk ₁ , ..., pk _N)	
06 if Trivial(sID*)		28 if ATTACK(sID*)	
07 return 0		29 return 0	
08 return b'		30 return b'	
<u>EST((i, j) ∈ [N]²)</u>		<u>INIT(sID)</u>	
09 cnt ++		31 if holder[sID] = ⊥	
10 holder[cnt] := i		32 return ⊥	//Session not established
11 peer[cnt] := j		33 if sent[sID] ≠ ⊥ return ⊥	//no re-use
12 return cnt		34 role[sID] := "initiator"	
<u>DER_{resp}(sID, M)</u>		35 (i, j) := (holder[sID], peer[sID])	
13 if holder[sID] = ⊥		36 (M, st) ← lnit(sk _i , pk _j)	
14 return ⊥	//Session not established	37 (sent[sID], state[sID]) := (M, st)	
15 if sKey[sID] ≠ ⊥ return ⊥	//no re-use	38 return M	
16 if role[sID] = "initiator" return ⊥		<u>DER_{init}(sID, M')</u>	
17 role[sID] := "responder"		39 if holder[sID] = ⊥ or state[sID] = ⊥	
18 (j, i) := (holder[sID], peer[sID])		40 return ⊥	//Session not initialised
19 (M', K') ← Der _{resp} (sk _j , pk _i , M)		41 if sKey[sID] ≠ ⊥ return ⊥	//no re-use
20 sKey[sID] := K'		42 (i, j) := (holder[sID], peer[sID])	
21 (received[sID], sent[sID]) := (M, M')		43 st := state[sID]	
22 return M'		44 sKey[sID] := Der _{init} (sk _i , pk _j , M', st)	
		45 received[sID] := M'	

Figure 17: 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 $O := \{\text{EST}, \text{INIT}, \text{DER}_{\text{resp}}, \text{DER}_{\text{init}}, \text{REVEAL}, \text{REV-STATE}, \text{CORRUPT}, \text{TEST}\}$. Oracles REVEAL, REV-STATE, CORRUPT, and TEST are given in Figure 18. 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.

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

Helper procedure Trivial (Figure 18) is used in all games to exclude the possibility of trivial attacks, and helper procedure ATTACK (also Figure 18) 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 $\mathfrak{M}(\text{sID}^*)$ of all matching sessions that were executed throughout the game (see line 55), and A's output bit b' counts in games IND-AA_b only if Trivial returns false, i.e., if test session sID^* was completed and all of the following conditions hold:

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

<u>Trivial(sID*)</u>		//helper procedure to exclude trivial attacks
46	if sKey[sID*] = \perp return true	//test session was never completed
47	$v := \text{false}$	
48	$(i, j) := (\text{holder}[\text{sID}^*], \text{peer}[\text{sID}^*])$	
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 $\text{Der}(sk_i, pk_j, \text{received}[\text{sID}^*], \text{state}[\text{sID}^*])$
52	$\mathfrak{M}(\text{sID}^*) := \emptyset$	//create list of matching sessions
53	for $1 \leq \text{ptr} \leq \text{cnt}$	
54	if (sent[ptr], received[ptr]) = (received[sID*], sent[sID*])	
	and (holder[ptr], peer[ptr]) = (j, i) and role[ptr] \neq role[sID*]	
55	$\mathfrak{M}(\text{sID}^*) := \mathfrak{M}(\text{sID}^*) \cup \{\text{ptr}\}$	//session matches
56	if revealed[ptr] $v := \text{true}$	//A trivially learned the test session's key via matching session
57	if corrupted[j] and revState[ptr]	
58	$v := \text{true}$	//A may simply compute $\text{Der}(sk_j, pk_i, \text{received}[\text{ptr}], \text{state}[\text{ptr}])$
59	if $ \mathfrak{M}(\text{sID}^*) > 1$ return false	//reward for adversary - protocol was not approx. random.
60	if $v = \text{true}$ return true	
61	if $\mathfrak{M}(\text{sID}^*) = \emptyset$ and corrupted[j] return true	//A tampered with test session, knowing sk_j
62	return false	
<u>ATTACK(sID*)</u>		//helper procedure to exclude trivial attacks as well as state-attacks
63	if Trivial(sID*) return true	//trivial attack
64	if $\mathfrak{M}(\text{sID}^*) = \emptyset$ and revState[sID*] return true	//state-attack
65	return false	
<u>REVEAL(sID)</u>	<u>REV-STATE(sID)</u>	<u>TEST(sID)</u> //only one query
66	if sKey[sID] = \perp return \perp	75 sID* := sID
67	revealed[sID] := true	76 if sKey[sID*] = \perp
68	return sKey[sID]	77 return \perp
	74 return state[sID]	78 $K_0^* := \text{sKey}[\text{sID}^*]$
		79 $K_1^* \leftarrow_{\mathfrak{s}} \mathcal{K}$
		80 return K_0^*
<u>CORRUPT($i \in [N]$)</u>		
69	if corrupted[i] return \perp	
70	corrupted[i] := true	
71	return sk_i	

Figure 18: Helper procedures Trivial and ATTACK and oracles REVEAL, REV-STATE, CORRUPT, and TEST of games IND-AA and IND-StAA defined in Figure 17.

same reason as discussed in 2: A could trivially compute $\text{Der}(sk_j, pk_i, \text{received}[\text{sID}], \text{state}[\text{sID}])$ for some matching session sID.

4. A did not both tamper with the test session and obtain the peer j 's secret key sk_j , see line 61. We enforce that $\mathfrak{M}(\text{sID}^*) \neq \emptyset$ or $\neg \text{corrupted}[j]$ to exclude the following trivial attack: A could learn the peer's secret key sk_j via query CORRUPT[j] and either

- receive a message M by querying INIT on sID*, compute $(M', K') \leftarrow \text{Der}_{\text{resp}}(sk_j, pk_i, M)$ without having to call DER_{resp} , and then call $\text{DER}_{\text{init}}(\text{sID}^*, M')$, thereby ensuring that $\text{sKey}[\text{sID}^*] = K'$,
- or compute $(M, \text{st}) \leftarrow \text{Init}(sk_j, pk_i)$ without having to call INIT, receive a message M' by querying $\text{DER}_{\text{resp}}(\text{sID}^*, M)$, and trivially compute $\text{Der}_{\text{init}}(sk_j, pk_i, M', \text{st})$.

A's output bit b' only counts in games IND-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 $\mathfrak{M}(\text{sID}^*) \neq \emptyset$ or $\neg \text{revState}[\text{sID}^*]$ in game IND-StAA for the following reason: In an active attack, given that the test session's internal state got leaked, it is possible for some protocols to choose a message M' such that the result of algorithm $\text{Der}_{\text{init}}(sk_i, pk_j, M', \text{st})$ can be computed without knowledge of any of the long-term keys sk_i or sk_j . In this setting, an adversary might

query INIT on sID^* , learn the internal state st by querying REV-STATE on sID^* , choose its own message M' without a call to DER_{resp} and finally call $DER_{init}(sID^*, M')$, thereby being enabled to anticipate the resulting key.

5 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 formalise 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 [38]. The second way in which security models differ is the concrete formalisation 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 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 formalisations of the same attack concept, say KCI attacks, may differ considerably. Unfortunately, in the literature these formalisations 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 excels is in the way we rigorously formalise 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 formalisation of a very strong security notion covering many advanced security features like weak PFS, KCI security, and security against reflection attacks [38]. 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.

6 Transformation from PKE to AKE

Transformation FO_{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 FO_{AKE} , we achieve IND-StAA-security from CPA security alone.

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

$$AKE = FO_{AKE}[PKE, G, H] = (KG, Init, Der_{resp}, Der_{init}) .$$

The algorithms of AKE are defined in Figure 19.

Init (sk_i, pk_j):	Der_{resp} (sk_j, pk_i, M):	Der_{init} (sk_i, pk_j, M', st):
01 $m_j \leftarrow_{\mathcal{S}} \mathcal{M}$	07 Parse (\tilde{pk}, c_j) := M	18 Parse (c_i, \tilde{c}) := M'
02 $c_j := \text{Enc}(pk_j, m_j; \mathbf{G}(m_j))$	08 $m_i, \tilde{m} \leftarrow_{\mathcal{S}} \mathcal{M}$	19 Parse ($\tilde{sk}, m_j, M := (\tilde{pk}, c_j)$) := st
03 $(\tilde{sk}, \tilde{pk}) \leftarrow \text{KG}$	09 $c_i := \text{Enc}(pk_i, m_i; \mathbf{G}(m_i))$	20 $m'_i := \text{Dec}(sk_i, c_i)$
04 $M := (\tilde{pk}, c_j)$	10 $\tilde{c} := \text{Enc}(\tilde{pk}, \tilde{m}; \mathbf{G}(\tilde{m}))$	21 $\tilde{m}' := \text{Dec}(\tilde{sk}, \tilde{c})$
05 $st := (sk, m_j, M)$	11 $M' := (c_i, \tilde{c})$	22 if $m'_i = \perp$
06 return (M, st)	12 $m'_j := \text{Dec}(sk_j, c_j)$	or $c_i \neq \text{Enc}(pk_i, m'_i; \mathbf{G}(m'_i))$
	13 if $m'_j = \perp$	23 if $\tilde{m}' = \perp$
	or $c_j \neq \text{Enc}(pk_j, m'_j; \mathbf{G}(m'_j))$	24 $K := \text{H}'_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$
	14 $K' := \text{H}'_R(m_i, c_j, \tilde{m}, i, j, M, M')$	25 else
	15 else	26 $K := \text{H}'_{L2}(c_i, m_j, \tilde{m}', i, j, M, M')$
	16 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$	27 else if $\tilde{m}' = \perp$
	17 return (M', K')	28 $K := \text{H}'_{L3}(m'_i, m_j, \tilde{c}, i, j, M, M')$
		29 else $K := \text{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
		30 return K

Figure 19: IND-StAA secure AKE protocol $\text{AKE} = \text{FO}_{\text{AKE}}[\text{PKE}, \mathbf{G}, \mathbf{H}]$. Oracles H'_R and H'_{L1} , H'_{L2} and H'_{L3} are used to generate random values whenever reencryption fails. (For encryption, this strategy is called *implicit reject* Amongst others, it is used in [29], [44] and [33].) For simplicity of the proof, H'_R and H'_{L1} , H'_{L2} and H'_{L3} are internal random oracles that cannot be accessed directly. For implementation, it would be sufficient to use a PRF.

IND-StAA SECURITY OF FO_{AKE} . The following theorem establishes that IND-StAA security of AKE reduces to DS and IND-CPA security of PKE (see Definition 6).

Theorem 6.1 (PKE DS + IND-CPA \Rightarrow AKE IND-StAA). *Assume PKE to be δ -correct, and to come with a sampling algorithm $\overline{\text{Enc}}$ such that it is ϵ -disjoint. Then, for any IND-StAA adversary \mathbf{B} that establishes S sessions and issues at most q_R (classical) queries to REVEAL, at most q_G (quantum) queries to random oracle \mathbf{G} and at most q_H (quantum) queries to random oracle \mathbf{H} , there exists an adversary \mathbf{A}_{DS} against the disjoint simulatability of $\text{T}[\text{PKE}, \mathbf{G}]$ issuing at most $q_G + 2q_H + 3S$ queries to \mathbf{G} such that*

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

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

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

and the running times of \mathbf{C}_{DS} and \mathbf{C}_{IND} is about that of \mathbf{B} .

PROOF SKETCH. To prove IND-StAA security of $\text{FO}_{\text{AKE}}[\text{PKE}, \mathbf{G}, \mathbf{H}]$, we consider an adversary \mathbf{B} with black-box access to the protocols' algorithms and to oracles that reveal keys of completed sessions, internal states, and long-term secret keys of participating parties as specified in game IND-StAA (see Figure 17). Intuitively, \mathbf{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 \mathbf{P}_i and \mathbf{P}_j :

1. We first consider the case that \mathbf{B} executed the test session honestly. Note that on the right-hand side of the protocol there exists no state. We assume that \mathbf{B} has learned the secret key of party \mathbf{P}_j and hence knows m_j . Additionally, \mathbf{B} could either learn the secret key of party \mathbf{P}_i and thereby, compute m_i , or the state on the left-hand side of the protocol including \tilde{sk} , and thereby, compute \tilde{m} , but not both.

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

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

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 [44] 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 $(\text{Enc}', \text{Dec}')$ (relative to the patched G) provides perfect correctness. This strategy is the AKE analogue to the technique used in our analysis of the Fujisaki-Okamoto transformation given in Section 3, in particular, during our proof of Theorem 2. The latter also explains why our transformation does not work with any deterministic encryption scheme, but only with the ones that are derived by using transformation T . For more details on this issue, we refer to Section 3.2.

Proof. Let B be an adversary against the IND-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 $\mathfrak{M}(\text{sID}^*) \neq \emptyset$, where $\mathfrak{M}(\text{sID}^*)$ is defined in Figure 18, line 55, as the list of matching sessions that were executed throughout game IND-StAA), in the second part we will examine the case that B tampered with the test session (i.e., the case that $\mathfrak{M}(\text{sID}^*) = \emptyset$).

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

Lemma 6.2 *There exists an adversary A such that*

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

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

The upper bound is proven in appendix A. 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 IND-CPA security, and the respective ciphertext can hence be replaced with a fake ciphertext (due to $T[\text{PKE}, G]$'s disjoint simulatability).

Lemma 6.3 *There exists an adversary A' such that*

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

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

The upper bound is proven in appendix B. 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 $\text{role}[\text{sID}^*] = \text{"initiator"}$ or $\text{role}[\text{sID}^*] = \text{"responder"}$, we can

rely on the secrecy of either m_i^* or m_j^* .

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

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

6.1 IND-StAA security without disjoint simulatability

The following theorem establishes that plugging in transformation `Punc` before using FO_{AKE} achieves IND-StAA security from IND-CPA security alone, as long as PKE is γ -spread.

Theorem 6.4 (IND-StAA security of $\text{FO}_{\text{AKE}} \circ \text{Punc}$). *Assume PKE_0 to be δ -correct and γ -spread, and let $\hat{m} \in \mathcal{M}$. Let $\text{AKE} := \text{FO}_{\text{AKE}}[\text{Punc}[\text{PKE}, \hat{m}], \mathbb{G}, \mathbb{H}]$. 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 \mathbb{G} and at most q_H (quantum) queries to random oracle \mathbb{H} , there exist adversaries B_1 and B_2 such that*

$$\begin{aligned} \text{Adv}_{\text{AKE}}^{\text{IND-StAA}}(B) & \leq 2S \cdot (S + 3 \cdot N) \cdot \left(\text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_1) + 2\sqrt{q \cdot \text{Adv}_{\text{PKE}_0}^{\text{IND-CPA}}(B_2)} \right) \\ & \quad + (S + 3N) \cdot (8q^2 \cdot (S + 4) + S) \cdot \delta + S \cdot (S + 3N) \cdot 2^{-\gamma} \\ & \quad + \frac{S(8q \cdot (S + 3N) + S^2)}{\sqrt{|\mathcal{M}|} - 1} + S \cdot (3S^2 + 2) \cdot \mu(\text{KG}) , \end{aligned}$$

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

Since the proof is essentially similar to the proof of Theorem 4 (like the proof of Theorem 3 is similar to the proof of Theorem 2), it is outsourced to Appendix C.

Acknowledgments

We would like to thank the anonymous reviewers of Eurocrypt 2018, Crypto 2019 and Asiacrypt 2019 for their helpful comments and suggestions. This work was supported by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC 2092 CASA, 390781972), and ERC Project ERCC (FP7/615074).

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 11.)
- [2] Alawatugoda, J., Boyd, C., Stebila, D.: Continuous after-the-fact leakage-resilient key exchange. In: Susilo, W., Mu, Y. (eds.) ACISP 14: 19th Australasian Conference on Information Security and Privacy. Lecture Notes in Computer Science, vol. 8544, pp. 258–273. Springer, Heidelberg, Germany, Wollongong, NSW, Australia (Jul 7–9, 2014) (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, 7, 11, 12, 17.)
- [4] Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th Annual Symposium on Foundations of Computer Science. pp. 474–483. IEEE Computer Society Press, Philadelphia, PA, USA (Oct 18–21, 2014) (Cited on page 12, 13.)

- [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 7.)
- [6] Beals, R., Buhrman, H., Cleve, R., Mosca, M., Wolf, R.: Quantum lower bounds by polynomials. In: 39th Annual Symposium on Foundations of Computer Science. pp. 352–361. IEEE Computer Society Press, Palo Alto, CA, USA (Nov 8–11, 1998) (Cited on page 11.)
- [7] Bellare, M., Boldyreva, A., O’Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 535–552. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007) (Cited on page 15.)
- [8] Bellare, M., Canetti, R., Krawczyk, H.: A modular approach to the design and analysis of authentication and key exchange protocols (extended abstract). In: 30th Annual ACM Symposium on Theory of Computing. pp. 419–428. ACM Press, Dallas, TX, USA (May 23–26, 1998) (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: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 93: 1st Conference on Computer and Communications Security. pp. 62–73. ACM Press, Fairfax, Virginia, USA (Nov 3–5, 1993) (Cited on page 3.)
- [11] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) Advances in Cryptology – CRYPTO’93. Lecture Notes in Computer Science, vol. 773, pp. 232–249. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 22–26, 1994) (Cited on page 1, 5.)
- [12] Bellare, M., Rogaway, P.: The security of triple encryption and a framework for code-based game-playing proofs. In: Vaudenay, S. (ed.) Advances in Cryptology – EUROCRYPT 2006. Lecture Notes in Computer Science, vol. 4004, pp. 409–426. Springer, Heidelberg, Germany, St. Petersburg, Russia (May 28 – Jun 1, 2006) (Cited on page 9.)
- [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, 8.)
- [14] Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of cca security in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/590 (2019), <https://eprint.iacr.org/2019/590> (Cited on page 7, 8.)
- [15] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011) (Cited on page 2, 11.)
- [16] Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013, Part II. Lecture Notes in Computer Science, vol. 8043, pp. 361–379. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013) (Cited on page 11.)
- [17] 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, 7.)
- [18] Boyd, C., Cliff, Y., Nieto, J.G., Paterson, K.G.: Efficient one-roundkey exchange in the standard model. ACISP 08: 13th Australasian Conference on Information Security and Privacy (2008) (Cited on page 1, 2, 5.)

- [19] Broadbent, A., Jeffery, S.: Quantum homomorphic encryption for circuits of low T-gate complexity. In: Gennaro, R., Robshaw, M.J.B. (eds.) *Advances in Cryptology – CRYPTO 2015, Part II*. Lecture Notes in Computer Science, vol. 9216, pp. 609–629. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015) (Cited on page 11.)
- [20] Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) *Advances in Cryptology – EUROCRYPT 2001*. Lecture Notes in Computer Science, vol. 2045, pp. 453–474. Springer, Heidelberg, Germany, Innsbruck, Austria (May 6–10, 2001) (Cited on page 1.)
- [21] 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 4.)
- [22] Dent, A.W.: A designer’s guide to KEMs. In: Paterson, K.G. (ed.) *9th IMA International Conference on Cryptography and Coding*. Lecture Notes in Computer Science, vol. 2898, pp. 133–151. Springer, Heidelberg, Germany, Cirencester, UK (Dec 16–18, 2003) (Cited on page 4.)
- [23] Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Cachin, C., Camenisch, J. (eds.) *Advances in Cryptology – EUROCRYPT 2004*. Lecture Notes in Computer Science, vol. 3027, pp. 342–360. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004) (Cited on page 2.)
- [24] Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*. Lecture Notes in Computer Science, vol. 7293, pp. 467–484. Springer, Heidelberg, Germany, Darmstadt, Germany (May 21–23, 2012) (Cited on page 1, 2, 6.)
- [25] Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In: Chen, K., Xie, Q., Qiu, W., Li, N., Tzeng, W.G. (eds.) *ASIACCS 13: 8th ACM Symposium on Information, Computer and Communications Security*. pp. 83–94. ACM Press, Hangzhou, China (May 8–10, 2013) (Cited on page 1, 3.)
- [26] Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) *Advances in Cryptology – CRYPTO’99*. Lecture Notes in Computer Science, vol. 1666, pp. 537–554. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 1999) (Cited on page 3, 9.)
- [27] 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.)
- [28] Gagliardini, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Robshaw, M., Katz, J. (eds.) *Advances in Cryptology – CRYPTO 2016, Part III*. Lecture Notes in Computer Science, vol. 9816, pp. 60–89. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016) (Cited on page 11.)
- [29] Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) *TCC 2017: 15th Theory of Cryptography Conference, Part I*. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017) (Cited on page 3, 4, 9, 17, 18, 32.)
- [30] Howgrave-Graham, N., Silverman, J.H., Whyte, W.: Choosing parameter sets forwithand. In: Menezes, A. (ed.) *Topics in Cryptology – CT-RSA 2005*. Lecture Notes in Computer Science, vol. 3376, pp. 118–135. Springer, Heidelberg, Germany, San Francisco, CA, USA (Feb 14–18, 2005) (Cited on page 2.)

- [31] Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I. Lecture Notes in Computer Science, vol. 9614, pp. 387–416. Springer, Heidelberg, Germany, Taipei, Taiwan (Mar 6–9, 2016) (Cited on page 12, 13, 14.)
- [32] Jager, T., Kohlar, F., Schäge, S., Schwenk, J.: On the security of TLS-DHE in the standard model. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 273–293. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012) (Cited on page 1.)
- [33] Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 96–125. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018) (Cited on page 3, 5, 7, 18, 32.)
- [34] 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.)
- [35] Jiang, H., Zhang, Z., Ma, Z.: On the non-tightness of measurement-based reductions for key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/494 (2019), <https://eprint.iacr.org/2019/494> (Cited on page 8.)
- [36] Jiang, H., Zhang, Z., Ma, Z.: Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/134 (2019), <https://eprint.iacr.org/2019/134> (Cited on page 7.)
- [37] Kiltz, E., Lyubashevsky, V., Schaffner, C.: A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) Advances in Cryptology – EUROCRYPT 2018, Part III. Lecture Notes in Computer Science, vol. 10822, pp. 552–586. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018) (Cited on page 11.)
- [38] Krawczyk, H.: HMQV: A high-performance secure Diffie-Hellman protocol. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 546–566. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005) (Cited on page 1, 5, 8, 30, 31.)
- [39] LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007: 1st International Conference on Provable Security. Lecture Notes in Computer Science, vol. 4784, pp. 1–16. Springer, Heidelberg, Germany, Wollongong, Australia (Nov 1–2, 2007) (Cited on page 1, 5.)
- [40] Li, Y., Schäge, S.: No-match attacks and robust partnering definitions: Defining trivial attacks for security protocols is not trivial. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017: 24th Conference on Computer and Communications Security. pp. 1343–1360. ACM Press, Dallas, TX, USA (Oct 31 – Nov 2, 2017) (Cited on page 1.)
- [41] 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 7.)
- [42] NIST: National institute for standards and technology. postquantum crypto project (2017), <http://csrc.nist.gov/groups/ST/post-quantum-crypto/> (Cited on page 2.)
- [43] Persichetti, E.: Improving the efficiency of code-based cryptography. Ph.D. thesis (2012), <http://persichetti.webs.com/Thesis%20Final.pdf> (Cited on page 4.)

- [44] Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) *Advances in Cryptology – EURO-CRYPT 2018, Part III*. Lecture Notes in Computer Science, vol. 10822, pp. 520–551. Springer, Heidelberg, Germany, Tel Aviv, Israel (Apr 29 – May 3, 2018) (Cited on page 3, 4, 5, 6, 9, 11, 14, 15, 17, 18, 21, 32, 33.)
- [45] Schäge, S.: TOPAS: 2-pass key exchange with full perfect forward secrecy and optimal communication complexity. In: Ray, I., Li, N., Kruegel, C. (eds.) *ACM CCS 2015: 22nd Conference on Computer and Communications Security*. pp. 1224–1235. ACM Press, Denver, CO, USA (Oct 12–16, 2015) (Cited on page 1.)
- [46] 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 9.)
- [47] 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.)
- [48] Yao, A.C.C., Zhao, Y.: OAKE: a new family of implicitly authenticated Diffie-Hellman protocols. In: Sadeghi, A.R., Gligor, V.D., Yung, M. (eds.) *ACM CCS 2013: 20th Conference on Computer and Communications Security*. pp. 1113–1128. ACM Press, Berlin, Germany (Nov 4–8, 2013) (Cited on page 1.)
- [49] Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) *Advances in Cryptology – CRYPTO 2012*. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012) (Cited on page 11.)

A Proof of Lemma 5

FAITHFUL EXECUTION OF THE PROTOCOL ($\mathfrak{M}(\text{sID}^*) \neq \emptyset$). Recall that we are proving an upper bound for $|\Pr[\text{IND-StAA}_1^{\text{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) \neq \emptyset] - \Pr[\text{IND-StAA}_0^{\text{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) \neq \emptyset]|$. First, we will enforce that indeed, we only need to consider the case where $\mathfrak{M}(\text{sID}^*) \neq \emptyset$, 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 IND-StAA_b ,

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

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

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

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

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

for both bits b . We claim

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

To verify this bound, we first consider the case that $\text{role}[\text{sID}^*] = \text{"initiator"}$: Let $i^* := \text{holder}[\text{sID}^*]$ and $j^* := \text{peer}[\text{sID}^*]$. To create more than one matching session, B has to establish and derive two

GAMES $G_{0,b} - G_{2,b}$	$\text{DER}_{\text{resp}}(\text{sID}, M = (\tilde{pk}, c_j))$
01 $\text{sID}, \text{sID}^* := 0$	22 if $\text{holder}[\text{sID}] = \perp$ or $\text{sKey}[\text{sID}] \neq \perp$
02 for $n \in [N]$	or $\text{role}[\text{sID}] = \text{"initiator"}$ return \perp
03 $(pk_n, sk_n) \leftarrow \text{KG}$	23 $\text{role}[\text{sID}] := \text{"responder"}$
04 $b' \leftarrow \mathcal{B}^{\text{O}, \mathcal{G} , \mathcal{H} }((pk_n)_{n \in [N]})$	24 $(j, i) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
05 if $\text{ATTACK}(\text{sID}^*)$	25 $m_i, \tilde{m} \leftarrow_{\mathcal{S}} \mathcal{M}$
06 return 0	26 $c_i := \text{Enc}(pk_i, m_i; \mathcal{G}(m_i))$
07 if $\mathfrak{M}(\text{sID}^*) = \emptyset$ ABORT // $G_{1,b}$	27 $\tilde{c} := \text{Enc}(pk, \tilde{m}; \mathcal{G}(\tilde{m}))$
08 if $ \mathfrak{M}(\text{sID}^*) > 1$	28 $M' := (c_i, \tilde{c})$
09 ABORT // $G_{2,b}$	29 $m'_j := \text{Dec}(sk_j, c_j)$
10 return b'	30 if $m'_j = \perp$ or $c_j \neq \text{Enc}(pk_j, m'_j; \mathcal{G}(m'_j))$
	31 $K' := \text{H}'_{\mathcal{R}}(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
	32 else $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
	33 $\text{sKey}[\text{sID}] := K'$
	34 $(\text{received}[\text{sID}], \text{sent}[\text{sID}]) := (M, M')$
	35 return M'
INIT (sID)	$\text{DER}_{\text{init}}(\text{sID}, M' = (c_i, \tilde{c}))$
11 if $\text{holder}[\text{sID}] = \perp$	36 if $\text{holder}[\text{sID}] = \perp$ or $\text{state}[\text{sID}] = \perp$
or $\text{sent}[\text{sID}] \neq \perp$ return \perp	or $\text{sKey}[\text{sID}] \neq \perp$ return \perp
12 $\text{role}[\text{sID}] := \text{"initiator"}$	37 $(i, j) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
13 $i := \text{holder}[\text{sID}]$	38 $(\tilde{sk}, m_j, M := (pk, c_j)) := \text{state}[\text{sID}]$
14 $j := \text{peer}[\text{sID}]$	39 $m'_i := \text{Dec}(sk_i, c_i)$
15 $m_j \leftarrow_{\mathcal{S}} \mathcal{M}$	40 $\tilde{m}' := \text{Dec}(\tilde{sk}, \tilde{c})$
16 $c_j := \text{Enc}(pk_j, m_j; \mathcal{G}(m_j))$	41 if $m'_i = \perp$ or $c_i \neq \text{Enc}(pk_i, m'_i; \mathcal{G}(m'_i))$
17 $(\tilde{pk}, \tilde{sk}) \leftarrow \text{KG}$	42 if $\tilde{m}' = \perp$
18 $M := (\tilde{pk}, c_j)$	43 $K := \text{H}'_{\mathcal{L}1}(c_i, m_j, \tilde{c}, i, j, M, M')$
19 $\text{state}[\text{sID}] := (\tilde{sk}, m_j, M)$	44 else
20 $\text{sent}[\text{sID}] := M$	45 $K := \text{H}'_{\mathcal{L}2}(c_i, m_j, \tilde{m}', i, j, M, M')$
21 return M	46 else if $\tilde{m}' = \perp$
	47 $K := \text{H}'_{\mathcal{L}3}(m'_i, m_j, \tilde{c}, i, j, M, M')$
	48 else $K := \text{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$
	49 $\text{sKey}[\text{sID}] := K$
	50 $\text{received}[\text{sID}] := M'$

Figure 20: Games $G_{0,b} - G_{2,b}$ for case one of the proof of Theorem 4. Helper procedure `ATTACK` and oracles `TEST`, `EST`, `CORRUPT`, `REVEAL` and `REV-STATE` remains as in the original `IND-StAA` game (see Figures 17 and 18).

distinct ("responder") sessions $\text{sID} \neq \text{sID}'$ with holder j^* and peer i^* via oracle call to DER_{resp} , such that $\text{sent}[\text{sID}] = \text{sent}[\text{sID}']$. This means that for $(c_i, \tilde{c}) := \text{sent}[\text{sID}]$ and $(c'_i, \tilde{c}') := \text{sent}[\text{sID}']$ it holds that both $c_i = c'_i$ and $\tilde{c} = \tilde{c}'$. All ciphertexts were generated by faithfully executing DER_{resp} , and therefore encryptions of messages that were drawn at random. Since both \tilde{pk} and pk_{j^*} were also generated faithfully,

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

Now we consider the case that $\text{role}[\text{sID}^*] = \text{"responder"}$: Let $j^* := \text{holder}[\text{sID}^*]$ and $i^* := \text{peer}[\text{sID}^*]$. To create more than one matching session, \mathcal{B} has to establish and derive two distinct sessions $\text{sID} \neq \text{sID}'$ with holder i^* and peer j^* via oracle call to `INIT` such that $\text{sent}[\text{sID}] = \text{sent}[\text{sID}']$. This means that for $(\tilde{pk}, c_j) := \text{sent}[\text{sID}]$ and $(\tilde{pk}', c'_j) := \text{sent}[\text{sID}']$ 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

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

So far, we established

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

Since games $G_{2,b}$ abort unless $|\mathfrak{M}(\text{sID}^*)| = 1$, 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 $\text{sID}_{\text{init}}^*$ denote the "initiator" session, i.e., pick $\text{sID}_{\text{init}}^* \in \{\text{sID}^*, \text{sID}'\}$ such that $\text{role}[\text{sID}_{\text{init}}^*] = \text{"initiator"}$, and let $\text{sID}_{\text{resp}}^*$ denote the other session. B's bit b' only counts in IND-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 $(\neg\text{st})$: the case that the initialising session's state was not revealed, i.e., the case that $\neg\text{revState}[\text{sID}_{\text{init}}^*]$,
- and case $(\neg\text{sk})$: the case that the initialising session's holder was not corrupted, i.e., the case that $\neg\text{corrupted}[\text{holder}[\text{sID}_{\text{init}}^*]]$

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

$$\begin{aligned} |\Pr[G_{2,1}^{\text{B}} \Rightarrow 1] - \Pr[G_{2,0}^{\text{B}} \Rightarrow 1]| & \leq |\Pr[G_{2,1}^{\text{B}} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G_{2,0}^{\text{B}} \Rightarrow 1 \wedge \neg\text{st}]| \\ & \quad + |\Pr[G_{2,1}^{\text{B}} \Rightarrow 1 \wedge \neg\text{sk}] - \Pr[G_{2,0}^{\text{B}} \Rightarrow 1 \wedge \neg\text{sk}]| . \end{aligned}$$

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

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

The proof is given in in Appendix A.1. Its main idea is that since the initialising session's state (in particular, ephemeral secret key $\tilde{s}k^*$) remains unrevealed throughout the game, at least message \tilde{m}^* (that was randomly picked by $\text{DER}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$) 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 $\tilde{s}k^*$ any longer. Since PKE is DS (and hence, so is $\text{T}[\text{PKE}, \text{G}]$, see Lemma 2), we can decouple the test session's key from \tilde{m}^* by replacing $\tilde{c} = \text{Enc}(\tilde{p}k, \tilde{m}^*; \text{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\bar{\text{Enc}}$, 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 $(\neg\text{sk})$. We claim that there exists an adversary $\text{A}_{\text{DS}}^{\neg\text{sk}}$ such that

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

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

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

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

the upper bound given in Lemma 5.

A.1 Case (\neg st) of the Proof of Lemma 5

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 $\neg \text{revState}[\text{sID}_{\text{init}}^*]$. Afterwards, we ensure that the game makes no use of ephemeral secret key \tilde{sk}^* of $\text{sID}_{\text{init}}^*$ any longer by patching encryption into the random oracle (in games $G_{2,b}^{\neg\text{st}}$ to $G_{10,b}^{\neg\text{st}}$, see Figure 21 and 22). Next, during execution of $\text{DER}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$, we replace $\tilde{c} = \text{Enc}(\tilde{pk}^*, \tilde{m}^*; \mathbf{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\overline{\text{Enc}}$ (games $G_{11,b}^{\neg\text{st}}$ to $G_{12,b}^{\neg\text{st}}$, Figure 24, see line 25). We show that after those changes, \mathbf{B} 's view does not change with overwhelming probability if we change TEST such that it always returns a random value (game $G_{14,0}^{\neg\text{st}}$, also Figure 24).

GAMES $G_{2,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	INIT(sID)
01 cnt, sID* := 0	15 if holder[sID] = \perp
02 $s'_{\text{init}} \leftarrow_{\$} [S]$ // $G_{4,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	or sent[sID] $\neq \perp$ return \perp
03 for $n \in [N]$	16 role[sID] := "initiator"
04 $(pk_n, sk_n) \leftarrow \text{KG}$	17 $i := \text{holder}[\text{sID}]$
05 $(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \text{KG}$ // $G_{5,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	18 $j := \text{peer}[\text{sID}]$
06 $b' \leftarrow \mathbf{B}^{\text{O}, \mathbf{G} , \mathbf{H} }((pk_n)_{n \in [N]})$	19 $m_j \leftarrow_{\$} \mathcal{M}$
07 if ATTACK(sID*)	20 $c_j := \text{Enc}(pk_j, m_j; \mathbf{G}(m_j))$
08 return 0	21 $(\tilde{pk}, \tilde{sk}) \leftarrow \text{KG}$
09 if $ \mathcal{M}(\text{sID}^*) \neq 1$ ABORT	22 if sID $\neq s'_{\text{init}}$ and $\tilde{pk} = \tilde{pk}^*$
10 if revState[sID* _{init}] ABORT // $G_{3,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	23 ABORT // $G_{6,b}^{\neg\text{st}}$
11 Pick sID* _{init} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.	24 if sID = s'_{init}
role[sID* _{init}] = "initiator" // $G_{4,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	25 $(\tilde{pk}, \tilde{sk}) := (\tilde{pk}^*, \tilde{sk}^*)$ // $G_{5,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$
12 if sID* _{init} $\neq s'_{\text{init}}$	26 $M := (\tilde{pk}, c_j)$
13 return 0 // $G_{4,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$	27 state[sID] := (\tilde{sk}, m_j, M)
14 return b'	28 sent[sID] := M
	29 return M

Figure 21: Games $G_{2,b}^{\neg\text{st}} - G_{6,b}^{\neg\text{st}}$ for case (\neg st) of the proof of Lemma 5. Oracles DER_{resp} , DER_{init} and TEST remain as in games $G_{0,b}^{\neg\text{st}}$ (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 17 and Figure 18, pages 28 and 29).

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

$$\begin{aligned} & |\Pr[G_{2,1}^{\text{B}} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G_{2,0}^{\text{B}} \Rightarrow 1 \wedge \neg\text{st}]| \\ &= |\Pr[G_{2,1}^{\neg\text{stB}} \Rightarrow 1 \wedge \neg\text{st}] - \Pr[G_{2,0}^{\neg\text{stB}} \Rightarrow 1 \wedge \neg\text{st}]|. \end{aligned}$$

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

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

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

GAMES $G_{4,b}^{\neg\text{st}}$. In both games $G_{4,b}^{\neg\text{st}}$, one of the sessions that get established during execution of \mathbf{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 $\text{sID}_{\text{init}}^*$. Since games $G_{4,b}^{\neg\text{st}}$ and $G_{3,b}^{\neg\text{st}}$ proceed identically for both bits b if $s'_{\text{init}} = \text{sID}_{\text{init}}^*$, and since games $G_{4,b}^{\neg\text{st}}$ output 0 if $s'_{\text{init}} \neq \text{sID}_{\text{init}}^*$,

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

GAMES $G_{5,b}^{\neg\text{st}}$. In both games $G_{5,b}^{\neg\text{st}}$, 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 $\text{sID}_{\text{init}}^*$. This

change is only conceptual, hence

$$\Pr[G_{4,b}^{-\text{st}} \Rightarrow 1] = \Pr[G_{5,b}^{-\text{st}} \Rightarrow 1] .$$

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

$$|\Pr[G_{5,b}^{-\text{st}} \Rightarrow 1] - \Pr[G_{6,b}^{-\text{st}} \Rightarrow 1]| \leq (S-1) \cdot \mu(\text{KG}) .$$

So far, we established

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

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 $G_{7,b}^{-\text{st}}$. In game $G_{7,b}^{-\text{st}}$, 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

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

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

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

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

To upper bound $|\Pr[G_{6,b}^{-\text{st}} \Rightarrow 1] - \Pr[G_{7,b}^{-\text{st}} \Rightarrow 1]|$ for each bit b , we construct (unbounded, quantum) adversaries C^b against the generic distinguishing problem with bounded probabilities GDPB_λ (see helper Lemma 1, page ??) in Figure 23, issuing at most $q_G + 3 \cdot S$ queries to F : Each C^b runs $(pk, sk) \leftarrow \text{KG}$ and uses this key pair as $(\tilde{pk}^*, \tilde{sk}^*)$ when simulating game $G_{6,b}^{-\text{st}}$. 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 $\text{GDPB}_{\lambda,1}$, 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 $G_{6,b}^{-\text{st}}$ if executed in game $\text{GDPB}_{\lambda,1}$. Since adversary C^b also perfectly simulates game $G_{7,b}^{-\text{st}}$ if executed in game $\text{GDPB}_{\lambda,0}$,

$$|\Pr[G_{6,b}^{-\text{st}} \Rightarrow 1] - \Pr[G_{7,b}^{-\text{st}} \Rightarrow 1]| = |\Pr[\text{GDPB}_{\lambda,1}^{\text{C}^b} = 1] - \Pr[\text{GDPB}_{\lambda,0}^{\text{C}^b} = 1]| ,$$

and according to Lemma 1,

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

Recall that the goal is to simulate the game without knowledge of \tilde{sk}^* . To this end, we will first change DER_{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 $G_{8,b}^{-\text{st}}$. In games $G_{8,b}^{-\text{st}}$, oracle DER_{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 $K := \mathsf{H}'_{\text{L1}}(c_i, m_j, \tilde{c}, i, j, M, M')$, as opposed to letting $K := \mathsf{H}'_{\text{L2}}(c_i, m_j, \tilde{m}', i, j, M, M')$ 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 $\text{DER}_{\text{init}}(s'_{\text{init}})$

<p>GAMES $G_{6,b}^{\neg\text{st}} - G_{10,b}^{\neg\text{st}}$</p> <pre> 01 cnt, sID* := 0 02 s'_{init} ←_§ [S] 03 for n ∈ [N] 04 (pk_n, sk_n) ← KG 05 (pk*, sk*) ← KG 06 G ←_§ R^M 07 Pick 2q-wise hash f 08 G := G_{pk*, sk*} 09 b' ← B^{O, (G), (H)}((pk_n)_{n ∈ [N]}) 10 if ATTACK(sID*) 11 return 0 12 if P(sID*) ≠ 1 ABORT 13 if revState[sID*_{init}] ABORT 14 Pick sID*_{init} ∈ {sID*, sID'} s. th. 15 role[sID*_{init}] = "initiator" 16 if sID*_{init} ≠ s'_{init} return 0 17 return b' DER_{resp}(sID, M = (pk, c_j)) 17 if holder[sID] = ⊥ or sKey[sID] ≠ ⊥ or role[sID] = "initiator" return ⊥ 18 role[sID] := "responder" 19 (j, i) := (holder[sID], peer[sID]) 20 m_i, m̃ ←_§ M 21 c_i := Enc(pk_i, m_i; G(m_i)) 22 c̃ := Enc(pk, m̃; G(m̃)) 23 M' := (c_i, c̃) 24 m'_j := Dec(sk_j, c_j) 25 if m'_j = ⊥ or c_j ≠ Enc(pk_j, m'_j; G(m'_j)) 26 K' := H'_R(m_i, c_j, m̃, i, j, c_i, M, M') 27 else 28 K' := H(m_i, m'_j, m̃, i, j, M, M') 29 if pk = pk* 30 K' := H_q(m_i, m'_j, c̃, i, j, M, M') // G_{9,b}^{\neg\text{st}} - G_{10,b}^{\neg\text{st}} 31 sKey[sID] := K' 32 (received[sID], sent[sID]) := (M, M') 33 return M' </pre>	<p>DER_{init}(sID, M' = (c_i, c̃))</p> <pre> 34 if holder[sID] = ⊥ or state[sID] = ⊥ or sKey[sID] ≠ ⊥ return ⊥ 35 (i, j) := (holder[sID], peer[sID]) 36 (sk, m_j, M := (pk, c_j)) := state[sID] 37 m'_i := Dec(sk_i, c_i) 38 m̃' := Dec(sk, c̃) 39 if m'_i = ⊥ or c_i ≠ Enc(pk_i, m'_i; G(m'_i)) 40 if m̃' = ⊥ 41 K := H'_{L1}(c_i, m_j, c̃, i, j, M, M') 42 else 43 K := H'_{L2}(c_i, m_j, m̃', i, j, M, M') 44 if sID = s'_{init} 45 K := H'_{L1}(c_i, m_j, c̃, i, j, M, M') // G_{8,b}^{\neg\text{st}} - G_{10,b}^{\neg\text{st}} 46 else 47 if m̃' = ⊥ 48 K := H'_{L3}(m'_i, m_j, c̃, i, j, M, M') 49 else 50 K := H(m'_i, m_j, m̃', i, j, M, M') 51 if sID = s'_{init} 52 K := H_q(m'_i, m_j, c̃, i, j, M, M') // G_{9,b}^{\neg\text{st}} 53 sKey[sID] := K 54 received[sID] := M' G_{pk*, sk*}(m) 55 r := Sample(R \ R_{bad}(pk*, sk*, m); f(m)) 56 return r H(m_1, m_2, m_3, i, j, M = (pk, c_j), M') // G_{9,b}^{\neg\text{st}} 57 if pk = pk* 58 return H_q(m_1, m_2, Enc(pk, m_3; G(m_3)), i, j, M, M') 59 return H'(m_1, m_2, m_3, pk, i, j) </pre>
---	---

Figure 22: Games $G_{6,b}^{\neg\text{st}} - G_{10,b}^{\neg\text{st}}$ for case ($\neg\text{st}$) of the proof of Lemma 5. Oracle Init remains as in games $G_{4,b}^{\neg\text{st}}$ (see Figure 21, page 42), (see Figure 17, page 28), 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 $q := q_G + q_H + S$, that cannot be accessed by \mathcal{B} . $\text{Sample}(Y)$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_{\S} Y$. $\text{Sample}(Y; f(m))$ denotes the deterministic execution of $\text{Sample}(Y)$ using explicitly given randomness $f(m)$.

uses ephemeral key pair $(\tilde{pk}^*, \tilde{sk}^*)$. Furthermore, let m_j^* be the message that was picked by $\text{INIT}(s'_{\text{init}})$, let (c_i^*, \tilde{c}^*) denote the message that is received by $\text{DER}_{\text{init}}(s'_{\text{init}})$, and let $\tilde{m}^* := \text{Dec}(\tilde{sk}^*, \tilde{c}^*)$. To distinguish the games, both values $H'_{L1}(c_i^*, m_j^*, \tilde{c}^*, i^*, j^*, M, M')$ and $H'_{L2}(c_i^*, m_j^*, \tilde{m}^*, \tilde{pk}^*, i^*, j^*, M, M')$ must be obtained. But both H'_{L1} and H'_{L2} are internal random oracles that cannot be accessed directly by \mathcal{B} . The only way to obtain oracle values of H'_{L1} and H'_{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 \mathcal{B} queries REVEAL on $\text{sID}^*_{\text{init}}$ or if $\text{sID}^*_{\text{init}} \neq s'_{\text{init}}$. Therefore, to distinguish $H'_{L1}(c_i^*, m_j^*, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c_j^*), M')$ from $H'_{L2}(c_i, m_j, \tilde{m}', \tilde{pk}^*, i, j, (\tilde{pk}^*, c_j^*), M')$ without triggering the game to output 0, another session $s \neq s'_{\text{init}}$ would have to be established and initialised, and it would be

$C_1^b = D_1^b$ 01 $(pk, sk) \leftarrow \text{KG}$ 02 for $m \in \mathcal{M}$ 03 $\lambda(m) := \delta(pk, sk, m)$ 04 return $(\lambda(m))_{m \in \mathcal{M}}$ $G(m)$ 05 if $F(m) = 0$ 06 $G(m) := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 07 else 08 $G(m) := \text{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 09 return $G(m)$	$C_2^{b F}, D_2^{b F}$ 10 $\text{cnt}, \text{sID}^* := 0$ 11 $s'_{\text{init}} \leftarrow_{\mathcal{S}} [S]$ 12 for $n \in [N]$ 13 $(pk_n, sk_n) \leftarrow \text{KG}$ 14 $(\tilde{pk}^*, \tilde{sk}^*) := (pk, sk)$ 15 Pick $2q$ -wise hash f 16 $b' \leftarrow \mathbf{B}^{O, G , H }((pk_n)_{n \in [N]})$ 17 if $\text{ATTACK}(\text{sID}^*)$ 18 return 0 19 if $ \mathcal{M}(\text{sID}^*) \neq 1$ ABORT 20 if $\text{revState}[\text{sID}^*_{\text{init}}]$ ABORT 21 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. $\text{role}[\text{sID}^*_{\text{init}}] = \text{"initiator"}$ 22 if $\text{sID}^*_{\text{init}} \neq s'_{\text{init}}$ return 0 23 return b'
---	--

Figure 23: Unbounded quantum adversaries $C^b = (C_1^b, C_2^b)$ and $D^b = (D_1^b, D_2^b)$ for $b \in \mathbb{F}_2$, executed in game $\text{GDPB}_{\delta(pk, sk)}$ with access to $|F\rangle$, for case $(\neg\text{st})$ of the proof of Lemma 5. $\delta(pk, sk)$ is defined in Equation (5). 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 $G_{6,b}^{\neg\text{st}}$, see Figure 22, while for adversaries D^b , DER_{resp} and DER_{init} and H are defined as in game $G_{9,b}^{\neg\text{st}}$ (also Figure 22).

necessary that the same ephemeral public key \tilde{pk}^* was drawn by $\text{INIT}(s)$. But recall that since game $G_{6,b}^{\neg\text{st}}$, 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 random, B 's view does not change and

$$\Pr[G_{7,b}^{\neg\text{st}} \Rightarrow 1] = \Pr[G_{8,b}^{\neg\text{st}} \Rightarrow 1] .$$

We can now get rid of \tilde{sk}^* altogether by changing DER_{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 not 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, DER_{resp} is changed accordingly. $\text{GAMES } G_{9,b}^{\neg\text{st}}$. In game $G_{9,b}^{\neg\text{st}}$, random oracle H is changed as follows: Instead of picking H uniformly random, we pick two random oracles H_q and H' and define

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

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

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

It remains to show that the keys derived by DER_{init} are still consistent. Since we enforced in game $G_{6,b}^{\neg\text{st}}$ 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 $\text{DER}_{\text{init}}(s'_{\text{init}})$, the change is only conceptual as well: Let (c_i^*, \tilde{c}^*) denote the message on which $\text{DER}_{\text{init}}(s'_{\text{init}})$ 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 $\tilde{c}^* = \text{Enc}(\tilde{p}k^*, \tilde{m}^*; \mathbf{G}(\tilde{m}^*))$. Since \mathbf{G} only samples from good randomness, it holds that $\tilde{m}' := \text{Dec}(\tilde{p}k^*, \tilde{c}^*) = \tilde{m}^*$, and reencryption works as well. Therefore, game $G_{9,b}^{\neg\text{st}}$ should return $H(m'_i, m_j, \tilde{m}', i, j, M = (\tilde{p}k^*, c_j), M')$, as does $G_{8,b}^{\neg\text{st}}$. Since $\tilde{m}' = \tilde{m}^*$ and hence $\text{Enc}(\tilde{p}k^*, m'; \mathbf{G}(m')) = \tilde{c}^*$, this is indeed the case:

$$\begin{aligned} H(m'_i, m_j, \tilde{m}', i, j, M = (\tilde{p}k^*, c_j), M') &= H_q(m'_i, m_j, \text{Enc}(\tilde{p}k^*, \tilde{m}'; \mathbf{G}(\tilde{m}')), i, j, M, M') \\ &= H_q(m'_i, m_j, \tilde{c}^*, i, j, M, M') . \end{aligned}$$

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

$$\Pr[G_{8,b}^{\neg\text{st}} \Rightarrow 1] = \Pr[G_{9,b}^{\neg\text{st}} \Rightarrow 1] .$$

GAMES $G_{10,b}^{\neg\text{st}}$. In games $G_{10,b}^{\neg\text{st}}$, we switch back to using $\mathbf{G} \leftarrow_{\S} \mathcal{R}^{\mathcal{M}}$ instead of $\mathbf{G}_{\tilde{p}k^*, \tilde{s}k^*}$. With the same reasoning as for the gamehop from game $G_{6,b}^{\neg\text{st}}$ to $G_{7,b}^{\neg\text{st}}$,

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

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

So far, we established

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

hence

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

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

GAMES $G_{11,b}^{\neg\text{st}}$. In game $G_{11,b}^{\neg\text{st}}$, 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 $\text{SID}_{\text{resp}}^*$. Again,

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

GAMES $G_{12,b}^{\neg\text{st}}$. In game $G_{12,b}^{\neg\text{st}}$, DER_{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 $\tilde{c} \leftarrow \overline{\text{Enc}}(\tilde{p}k^*)$. Consider the adversaries $\text{A}_{\text{DS},b}^{\neg\text{st}}$ against the disjoint simulatability of $\text{T}[\text{PKE}, \mathbf{G}]$ given in Figure 25. Each adversary $\text{A}_{\text{DS},b}^{\neg\text{st}}$ 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 H_q^1 to H_q^3 and H' (at most $q_{\mathbf{H}} + S$ times) as well as \mathbf{G} (at most $q_{\mathbf{G}} + q_{\mathbf{H}} + 3S$ times), and to lazy sample (at most S times). Hence the total running time is upper bounded as follows:

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

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

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

<p>GAMES $G_{10,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>01 cnt, sID* := 0</p> <p>02 $s'_{\text{init}} \leftarrow_{\\$} [S]$</p> <p>03 $s'_{\text{resp}} \leftarrow_{\\$} [S]$ // $G_{11,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>04 for $n \in [N]$</p> <p>05 $(pk_n, sk_n) \leftarrow \text{KG}$</p> <p>06 $(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \text{KG}$</p> <p>07 $b' \leftarrow \text{B}^{\text{O}, \text{G}, \text{H}}((pk_n)_{n \in [N]})$</p> <p>08 if ATTACK(sID*)</p> <p>09 return 0</p> <p>10 if $\mathfrak{M}(\text{sID}^*) \neq 1$ ABORT</p> <p>11 if revState[sID*_{init}] ABORT</p> <p>12 Pick sID*_{init} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.</p> <p> role[sID*_{init}] = "initiator"</p> <p>13 if sID*_{init} $\neq s'_{\text{init}}$ return 0</p> <p>14 Pick sID*_{resp} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.</p> <p> role[sID*_{resp}] = "responder" // $G_{11,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>15 if sID*_{resp} $\neq s'_{\text{resp}}$</p> <p>16 return 0 // $G_{11,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>17 return b'</p>	<p>DER_{resp}(sID, $M = (\tilde{pk}, c_j)$)</p> <p>18 if holder[sID] = \perp or sKey[sID] $\neq \perp$</p> <p> or role[sID] = "initiator" return \perp</p> <p>19 role[sID] := "responder"</p> <p>20 $(j, i) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$</p> <p>21 $m_i, \tilde{m} \leftarrow_{\\$} \mathcal{M}$</p> <p>22 $c_i := \text{Enc}(pk_i, m_i; \text{G}(m_i))$</p> <p>23 $\tilde{c} := \text{Enc}(\tilde{pk}, \tilde{m}; \text{G}(\tilde{m}))$</p> <p>24 if sID = s'_{resp}</p> <p>25 $\tilde{c} \leftarrow \overline{\text{Enc}}(\tilde{pk}^*)$ // $G_{12,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>26 if $\tilde{c} \in \text{Enc}(\tilde{pk}, \mathcal{M}; \mathcal{R})$</p> <p>27 ABORT // $G_{13,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$</p> <p>28 $M' := (c_i, \tilde{c})$</p> <p>29 $m'_j := \text{Dec}(sk_j, c_j)$</p> <p>30 if $m'_j = \perp$ or $c_j \neq \text{Enc}(pk_j, m'_j; \text{G}(m'_j))$</p> <p>31 $K' := \text{H}'_R(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$</p> <p>32 else</p> <p>33 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$</p> <p>34 if $\tilde{pk} = \tilde{pk}^*$</p> <p>35 $K' := \text{H}_q(m_i, m'_j, \tilde{c}, i, j, M, M')$</p> <p>36 sKey[sID] := K'</p> <p>37 (received[sID], sent[sID]) := (M, M')</p> <p>38 return M'</p> <p>TEST(sID) //only one query</p> <p>39 sID* := sID</p> <p>40 if sKey[sID*] = \perp return \perp</p> <p>41 $K_0^* := \text{sKey}[\text{sID}^*]$ // $G_{10,b}^{\neg\text{st}} - G_{12,b}^{\neg\text{st}}$</p> <p>42 $K_0^* \leftarrow_{\\$} \mathcal{K}$ // $G_{14,0}^{\neg\text{st}}$</p> <p>43 $K_1^* \leftarrow_{\\$} \mathcal{K}$</p> <p>44 return K_b^*</p>
---	--

Figure 24: Games $G_{10,b}^{\neg\text{st}} - G_{14,b}^{\neg\text{st}}$ for case ($\neg\text{st}$) of the proof of Lemma 5. All oracles except for TEST and DER_{resp} remain as in game $G_{10,b}^{\neg\text{st}}$ (see Figure 22, page 43).

Folding adversaries $A_{\text{DS},0}^{\neg\text{st}}$ and $A_{\text{DS},1}^{\neg\text{st}}$ into one adversary $A_{\text{DS}}^{\neg\text{st}}$ yields

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

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

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

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

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

It remains to upper bound $|\Pr[G_{14,0}^{\neg\text{st}} \Rightarrow 1] - \Pr[G_{13,0}^{\neg\text{st}} \Rightarrow 1]|$. B cannot distinguish the value $K_0^* = \text{sKey}[\text{sID}^*]$ that is returned by TEST(sID*) from random in game $G_{13,0}^{\neg\text{st}}$ 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.

$\mathbf{A}_{\text{DS},b}^{-\text{st}}(\mathbf{H}'\rangle, \mathbf{H}_q\rangle, \mathbf{G}\rangle)(\tilde{pk}^*, c^*)$ <pre style="font-family: monospace; font-size: 0.9em; margin: 0;"> 01 cnt, sID* := 0 02 s'_init ←_{\\$} [S], s'_resp ←_{\\$} [S] 03 for n ∈ [N] 04 (pk_n, sk_n) ← KG 05 b' ← B^{O, G⟩, H⟩}((pk_n)_{n∈[N]}) 06 if ATTACK(sID*) return 0 07 if M(sID*) ≠ 1 ABORT 08 if revState[sID*_init] ABORT 09 Pick sID*_init ∈ {sID*, sID'} s. th. role[sID*_init] = "initiator" 10 if sID*_init ≠ s'_init return 0 11 Pick sID*_resp ∈ {sID*, sID'} s. th. role[sID*_resp] = "responder" 12 if sID*_resp ≠ s'_resp return 0 13 return b' REV-STATE(sID ≠ s'_init) 14 if state[sID] = ⊥ return ⊥ 15 revState[sID] := true 16 return state[sID]</pre>	$\text{DER}_{\text{resp}}(\text{sID}, M = (\tilde{pk}, c_j))$ <pre style="font-family: monospace; font-size: 0.9em; margin: 0;"> 17 if holder[sID] = ⊥ or sKey[sID] ≠ ⊥ or role[sID] = "initiator" return ⊥ 18 role[sID] := "responder" 19 (j, i) := (holder[sID], peer[sID]) 20 m_i, m̃ ←_{\\$} M 21 c_i := Enc(pk_i, m_i; G(m_i)) 22 c̃ := Enc(ṽk, m̃; G(m̃)) 23 if sID = s'_resp 24 c̃ := c* 25 M' := (c_i, c̃) 26 m'_j := Dec(sk_j, c_j) 27 if m'_j = ⊥ or c_j ≠ Enc(pk_j, m'_j; G(m'_j)) 28 K' := H'_R(m_i, c_j, m̃, i, j, c_i, M, M') 29 else 30 K' := H(m_i, m'_j, m̃, i, j, M, M') 31 if ṽk = ṽk* 32 K' := H_q(m_i, m'_j, c̃, i, j, M, M') 33 sKey[sID] := K' 34 (received[sID], sent[sID]) := (M, M') 35 return M'</pre>
--	--

Figure 25: Adversaries $\mathbf{A}_{\text{DS},b}^{-\text{st}}$ for case $(\neg\text{st})$ of the proof of Lemma 5, with oracle access to $|\mathbf{H}'\rangle$, $|\mathbf{H}_q\rangle$ and $|\mathbf{G}\rangle$. All oracles except for DER_{resp} and REV-STATE are defined as in game $G_{11,b}^{-\text{st}}$ (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 $\text{INIT}(\text{sID}_{\text{init}}^*)$ (line 25, also Figure 21). Let m_j^* denote the randomly chosen message with encryption $c_j^* := \text{Enc}(pk_{j^*}, m_j^*; \mathbf{G}(m_j^*))$ that was sampled during execution of $\text{INIT}(\text{sID}_{\text{init}}^*)$, furthermore let \tilde{c}^* denote the fake ciphertext that was sampled under \tilde{pk}^* during execution of $\text{Der}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$ (Figure 24, line 25) and let m_i^* denote the randomly chosen message with encryption $c_i^* := \text{Enc}(pk_{i^*}, m_i^*; \mathbf{G}(m_i^*))$ that was picked during execution of $\text{DER}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$. We changed the key derivation such that since \tilde{pk}^* is used, in the case that sID^* is an "initiator" session, we have

$$K_0^* = \begin{cases} \text{H}'_{\text{L1}}(c_i^*, m_j^*, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c_j^*), (c_i^*, \tilde{c}^*)) & m'_i = \perp \text{ or } \text{Enc}(pk_{i^*}, m'_i) \neq c_i^* \\ \text{H}_q(m'_i, m_j^*, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c_j^*), (c_i^*, \tilde{c}^*)) & \text{o.w.} \end{cases},$$

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

$$K_0^* = \begin{cases} \text{H}'_{\text{R}}(m_i^*, c_j^*, \tilde{m}^*, i^*, j^*, (\tilde{pk}^*, c_j^*), (c_i^*, \tilde{c}^*)) & m'_j = \perp \text{ or } \text{Enc}(pk_{j^*}, m'_j) \neq c_j^* \\ \text{H}_q(m_i^*, m_j^*, \tilde{c}^*, i^*, j^*, (\tilde{pk}^*, c_j^*), (c_i^*, \tilde{c}^*)) & \text{o.w.} \end{cases},$$

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

First, we will argue that \mathbf{B} could not possibly obtain K_0^* by a query to REVEAL : Recall that \mathbf{B} trivially loses if it revealed the test session or its match. Hence, \mathbf{B} would have to create some session $\text{sID} \notin \{\text{sID}_{\text{init}}^*, \text{sID}_{\text{resp}}^*\}$ that derives the same key as the test session. First, we argue that the key cannot be obtained via any session sID with $\text{role}[\text{sID}] \neq \text{role}[\text{sID}^*]$: 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 $\text{role}[\text{sID}] = \text{role}[\text{sID}^*]$.

We first consider the case that sID^* is an "initiator" session: To obtain K_0^* via another "initiator" session, \mathbf{B} would have to establish and initialise another "initiator" session $\text{sID} \neq \text{sID}_{\text{init}}^*$ with holder i^* and peer j^* . The subsequent call to DER_{init} could only result in the same key if $\text{INIT}(\text{sID})$ had also picked ephemeral key \tilde{pk}^* , which is impossible since we enforced in game $G_{6,b}^{-\text{st}}$ 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 , \mathbf{B} would have to call DER_{resp} for some session $\text{sID} \neq \text{sID}_{\text{resp}}^*$ 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 $\text{Enc}(\tilde{pk}, -, -)$, 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 $|H\rangle$ could contain session key $K_0^* = H_q(m_i^*, m_j^*, \tilde{c}^*, i^*, j^*, M, M')$, recall that for any message $M = (\tilde{pk}^*, c_j)$, where c_j is arbitrary,

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

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

In conclusion,

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

hence

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

Collecting the probabilities yields

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

the upper bound we claimed in equation (3).

A.2 Case $(\neg sk)$ of the Proof of Lemma 5

CASE $(\neg sk)$ (INITIALISING SESSION'S OWNER WAS NOT CORRUPTED). Intuition is as follows: While \mathbf{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 $\neg\text{corrupted}[\text{holder}[\text{sID}_{\text{init}}^*]]$. Afterwards, we ensure that the game makes no use of sk_{i^*} any longer by patching encryption into the random oracle (in games $G_{2,b}^{\neg sk}$ to $G_{9,b}^{\neg sk}$, 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 $c_i = \text{Enc}(pk_{i^*}, m_i^*)$ with a fake ciphertext that gets sampled using $\overline{\text{Enc}}$ (games $G_{10,b}^{\neg sk}$ to $G_{13,b}^{\neg sk}$, see Figure 29). We show that after those changes, \mathbf{B} 's view does not change with overwhelming probability if we finally change TEST such that it always returns a random value (game $G_{13,b}^{\neg sk}$, also Figure 29).

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

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

GAMES $G_{2,b}^{\neg sk} - G_{4,b}^{\neg sk}$	
01 cnt, sID* := 0	
02 $i' \leftarrow_{\$} [N]$	$// G_{4,b}^{\neg sk}$
03 for $n \in [N]$	
04 $(pk_n, sk_n) \leftarrow \text{KG}$	
05 $b' \leftarrow \text{B}^{O, G , H }((pk_n)_{n \in [N]})$	
06 if ATTACK(sID*)	
07 return 0	
08 if $ \mathcal{M}(\text{sID}^*) \neq 1$ ABORT	
09 Pick $\text{sID}_{\text{init}}^* \in \{\text{sID}^*, \text{sID}'\}$	
s. th. $\text{role}[\text{sID}_{\text{init}}^*] = \text{"initiator"}$	$// G_{3,b}^{\neg sk} - G_{4,b}^{\neg sk}$
10 if corrupted[holder[sID _{init} [*]]]	
11 ABORT	$// G_{3,b}^{\neg sk} - G_{7,b}^{\neg sk}$
12 if holder[sID _{init} [*]] $\neq i'$	
13 return 0	$// G_{4,b}^{\neg sk} - G_{4,b}^{\neg sk}$
14 return b'	

Figure 26: Games $G_{2,b}^{\neg sk} - G_{4,b}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5. Helper procedure ATTACK and oracles TEST, Init, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 17 and Figure 18, pages 28 and 29).

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

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

Analogous to our proof of case $(\neg st)$, the first goal is not to have to make use of the initialiser'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 $G_{pk_{i'}, sk_{i'}}$ that only samples randomness under which decryption never fails, thereby allowing for a reduction to game GDPB.

GAMES $G_{4,b}^{\neg sk}$. In both games $G_{4,b}^{\neg sk}$, 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 $G_{4,b}^{\neg sk}$ and $G_{3,b}^{\neg sk}$ proceed identically if $\text{holder}[\text{sID}_{\text{init}}^*] = i'$, and since games $G_{4,b}^{\neg sk}$ output 0 if $\text{holder}[\text{sID}_{\text{init}}^*] \neq i'$, we have that $\Pr[G_{3,b}^{\neg sk} \Rightarrow 1] = N \cdot \Pr[G_{4,b}^{\neg sk} \Rightarrow 1]$ and hence,

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

GAMES $G_{4,b}^{\neg sk} - G_{9,b}^{\neg sk}$		DER _{resp} (sID, $M = (\tilde{pk}, c_j)$)
01 Pick 2 q -wise hash f	$\parallel G_{5,b}^{\neg sk} - G_{8,b}^{\neg sk}$	46 if holder[sID] = \perp or sKey[sID] $\neq \perp$
02 cnt, sID* := 0		or role[sID] = "initiator" return \perp
03 $i' \leftarrow_{\$} [N]$		47 role[sID] := "responder"
04 for $n \in [N]$		48 $(j, i) := (\text{holder[sID]}, \text{peer[sID]})$
05 $(pk_n, sk_n) \leftarrow \text{KG}$		49 $m_i, \tilde{m} \leftarrow_{\$} \mathcal{M}$
06 $\mathbf{G} := \mathbf{G}_{pk_{i'}, sk_{i'}}$	$\parallel G_{5,b}^{\neg sk} - G_{8,b}^{\neg sk}$	50 $c_i := \text{Enc}(pk_i, m_i; \mathbf{G}(m_i))$
07 $b' \leftarrow \mathbf{B}^{\text{O}, \{G, H\}}((pk_n)_{n \in [N]})$		51 $\tilde{c} := \text{Enc}(pk, \tilde{m}; \mathbf{G}(\tilde{m}))$
08 if ATTACK(sID*)		52 $M' := (c_i, \tilde{c})$
09 return 0		53 $m'_j := \text{Dec}(sk_j, c_j)$
10 if $ \mathfrak{M}(\text{sID}^*) \neq 1$ ABORT		54 if $m'_j = \perp$
11 Pick sID* _{init} $\in \{\text{sID}^*, \text{sID}'\}$		or $c_j \neq \text{Enc}(pk_j, m'_j; \mathbf{G}(m'_j))$
s. th. role[sID* _{init}] = "initiator"		55 $K' := \text{H}_R^1(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
12 if corrupted[holder[sID* _{init}]]		56 if $j = i'$ and $i \neq i'$
13 ABORT		57 $K' := \text{H}_q^2(m_i, c_j, \tilde{m}, i, i', M, M')$
14 if holder[sID* _{init}] $\neq i'$		58 if $i = j = i'$
15 return 0		59 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}, i', i', M, M')$
16 return b'		60 else
		61 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
DER _{init} (sID, $M' = (c_i, \tilde{c})$)		62 if $i = i'$ and $j \neq i'$
17 if holder[sID] = \perp or state[sID] = \perp		63 $K' := \text{H}_q^1(c_i, m'_j, \tilde{m}, i', j, M, M')$
or sKey[sID] $\neq \perp$ return \perp		64 if $j = i'$ and $i \neq i'$
18 $(i, j) := (\text{holder[sID]}, \text{peer[sID]})$		65 $K' := \text{H}_q^2(m_i, c_j, \tilde{m}, i, i', M, M')$
19 $(sk, m_j, M := (pk, c_j)) := \text{state[sID]}$		66 if $i = j = i'$
20 $m'_i := \text{Dec}(sk_i, c_i)$		67 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}, i', i', M, M')$
21 $\tilde{m}' := \text{Dec}(sk, \tilde{c})$		68 sKey[sID] := K'
22 if $m'_i = \perp$ or $c_i \neq \text{Enc}(pk_i, m'_i; \mathbf{G}(m'_i))$		69 (received[sID], sent[sID]) := (M, M')
23 if $\tilde{m}' = \perp$		70 return M'
24 $K := \text{H}_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$		
25 else		$\mathbf{G}_{pk_{i'}, sk_{i'}}(m)$
26 $K := \text{H}_{L2}(c_i, m_j, \tilde{m}', i, j, M, M')$		71 $r := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk_{i'}, sk_{i'}, m); f(m))$
27 if $i = i'$ and $j \neq i'$		72 return r
28 $K' := \text{H}_q^1(c_i, m_j, \tilde{m}', i', j, M, M')$	$\parallel G_{8,b}^{\neg sk} -$	
$G_{9,b}^{\neg sk}$		$\text{H}(m_1, m_2, m_3, i, j, M, M')$
29 if $i = j = i'$		73 if $i = i'$ and $j \neq i'$
$G_{9,b}^{\neg sk}$		74 return $\text{H}_q^1(\text{Enc}(pk_{i'}, m_1; \mathbf{G}(m_1)), m_2, m_3, i, j, M, M')$
30 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}', i', i', M, M')$	$\parallel G_{8,b}^{\neg sk} -$	75 if $j = i'$ and $i \neq i'$
$G_{9,b}^{\neg sk}$		76 return $\text{H}_q^2(m_1, \text{Enc}(pk_{i'}, m_2; \mathbf{G}(m_2)), m_3, i, j, M, M')$
31 else		77 if $i = j = i'$
32 if $\tilde{m}' = \perp$		78 return $\text{H}_q^3(\text{Enc}(pk_{i'}, m_1; \mathbf{G}(m_1)), \text{Enc}(pk_{i'}, m_2; \mathbf{G}(m_2)), m_3, i, j, M, M')$
33 $K := \text{H}_{L3}(m'_i, m_j, \tilde{c}, i, j, M, M')$		79 return $\text{H}(m_1, m_2, m_3, i, j, M, M')$
34 if $i = i'$		
35 $K := \text{H}_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$	$\parallel G_{6,b}^{\neg sk} -$	
$G_{9,b}^{\neg sk}$		
36 else		
37 $K := \text{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$		
38 if $i = i'$ and $j \neq i'$		
39 $K' := \text{H}_q^1(c_i, m_j, \tilde{m}', i', j, M, M')$	$\parallel G_{7,b}^{\neg sk} -$	
$G_{9,b}^{\neg sk}$		
40 if $j = i'$ and $i \neq i'$		
41 $K' := \text{H}_q^2(m'_i, c_j, \tilde{m}', i, i', M, M')$	$\parallel G_{7,b}^{\neg sk} -$	
$G_{9,b}^{\neg sk}$		
42 if $i = j = i'$		
43 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}', i', i', M, M')$	$\parallel G_{7,b}^{\neg sk} -$	
$G_{9,b}^{\neg sk}$		
44 sKey[sID] := K		
45 received[sID] := M'		

Figure 27: Games $G_{4,b}^{\neg sk} - G_{9,b}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5. Helper procedure ATTACK and oracles TEST, INIT, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 17 and Figure 18, pages 28 and 29).

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

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

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

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

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

denoting the fraction of bad randomness, and $\delta(pk, sk) := \max_{m \in \mathcal{M}} \delta(pk, sk, m)$. (As in the proof of $(\neg\text{st})$.) Recall that with this notation, $\delta = \mathbf{E}[\max_{m \in \mathcal{M}} \delta(pk, sk, m)]$, where the expectation is taken over $(pk, sk) \leftarrow \text{KG}$. To upper bound $|\Pr[G_{4,b}^{\neg sk} \Rightarrow 1] - \Pr[G_{5,b}^{\neg sk} \Rightarrow 1]|$ for each bit b , we construct (unbounded, quantum) adversaries \mathbf{C}^b against the generic distinguishing problem with bounded probabilities GDPB_λ in Figure 28, issuing at most $q_G + 3S$ queries to $|F\rangle$. With the same analysis as in our proof for case $(\neg\text{st})$ (see page 44),

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

and according to Lemma 1,

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

$\mathbf{C}_1^b = \mathbf{D}_1^b$ 01 $(pk, sk) \leftarrow \text{KG}$ 02 for $m \in \mathcal{M}$ 03 $\lambda(m) := \delta(pk, sk, m)$ 04 return $(\lambda(m))_{m \in \mathcal{M}}$ $\mathbf{G}(m)$ 05 if $F(m) = 0$ 06 $\mathbf{G}(m) := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 07 else 08 $\mathbf{G}(m) := \text{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$ 09 return $\mathbf{G}(m)$	$\mathbf{C}_2^{b(F)}, \mathbf{D}_2^{b(F)}$ 10 $\text{cnt}, \text{sID}^* := 0$ 11 $i' \leftarrow_{\mathcal{S}} [N]$ 12 for $n \in [N] \setminus \{i'\}$ 13 $(pk_n, sk_n) \leftarrow \text{KG}$ 14 $(pk_{i'}, sk_{i'}) := (pk, sk)$ 15 $b' \leftarrow \mathbf{B}^{O, G\rangle, H\rangle}((pk_n)_{n \in [N]})$ 16 if $\text{ATTACK}(\text{sID}^*)$ 17 return 0 18 if $ \mathfrak{M}(\text{sID}^*) \neq 1$ ABORT 19 Pick $\text{sID}_{\text{init}}^* \in \{\text{sID}^*, \text{sID}'\}$ s. th. $\text{role}[\text{sID}_{\text{init}}^*] = \text{"initiator"}$ 20 if $\text{corrupted}[\text{holder}[\text{sID}_{\text{init}}^*]]$ ABORT 21 if $\text{holder}[\text{sID}_{\text{init}}^*] \neq i'$ 22 return 0 23 return b'
---	--

Figure 28: Unbounded quantum adversaries \mathbf{C}^b and \mathbf{D}^b , executed in game $\text{GDPB}_{\delta(pk, sk)}$, for case $(\neg sk)$ of the proof of Lemma 5. The adversaries only differ in their definition of DER_{resp} , DER_{init} and \mathbf{H} : For adversaries \mathbf{C}^b , DER_{resp} , DER_{init} and \mathbf{H} are defined as in game $G_{4,b}^{\neg sk}$, see Figure 27, while for adversaries \mathbf{D}^b , DER_{resp} and DER_{init} and \mathbf{H} are defined as in game $G_{5,b}^{\neg sk}$ (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 DER_{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 $G_{6,b}^{\neg sk}$. In both games $G_{6,b}^{\neg sk}$, we change oracle DER_{init} in line 35 for session holder i' as follows: Whenever \tilde{c} does not decrypt to a message \tilde{m}' s. th. $\tilde{c} = \text{Enc}(pk, \tilde{m}', \mathbf{G}(\tilde{m}'))$, the session key is defined as $K := \mathbf{H}'_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$. (Before this change we let $K := \mathbf{H}'_{L3}(m'_i, m_j, \tilde{c}, i, j, M, M')$ in the case that \tilde{c} fails to decrypt, but c_i decrypts correctly.)

We claim that this change does not affect \mathbf{B} 's view since the cases are logically distinct: Since both \mathbf{H}'_{L1} and \mathbf{H}'_{L2} are internal random oracles that cannot be accessed directly by \mathbf{B} , the only way to obtain oracle values of \mathbf{H}'_{L1} and \mathbf{H}'_{L2} is via calls to REVEAL and TEST after execution of DER_{init} . Intuitively, \mathbf{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 $G_{5,b}^{\neg sk}$, but the same key in game $G_{6,b}^{\neg sk}$. In more detail, the following requirements would have to be met:

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

To achieve key equality in game $G_{6,b}^{\neg sk}$ it is required that both $c'_i = c_i$ and $\tilde{c}' = \tilde{c}$, i.e., both $\text{DER}_{\text{init}}(s)$ and $\text{DER}_{\text{init}}(s')$ were called on the same message (c_i, \tilde{c}) . Requiring that $\text{DER}_{\text{init}}(s)$ computes its key as $H'_{L1}(c_i, m_j, \tilde{c}, i', j, M, M')$ in game $G_{5,b}^{\neg sk}$ implies that ciphertext c_i is problematic. But additionally requiring that $\text{DER}_{\text{init}}(s)$ computes its key as $H'_{L3}(\text{Dec}(sk_i, c'_i), m_j, \tilde{c}', i, j, M, M')$ in game $G_{5,b}^{\neg sk}$ 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

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

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 not use the decryptions under $sk_{i'}$, but the ciphertexts themselves. Similar to case (-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 $G_{7,b}^{\neg sk}$. In games $G_{7,b}^{\neg sk}$, the random oracle is changed as follows: Instead of picking H uniformly random, we pick four random oracles H_q^1 to H_q^3 , and H' , and define

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

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 DER_{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

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

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

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

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

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

if both holder and peer are i' .

Likewise, make the change of H explicit in DER_{init} : We have to change DER_{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

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

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

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

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

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

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

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

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 H_q^1 to H_q^3 if a ciphertext fails to de- or reencrypt under $sk_{i'}$, instead of using the implicit reject oracles.

GAMES $G_{8,b}^{-sk}$. In games $G_{8,b}^{-sk}$, we change DER_{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 H_q^2 or H_q^3 instead of rejecting implicitly, just as if c_j had reencrypted correctly. Likewise, we change DER_{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 H_q^1 or H_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 $G_{7,b}^{-sk}$ due to an implicit reject, while in game $G_{8,b}^{-sk}$, 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 $G_{7,b}^{-sk}$, while the key can be linked to random oracle H in game $G_{8,b}^{-sk}$.

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 $\text{INIT}(s)$ returned, and c_j is the encryption that DER_{resp} returned. The conditions above are unsatisfiable since $\mathbf{G}_{pk_{i'}, sk_{i'}}$ only samples good randomness: We have $c_i = \text{Enc}(pk_i, m_i; \mathbf{G}_{pk_{i'}, sk_{i'}}(m_i))$ and $c_j = \text{Enc}(pk_j, m_j; \mathbf{G}_{pk_{i'}, sk_{i'}}(m_j))$. Both $\mathbf{G}_{pk_{i'}, sk_{i'}}(m_i)$ and $\mathbf{G}_{pk_{i'}, sk_{i'}}(m_j)$ 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 $G_{8,b}^{\neg sk}$ in any other case, a loss of mismatching keys is impossible.

The latter, i.e., linking a key to \mathbf{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 $c_i = \text{Enc}(pk_i, m_i; \mathbf{G}_{pk_{i'}, sk_{i'}}(m_i))$: 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 $c_i = \text{Enc}(pk_i, m_i; \mathbf{G}_{pk_{i'}, sk_{i'}}(m_i))$: In this case, while the derived key would be defined as $\mathbf{H}_q^1(c_i, m_j, \tilde{m}', i', j, M, M')$ or $\mathbf{H}_q^3(c_i, c_j, \tilde{m}', i', i', M, M')$, respectively, it could not possibly correlate to any random oracle query to $|\mathbf{H}|$: Since $i = i'$, $|\mathbf{H}|$ 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 \mathbf{H} . The argument is completely symmetrical, hence a link of keys can also not happen if the session is a "responder" session.

In conclusion, \mathbf{B} 's view is identical in both games and

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

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

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

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

So far, we established

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

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

GAMES $G_{10,b}^{\neg sk}$. Games $G_{10,b}^{\neg sk}$ do not differ from games $G_{9,b}^{\neg sk}$. 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 $G_{9,b}^{\neg sk}$.

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

We stress that from game $G_{9,b}^{\neg sk}$ 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 \mathbf{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 $G_{11,b}^{\neg sk}$. In games $G_{11,b}^{\neg sk}$, one of the sessions that get established during execution of \mathbf{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 $G_{11,b}^{\neg sk}$ and $G_{10,b}^{\neg sk}$ proceed identically unless $s'_{\text{resp}} \neq \text{sID}_{\text{resp}}^*$, and since games $G_{11,b}^{\neg sk}$ output 0 if $s'_{\text{resp}} \neq \text{sID}_{\text{resp}}^*$,

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

GAMES $G_{12,b}^{\neg sk}$. In games $G_{12,b}^{\neg sk}$, 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 $c_i := \text{Enc}(pk_{i'}, m_i; \mathbf{G}(m_i))$ for some randomly drawn message m_i , but a fake

GAMES $G_{10,b}^{\neg sk} - G_{13,b}^{\neg sk}$	DER _{resp} (sID, $M = (\tilde{pk}, c_j)$)
01 cnt, sID* := 0	47 if holder[sID] = \perp or sKey[sID] $\neq \perp$
02 $i' \leftarrow_{\mathcal{S}} [N]$	or role[sID] = "initiator"
03 $s'_{\text{resp}} \leftarrow_{\mathcal{S}} [S]$	return \perp
04 for $n \in [N]$	48 role[sID] := "responder"
05 $(pk_n, sk_n) \leftarrow \text{KG}$	49 $(j, i) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
06 $b' \leftarrow \mathbf{B}^{O, G , H }((pk_n)_{n \in [N]})$	50 $m_i, \tilde{m} \leftarrow_{\mathcal{S}} \mathcal{M}$
07 if ATTACK(sID*)	51 $c_i := \text{Enc}(pk_i, m_i; \mathbf{G}(m_i))$
08 return 0	52 if sID = s'_{resp}
09 if $ \mathfrak{M}(\text{sID}^*) \neq 1$ ABORT	53 $c_i \leftarrow \overline{\text{Enc}}(pk_{i'}, i')$ // $G_{12,b}^{\neg sk} - G_{13,b}^{\neg sk}$
10 Pick sID* _{init} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.	54 if $c_i \in \text{Enc}(pk_{i'}, \mathcal{M}; \mathcal{R})$
role[sID* _{init}] = "initiator"	55 ABORT // $G_{13,b}^{\neg sk} - G_{13,b}^{\neg sk}$
11 if corrupted[holder[sID* _{init}]] ABORT	56 $\tilde{c} := \text{Enc}(\tilde{pk}, \tilde{m}; \mathbf{G}(\tilde{m}))$
12 Pick sID* _{resp} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.	57 $M' := (c_i, \tilde{c})$
role[sID* _{resp}] = "responder" // $G_{11,b}^{\neg sk} - G_{13,b}^{\neg sk}$	58 if $j = i'$
13 if holder[sID* _{init}] $\neq i'$	59 if $i = i'$
14 return 0	60 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}, i', i', M, M')$
15 if sID* _{resp} $\neq s'_{\text{resp}}$	61 else
16 return 0 // $G_{11,b}^{\neg sk} - G_{13,b}^{\neg sk}$	62 $K' := \text{H}_q^2(m_i, c_j, \tilde{m}, i, i', M, M')$
17 return b'	63 else
DER _{init} (sID, $M' = (c_i, \tilde{c})$)	64 $m'_j := \text{Dec}(sk_j, c_j)$
18 if holder[sID] = \perp or state[sID] = \perp	65 if $m'_j = \perp$
or sKey[sID] $\neq \perp$ return \perp	or $c_j \neq \text{Enc}(pk_j, m'_j; \mathbf{G}(m'_j))$
19 $(i, j) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$	66 $K' := \text{H}_q^1(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
20 $(sk, m_j, M := (pk, c_j)) := \text{state}[\text{sID}]$	67 else
21 $\tilde{m}' := \text{Dec}(sk, \tilde{c})$	68 if $i' = i$
22 if $i = i'$	69 $K' := \text{H}_q^1(c_i, m'_j, \tilde{m}, i', j, M, M')$
23 if $\tilde{m}' = \perp$	70 else
24 $K := \text{H}_{L1}^1(c_i, m_j, \tilde{c}, i, j, M, M')$	71 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
25 else	72 sKey[sID] := K'
26 if $j = i'$	73 (received[sID], sent[sID]) := (M, M')
27 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}', i', i', M, M')$	74 return M'
28 else	TEST(sID) // only one query
29 $K' := \text{H}_q^1(c_i, m_j, \tilde{m}', i', j, M, M')$	75 sID* := sID
30 else	76 if sKey[sID*] = \perp
31 $m'_i := \text{Dec}(sk_i, c_i)$	77 return \perp
32 if $m'_i = \perp$ or $c_i \neq \text{Enc}(pk_i, m'_i; \mathbf{G}(m'_i))$	78 $K_0^* := \text{sKey}[\text{sID}^*]$ // $G_{10,b}^{\neg sk} - G_{13,b}^{\neg sk}$
33 if $\tilde{m}' = \perp$	79 $K_0^* \leftarrow_{\mathcal{S}} \mathcal{K}$ // $G_{13,0}^{\neg sk}$
34 $K := \text{H}_{L1}^1(c_i, m_j, \tilde{c}, i, j, M, M')$	80 $K_1^* \leftarrow_{\mathcal{S}} \mathcal{K}$
35 else	81 return K_b^*
36 $K := \text{H}_{L2}^1(c_i, m_j, \tilde{m}', i, j, M, M')$	
37 else	
38 if $\tilde{m}' = \perp$	
39 $K := \text{H}_{L3}^1(m'_i, m_j, \tilde{c}, i, j, M, M')$	
40 else	
41 if $j = i'$	
42 $K' := \text{H}_q^2(m'_i, c_j, \tilde{m}', i, i', M, M')$	
43 else	
44 $K := \text{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$	
45 sKey[sID] := K	
46 received[sID] := M'	

Figure 29: Games $G_{10,b}^{\neg sk} - G_{13,b}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5. Random oracles G and H remain as in game $G_{10,b}^{\neg sk}$.

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

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

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

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

$$|\Pr[G_{11,b}^{\neg sk} \Rightarrow 1] - \Pr[G_{12,b}^{\neg sk} \Rightarrow 1]| = \text{Adv}_{\text{T}[\text{PKE}, \text{G}]}^{\text{DS}}(A_{\text{DS},b}^{\neg sk})$$

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

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

$A_{\text{DS},b}^{\neg sk, \text{H}'\rangle, \text{H}_q\rangle, \text{G}\rangle}(pk, c^*)$	$\text{DER}_{\text{resp}}(\text{sID}, M = (\tilde{pk}, c_j))$
01 cnt, sID* := 0	19 if holder[sID] = \perp or sKey[sID] $\neq \perp$
02 $i' \leftarrow_{\S} [N]$	or role[sID] = "initiator"
03 $s'_{\text{resp}} \leftarrow_{\S} [S]$	20 return \perp
04 for $n \in [N] \setminus \{i'\}$	21 role[sID] := "responder"
05 $(pk_n, sk_n) \leftarrow \text{KG}$	22 $(j, i) := (\text{holder}[\text{sID}], \text{peer}[\text{sID}])$
06 $(pk_{i'}, sk_{i'}) := (pk, \perp)$	23 $m_i, \tilde{m} \leftarrow_{\S} \mathcal{M}$
07 $b' \leftarrow \mathbf{B}^{\text{O}, \text{G}\rangle, \text{H}\rangle}((pk_n)_{n \in [N]})$	24 $c_i := \text{Enc}(pk_i, m_i; \text{G}(m_i))$
08 if ATTACK(sID*)	25 if sID = s'_{resp}
09 return 0	26 $c_i := c^*$
10 if $ \mathfrak{M}(\text{sID}^*) \neq 1$ ABORT	27 $\tilde{c} := \text{Enc}(\tilde{pk}, \tilde{m}; \text{G}(\tilde{m}))$
11 Pick sID* _{init} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.	28 $M' := (c_i, \tilde{c})$
role[sID* _{init}] = "initiator"	29 if $j = i'$
12 if corrupted[holder[sID* _{init}]] ABORT	30 if $i = i'$
13 Pick sID* _{resp} $\in \{\text{sID}^*, \text{sID}'\}$ s. th.	31 $K' := \text{H}_q^3(c_i, c_j, \tilde{m}, i', i', M, M')$
role[sID* _{resp}] = "responder"	32 else
14 if holder[sID* _{init}] $\neq i'$	33 $K' := \text{H}_q^2(m_i, c_j, \tilde{m}, i, i', M, M')$
15 return 0	34 else
16 if sID* _{resp} $\neq s'_{\text{resp}}$	35 $m'_j := \text{Dec}(sk_j, c_j)$
17 return 0	36 if $m'_j = \perp$
18 return b'	or $c_j \neq \text{Enc}(pk_j, m'_j; \text{G}(m'_j))$
	37 $K' := \text{H}_R(m_i, c_j, \tilde{m}, i, j, c_i, M, M')$
	38 else
	39 if $i' = i$
	40 $K' := \text{H}_q^1(c_i, m'_j, \tilde{m}, i', j, M, M')$
	41 else
	42 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$
	43 sKey[sID] := K'
	44 (received[sID], sent[sID]) := (M, M')
	45 return M'

Figure 30: Adversaries $A_{\text{DS},b}^{\neg sk}$ for case (\neg sk) of the proof of Lemma 5, with oracle access to $|\text{H}'\rangle$, $|\text{H}_q\rangle$ and $|\text{G}\rangle$. All oracles except for DER_{resp} and CORRUPT are defined as in game $G_{11,b}^{\neg sk}$ (see Figure 29).

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

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

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

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

It remains to upper bound $|\Pr[G_{13,0}^{\neg sk} \Rightarrow 1] - \Pr[G_{12,0}^{\neg sk} \Rightarrow 1]|$. B cannot distinguish the value $K_0^* = \text{sKey}[\text{sID}^*]$ that is returned by TEST(sID^*) from random in game $G_{12,0}^{\neg sk}$ 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 $c_j^* := \text{Enc}(pk_{j^*}, m_j^*; \mathbf{G}(m_j^*))$ that was sampled during execution of INIT($\text{sID}_{\text{init}}^*$). Let $(\tilde{p}k^*, \tilde{s}k^*)$ 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 $\text{Der}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$ (Figure 29, line 53) and let \tilde{m}^* denote the randomly chosen message with encryption $\tilde{c}^* := \text{Enc}(\tilde{p}k^*, \tilde{m}^*; \mathbf{G}(\tilde{m}^*))$ that was picked during execution of $\text{DER}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$.

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

$$K_0^* = \begin{cases} \text{H}_q^3(c_i^*, c_j^*, \tilde{m}^*, i', i', (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & j^* = i' \\ \text{H}_R(m_i^*, c_j^*, \tilde{m}^*, i', j^*, (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & j^* \neq i' \text{ and } (m_j' = \perp \text{ or } \text{Enc}(pk_{j^*}, m_j') \neq c_j^*) \\ \text{H}_q^1(c_i^*, m_j', \tilde{m}^*, i', j^*, (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & \text{o.w.} \end{cases} ,$$

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

$$K_0^* = \begin{cases} \text{H}'_{L1}(c_i^*, m_j^*, \tilde{c}^*, \tilde{p}k^*, i', j^*, (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & \tilde{m}' = \perp \text{ or } \tilde{c}^* \neq \text{Enc}(\tilde{p}k^*, \tilde{m}'; \mathbf{G}(\tilde{m}')) \\ \text{H}_q^3(c_i^*, c_j^*, \tilde{m}', i', i', (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & \tilde{c}^* = \text{Enc}(\tilde{p}k^*, \tilde{m}'; \mathbf{G}(\tilde{m}')) \text{ and } j^* = i' \\ \text{H}_q^1(c_i^*, m_j^*, \tilde{m}', i', j^*, (\tilde{p}k^*, c_j^*), (c_i^*, \tilde{c}^*)) & \text{o.w.} \end{cases} ,$$

where $m_i' := \text{Dec}(sk_{i'}, c_i^*)$ and $\tilde{m}' := \text{Dec}(\tilde{s}k^*, \tilde{c}^*)$.

With an argument similar to case ($\neg st$), we can show that none of the quantum answers of $|H\rangle$ could contain the session key: In any of the cases, to trigger a query to $|H_q\rangle$ such that its answer contains K_0 , B would need to come up with a message m such that $\text{Enc}(pk_{i'}, m; \mathbf{G}(m)) = c_i^*$. Since we abort if c_i^* lies in the range of $\text{Enc}(pk_{i'}, -; -)$, 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 initialise 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 $\tilde{p}k^*$ 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 $(S-2) \cdot \mu(\text{KG}) \cdot \mu(\text{Enc})$. 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 $\text{Enc}(pk_{i'}, -; -)$, this is impossible. Hence, we can upper bound the probability of recreation, and therefore, the game distance, by

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

Collecting the probabilities, we obtain

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

the upper bound we claimed in equation (4).

B Proof of Lemma 6

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

<p>GAMES $G_{0,b} - G_{1,b}$</p> <pre> 01 cnt, sID* := 0 02 for n ∈ [N] 03 (pk_n, sk_n) ← KG 04 b' ← B^{O:(G),H}((pk_n)_{n∈[N]}) 05 if ATTACK(sID*) 06 return 0 07 if $\mathfrak{M}(\text{sID}^*) \neq \emptyset$ ABORT // $G_{1,b}$ 08 return b'</pre> <p>INIT(sID)</p> <pre> 09 if holder[sID] = ⊥ or sent[sID] ≠ ⊥ return ⊥ 10 role[sID] := "initiator" 11 i := holder[sID] 12 j := peer[sID] 13 m_j ←_§ M 14 c_j := Enc(pk_j, m_j; G(m_j)) 15 (p_k, s_k) ← KG 16 M := (p_k, c_j) 17 state[sID] := (s_k, m_j, M) 18 sent[sID] := M 19 return M</pre>	<p>DER_{resp}(sID, M = (p_k, c_j))</p> <pre> 20 if holder[sID] = ⊥ or sKey[sID] ≠ ⊥ or role[sID] = "initiator" return ⊥ 21 role[sID] := "responder" 22 (j, i) := (holder[sID], peer[sID]) 23 m_i, m̃ ←_§ M 24 c_i := Enc(pk_i, m_i; G(m_i)) 25 c̃ := Enc(p_k, m̃; G(m̃)) 26 M' := (c_i, c̃) 27 m'_j := Dec(sk_j, c_j) 28 if m'_j = ⊥ or c_j ≠ Enc(pk_j, m'_j; G(m'_j)) 29 K' := H'_R(m_i, c_j, m̃, i, j, c_i, M, M') 30 else K' := H(m_i, m'_j, m̃, i, j, M, M') 31 sKey[sID] := K' 32 (received[sID], sent[sID]) := (M, M') 33 return M'</pre> <p>DER_{init}(sID, M' = (c_i, c̃))</p> <pre> 34 if holder[sID] = ⊥ or state[sID] = ⊥ or sKey[sID] ≠ ⊥ return ⊥ 35 (i, j) := (holder[sID], peer[sID]) 36 (s_k, m_j, M := (p_k, c_j)) := state[sID] 37 m'_i := Dec(sk_i, c_i) 38 m̃' := Dec(s_k, c̃) 39 if m'_i = ⊥ or c_i ≠ Enc(pk_i, m'_i; G(m'_i)) 40 if m̃' = ⊥ 41 K := H'_{L1}(c_i, m_j, c̃, i, j, M, M') 42 else 43 K := H'_{L2}(c_i, m_j, m̃', i, j, M, M') 44 else if m̃' = ⊥ 45 K := H'_{L3}(m'_i, m_j, c̃, i, j, M, M') 46 else K := H(m'_i, m_j, m̃', i, j, M, M') 47 sKey[sID] := K 48 received[sID] := M'</pre>
--	--

Figure 31: Games $G_{0,b} - G_{1,b}$ for case two of the proof of Theorem 4. Helper procedure ATTACK and oracles TEST, EST, CORRUPT, REVEAL and REV-STATE remains as in the original IND-StAA game (see Figures 17 and 18).

GAMES $G_{0,b}$. Since for both bits b , game $G_{0,b}$ is the original game IND-StAA_b,

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

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

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

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

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

As discussed below Definition 8, B's bit only counts in game IND-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 $\text{INIT}(\text{sID}^*)$ 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 $G_{2,b}^{\text{init}}$ to $G_{8,b}^{\text{init}}$, see Figure 32). Again, this is the only part of the proof where the correctness error comes into play. Next, during execution of $\text{INIT}(\text{sID}^*)$, we replace ciphertext c_j^* with a fake ciphertext that gets sampled using $\overline{\text{Enc}}$ (games $G_{9,b}^{\text{init}}$ to $G_{10,b}^{\text{init}}$, 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 $G_{12,b}^{\text{init}}$, also Figure 34).

GAMES $G_{1,b}^{\text{init}} - G_{8,b}^{\text{init}}$		DER _{resp} (sID, $M = (\tilde{pk}, c_j)$)	
01 Pick $2q$ -wise hash f	$\parallel G_{4,b}^{\text{init}} - G_{7,b}^{\text{init}}$	45 if holder[sID] = \perp or sKey[sID] $\neq \perp$	
02 cnt, sID* := 0		46 or role[sID] = "initiator" return \perp	
03 $j' \leftarrow_{\$} [N]$		46 role[sID] := "responder"	
04 for $n \in [N]$		47 $(j, i) := (\text{holder[sID]}, \text{peer[sID]})$	
05 $(pk_n, sk_n) \leftarrow \text{KG}$		48 $m_i, \tilde{m}_i \leftarrow_{\$} \mathcal{M}$	
06 $\mathbf{G} := \mathbf{G}_{pk_{j'}, sk_{j'}}$	$\parallel G_{4,b}^{\text{init}} - G_{7,b}^{\text{init}}$	49 $c_i := \text{Enc}(pk_i, m_i; \mathbf{G}(m_i))$	
07 $b' \leftarrow \mathbf{B}^{\mathbf{O}, \mathbf{G}, \mathbf{H}}((pk_n)_{n \in [N]})$		50 $\tilde{c} := \text{Enc}(pk, \tilde{m}; \mathbf{G}(\tilde{m}))$	
08 if ATTACK(sID*)		51 $M' := (c_i, \tilde{c})$	
09 return 0		52 $m'_j := \text{Dec}(sk_{j'}, c_j)$	
10 if $\mathfrak{N}(\text{sID}^*) \neq \emptyset$ ABORT		53 if $m'_j = \perp$	
11 if role[sID*] = "responder"	$\parallel G_{2,b}^{\text{init}} - G_{7,b}^{\text{init}}$	54 or $c_j \neq \text{Enc}(pk_{j'}, m'_j; \mathbf{G}(m'_j))$	
12 ABORT		54 $K' := \text{H}'_R(m_i, c_j, \tilde{m}, i, j, M, M')$	
13 if peer[sID*] $\neq j'$	$\parallel G_{3,b}^{\text{init}} - G_{8,b}^{\text{init}}$	55 if $j = j'$ and $i \neq j'$	$\parallel G_{7,b}^{\text{init}} - G_{9,b}^{-\text{sk}}$
14 return 0		56 $K' := \text{H}'_q(m_i, c_j, \tilde{m}, i, j, M, M')$	
15 return b'		57 if $i = j = j'$	$\parallel G_{7,b}^{\text{init}} - G_{9,b}^{-\text{sk}}$
		58 $K' := \text{H}'_q(c_i, c_j, \tilde{m}, j', i, M, M')$	$\parallel G_{7,b}^{\text{init}} - G_{9,b}^{-\text{sk}}$
		59 else	
DER _{init} (sID, $M' = (c_i, \tilde{c})$)		60 $K' := \text{H}(m_i, m'_j, \tilde{m}, i, j, M, M')$	
16 if holder[sID] = \perp or state[sID] = \perp		61 if $i = j'$ and $j \neq j'$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
17 or sKey[sID] $\neq \perp$ return \perp		62 $K' := \text{H}'_q(c_i, m'_j, \tilde{m}, j', i, M, M')$	
18 $(i, j) := (\text{holder[sID]}, \text{peer[sID]})$		63 if $j = j'$ and $i \neq j'$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
19 $(sk, m_j, M := (pk, c_j)) := \text{state[sID]}$		64 $K' := \text{H}'_q(m_i, c_j, \tilde{m}, i, j, M, M')$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
20 $m'_j := \text{Dec}(sk_i, c_i)$		65 if $i = j = j'$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
21 $\tilde{m}' := \text{Dec}(sk, \tilde{c})$		66 $K' := \text{H}'_q(c_i, c_j, \tilde{m}, j', i, M, M')$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
22 if $m'_j = \perp$ or $c_i \neq \text{Enc}(pk_i, m'_j; \mathbf{G}(m'_j))$		67 sKey[sID] := K'	
23 if $\tilde{m}' = \perp$		68 (received[sID], sent[sID]) := (M, M')	
24 $K := \text{H}'_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$		69 return M'	
25 else			
26 $K := \text{H}'_{L2}(c_i, m_j, \tilde{m}', i, j, M, M')$		$\mathbf{G}_{pk_{j'}, sk_{j'}}(m)$	
27 if $i = j'$ and $j \neq j'$	$\parallel G_{7,b}^{\text{init}}$	70 $r := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk_{j'}, sk_{j'}, m); f(m))$	
28 $K' := \text{H}'_q(c_i, m_j, \tilde{m}', j', i, M, M')$		71 return r	
29 if $i = j = j'$	$\parallel G_{7,b}^{\text{init}}$	$\text{H}(m_1, m_2, m_3, i, j, M, M')$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
30 $K' := \text{H}'_q(c_i, c_j, \tilde{m}', j', i, M, M')$		72 if $i = j'$ and $j \neq j'$	
31 else		73 return $\text{H}'_q(\text{Enc}(pk_{j'}, m_1; \mathbf{G}(m_1)), m_2, m_3, i, j, M, M')$	
32 if $\tilde{m}' = \perp$		74 if $j = j'$ and $i \neq j'$	
33 $K := \text{H}'_{L3}(m'_j, m_j, \tilde{c}, i, j, M, M')$		75 return $\text{H}'_q(m_1, \text{Enc}(pk_{j'}, m_2; \mathbf{G}(m_2)), m_3, i, j, M, M')$	
34 if $i = j'$		76 if $i = j = j'$	
35 $K := \text{H}'_{L1}(c_i, m_j, \tilde{c}, i, j, M, M')$	$\parallel G_{5,b}^{\text{init}}$	77 return $\text{H}'_q(\text{Enc}(pk_{j'}, m_1; \mathbf{G}(m_1)), \text{Enc}(pk_{j'}, m_2; \mathbf{G}(m_2)), m_3, i, j, M, M')$	$\parallel G_{6,b}^{\text{init}} - G_{8,b}^{\text{init}}$
36 else		78 return $\text{H}'(m_1, m_2, m_3, i, j, M, M')$	
37 $K := \text{H}(m'_i, m_j, \tilde{m}', i, j, M, M')$			
38 if $i = j'$ and $j \neq j'$	$\parallel G_{6,b}^{\text{init}}$		
39 $K' := \text{H}'_q(c_i, m_j, \tilde{m}', j', i, M, M')$			
40 if $j = j'$ and $i \neq j'$	$\parallel G_{6,b}^{\text{init}}$		
41 $K' := \text{H}'_q(m'_i, c_j, \tilde{m}', i, j, M, M')$			
42 if $i = j = j'$	$\parallel G_{6,b}^{\text{init}}$		
43 $K' := \text{H}'_q(c_i, c_j, \tilde{m}', j', i, M, M')$			
44 sKey[sID] := K			
45 received[sID] := M'			

Figure 32: Games $G_{1,b}^{\text{init}} - G_{8,b}^{\text{init}}$ for case (init) of the proof of Lemma 6. Helper procedure ATTACK and oracles INIT, TEST, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 17 and Figure 18, pages 28 and 29).

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

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

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

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

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

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 $\mathbf{G}_{pk,sk}$ that only samples randomness under which decryption never fails, allowing for a reduction to game GDPB.

GAMES $G_{3,b}^{\text{init}}$. In games $G_{3,b}^{\text{init}}$, 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.

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

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

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

and according to Lemma 1,

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

$\mathbf{C}_1^b = \mathbf{D}_1^b$	$\mathbf{C}_2^{b \mathbf{F}}, \mathbf{D}_2^{b \mathbf{F}}$
01 $(pk, sk) \leftarrow \text{KG}$	10 $\text{cnt}, \text{sID}^* := 0$
02 for $m \in \mathcal{M}$	11 $j' \leftarrow_{\$} [N]$
03 $\lambda(m) := \delta(pk, sk, m)$	12 for $n \in [N]$
04 return $(\lambda(m))_{m \in \mathcal{M}}$	13 $(pk_n, sk_n) \leftarrow \text{KG}$
G (m)	14 $b' \leftarrow \mathbf{B}^{O, \mathbf{G} , \mathbf{H} }((pk_n)_{n \in [N]})$
05 if $\mathbf{F}(m) = 0$	15 if $\text{ATTACK}(\text{sID}^*)$
06 $\mathbf{G}(m) := \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$	16 return 0
07 else	17 if $ \mathfrak{M}(\text{sID}^*) \neq 1$ ABORT
08 $\mathbf{G}(m) := \text{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$	18 Pick $\text{sID}_{\text{init}}^* \in \{\text{sID}^*, \text{sID}'\}$
09 return $\mathbf{G}(m)$	19 if $\text{corrupted}[\text{holder}[\text{sID}_{\text{init}}^*]]$ ABORT
	20 if $\text{holder}[\text{sID}_{\text{init}}^*] \neq j'$
	21 return 0
	22 return b'

Figure 33: Unbounded quantum adversaries \mathbf{C}^b and \mathbf{D}^b , executed in game $\text{GDPB}_{\delta(pk,sk)}$, for case $(\neg sk)$ of the proof of Lemma 5. The adversaries only differ in their definition of DER_{resp} , DER_{init} and **H**: For adversaries \mathbf{C}^b , DER_{resp} , DER_{init} and **H** are defined as in game $G_{3,b}^{\text{init}}$, see Figure 32, while for adversaries \mathbf{D}^b , DER_{resp} and DER_{init} and **H** are defined as in game $G_{7,b}^{\text{init}}$ (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 $G_{5,b}^{\text{init}}$. In both games $G_{5,b}^{\text{init}}$, 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. $\tilde{c} = \text{Enc}(pk, \tilde{m}', G(\tilde{m}'))$, the session key is defined as $K := \text{H}'_{\perp 1}(c_i, m_j, \tilde{c}, i, j, M, M')$. (Before this change we let $K := \text{H}'_{\perp 3}(m'_i, m_j, \tilde{c}, i, j, M, M')$ in the case that \tilde{c} fails to decrypt, but c_i decrypts correctly.) Due to the same argument as for games $G_{6,b}^{-sk}$ (that the cases are logically distinct, see page 54), B's view does not change and

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

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 not use the decryptions under $sk_{j'}$, but the ciphertexts themselves. Similar to games $G_{7,b}^{-sk}$ (see page 55), we "plug in" encryption into random oracle H whenever j' appears as one of the involved parties.

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

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

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 $G_{7,b}^{-sk}$, 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 $G_{7,b}^{-sk}$ (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

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

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 $G_{8,b}^{-sk}$ (see page 57), we also use H'_q to H^3_q if a ciphertext fails to de-or reencrypt under $sk_{j'}$, instead of using the implicit reject oracles.

GAMES $G_{7,b}^{\text{init}}$. In games $G_{7,b}^{\text{init}}$, 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 H^2_q or H^3_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 H'_q or H^3_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 $G_{7,b}^{\text{init}}$, 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 $G_{7,b}^{\text{init}}$.

Due to the same argument as for games $G_{8,b}^{-sk}$ ($G_{pk_{j'}, sk_{j'}}$ 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

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

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

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

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

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

GAMES $G_{8,b}^{\text{init}} - G_{12,b}^{\text{init}}$	INIT(sID)
01 cnt, sID* := 0	19 if holder[sID] = \perp or sent[sID] $\neq \perp$
02 $j' \leftarrow_{\mathcal{S}} [N]$	20 return \perp
03 for $n \in [N]$	21 role[sID] := "initiator"
04 $(pk_n, sk_n) \leftarrow \text{KG}$	22 $i := \text{holder}[\text{sID}], j := \text{peer}[\text{sID}]$
05 $s' \leftarrow_{\mathcal{S}} [S]$ // $G_{9,b}^{\text{init}} - G_{12,b}^{\text{init}}$	23 $m_j \leftarrow_{\mathcal{S}} \mathcal{M}$
06 $b' \leftarrow \mathcal{B}^{\text{O}, \langle \mathcal{G} \rangle, \langle \mathcal{H} \rangle}((pk_n)_{n \in [N]})$	24 $c_j := \text{Enc}(pk_j, m_j; \mathcal{G}(m_j))$
07 if ATTACK(sID*)	25 if sID = s'
08 return 0	26 $c_j \leftarrow \overline{\text{Enc}}(pk_{j'})$ // $G_{10,b}^{\text{init}} - G_{12,b}^{\text{init}}$
09 if $\mathfrak{M}(\text{sID}^*) \neq \emptyset$ ABORT	27 if $c_j \in \text{Enc}(pk_{j'}, \mathcal{M}; \mathcal{R})$
10 if role[sID*] = "responder"	28 ABORT // $G_{11,b}^{\text{init}} - G_{12,b}^{\text{init}}$
11 ABORT	29 $(\tilde{sk}, \tilde{pk}) \leftarrow \text{KG}$
12 if peer[sID*] $\neq j'$	30 $M := (\tilde{pk}, c_j)$
13 return 0	31 state[sID] := (\tilde{sk}, m_j, M)
14 if peer[sID*] $\neq j'$	32 sent[sID] := M
15 return 0	33 return M
16 if sID* $\neq s'$	
17 return 0 // $G_{9,b}^{\text{init}} - G_{12,b}^{\text{init}}$	<u>TEST(sID)</u> // only one query
18 return b'	34 sID* := sID
	35 if sKey[sID*] = \perp return \perp
	36 $K_0^* := \text{sKey}[\text{sID}^*]$ // $G_{8,b}^{\text{init}} - G_{10,b}^{\text{init}}$
	37 $K_0^* \leftarrow_{\mathcal{S}} \mathcal{K}$ // $G_{12,0}^{\text{init}}$
	38 $K_1^* \leftarrow_{\mathcal{S}} \mathcal{K}$
	39 return K_b^*

Figure 34: Games $G_{8,b}^{\text{init}} - G_{12,b}^{\text{init}}$ for case (init) of the proof of Lemma 6. All oracles except for INIT and TEST remain as in game $G_{8,b}^{\text{init}}$ (see Figure 32).

We stress that from game $G_{8,b}^{\text{init}}$ 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 $G_{9,b}^{\text{init}}$. In games $G_{9,b}^{\text{init}}$, 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^* .

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

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

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

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

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

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

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

$A_{DS,b}^{\text{init}}(H'\rangle, H_q\rangle, G\rangle)(pk, c)$ 01 cnt, sID* := 0 02 $j' \leftarrow_{\S} [N]$ 03 $s' \leftarrow_{\S} [S]$ 04 for $n \in [N] \setminus \{j'\}$ 05 $(pk_n, sk_n) \leftarrow \text{KG}$ 06 $(pk_{j'}, sk_{j'}) := (pk, \perp)$ 07 $b' \leftarrow \mathcal{B}^{\mathcal{O}, \text{RO}\rangle}(pk_1, \dots, pk_N)$ 08 if ATTACK(sID*) 09 return 0 10 if $\mathfrak{M}(\text{sID}^*) \neq \emptyset$ ABORT 11 if role[sID*] = "responder" 12 ABORT 13 if peer[sID*] $\neq j'$ return 0 14 if peer[sID*] $\neq j'$ return 0 15 if sID* $\neq s'$ return 0 16 return b' CORRUPT($i \in [N] \setminus \{j'\}$) 17 if corrupted[i] return \perp 18 corrupted[i] := true 19 return sk_i	INIT(sID) 20 if holder[sID] = \perp 21 return \perp 22 if sent[sID] $\neq \perp$ 23 return \perp 24 role[sID] := "initiator" 25 $i := \text{holder}[\text{sID}]$ 26 $j := \text{peer}[\text{sID}]$ 27 $m_j \leftarrow_{\S} \mathcal{M}$ 28 $c_j := \text{Enc}(pk_j, m_j; \mathcal{G}(m_j))$ 29 if sID = s' 30 $c_j := c$ 31 $(\tilde{sk}, \tilde{pk}) \leftarrow \text{KG}$ 32 $M := (\tilde{pk}, c_j)$ 33 state[sID] := (\tilde{sk}, m_j, M) 34 sent[sID] := M 35 return M
---	---

Figure 35: Adversaries $A_{DS,b}^{\text{init}}$ for case (init) of the proof of Lemma 6, with oracle access to $|H'\rangle$, $|H_q\rangle$ and $|G\rangle$. All oracles except for INIT and CORRUPT are defined as in game $G_{9,b}^{\text{init}}$ (see Figure 34).

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

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

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

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

It remains to upper bound $|\Pr[G_{12,0}^{\text{init B}} \Rightarrow 1] - \Pr[G_{11,0}^{\text{init B}} \Rightarrow 1]|$, which means upper bounding the probability that B obtained the test session's key in game $G_{11,0}^{\text{init B}}$ 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 $\text{Enc}(pk_{j'}, -; -)$, this is impossible.

With an argument similar to the previous cases, we can show that none of the quantum answers of $|H\rangle$ 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 $|H_q\rangle$ such that its answer contains the key, B would need to come up with a message m such that $\text{Enc}(pk_{j'}, m; \mathcal{G}(m)) = c_j^*$, which is impossible.

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

Collecting the probabilities, we obtain

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

CASE (resp). Intuition is as follows: While B could pick message $(c_j, \tilde{p}k)$ 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 $\text{DER}_{\text{resp}}(\text{sID}^*, (c_j, \tilde{p}k))$) cannot be computed trivially. The proof differs from case (init) only in the following way: instead of changing $\text{INIT}(\text{sID}^*)$ such that it outputs a fake encryption c_j , we change $\text{DER}_{\text{resp}}(\text{sID}^*, m)$ such that it outputs a fake encryption c_i . In the last game, c_i does not lie in the range of $\text{Enc}(pk_{i'}, -, -)$ 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 $\mathbf{A}_{\text{DS}}^{\text{resp}}$ such that

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

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

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

the upper bound bound given in Lemma 6.

C 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 6, and that plugging transformation **Punc** into transformation **T** achieves DS and CPA security from CPA security (see ??):

Corollary C.1 (DS security of **TPunc**). *For all adversaries \mathbf{A} issuing at most q_G queries to $|\text{G}|$, there exist two adversaries \mathbf{B}_1 and \mathbf{B}_2 such that*

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

and the running time of each adversary is about that of \mathbf{B} .

C.1 Proof of Theorem 5

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

Theorem (CCA security of $\text{FO}_{\text{AKE}}^{\neq} \circ \text{Punc}$). Assume PKE_0 to be δ -correct and γ -spread, and let $\hat{m} \in \mathcal{M}$. Let $\text{AKE} := \text{FO}_{\text{AKE}}[\text{Punc}[\text{PKE}, \hat{m}], \text{G}, \text{H}]$. Then, for any IND-StAA adversary \mathbf{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 \mathbf{B}_1 and \mathbf{B}_2 such that

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

and the running time of \mathbf{B}_1 and \mathbf{B}_2 is about that of \mathbf{B} .

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

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

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

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

and

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

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

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

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

and

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

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

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

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

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

and

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

Folding the adversaries, we obtain

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