# Generic Authenticated Key Exchange in the Quantum Random Oracle Model

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

February 14, 2019

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

**Abstract**

We propose $\mathsf{FO}_{\mathsf{AKE}}$, a generic construction of two-message authenticated key exchange (AKE) from any passively secure public key encryption (PKE) in the quantum random oracle model (QROM). Whereas previous AKE constructions relied on a Diffie-Hellman key exchange or required the underlying PKE scheme to be perfectly correct, our transformation allows arbitrary PKE schemes with non-perfect correctness. 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 post-quantum competition, e.g., ones based on codes and lattices.

$\mathsf{FO}_{\mathsf{AKE}}$ can be seen as a generalization of the well known Fujisaki-Okamoto transformation (for building actively secure PKE from passively secure PKE) to the AKE setting. Therefore, as a helper result, we also provide a security proof for the Fujisaki-Okamoto transformation in the QROM for PKE with non-perfect correctness. Our reduction fixes several gaps in a previous proof (CRYPTO 2018), is tighter, and tolerates a larger correctness error.

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

## 1 Introduction

Authenticated Key Exchange. Besides public key encryption (PKE) and digital signatures, authenticated key exchange (AKE) is 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 [BR94, CK01, LLM07, JKSS12] as well as increasingly efficient designs of AKE protocols [Kra05, YZ13, Sch15]. 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 [BCK98, BCNP08]. Even rarer are constructions that only rely on PKE, discarding Diffie-Hellman key exchanges entirely. Notable recent exceptions are [FSXY12] and the protocol in [ABS14], the latter of which has been criticized for having a flawed security proof and a weak security model [Too15, LS17].

The NIST Post-Quantum Competition. Recently, some of the above mentioned designs have gathered renewed interest in the quest of finding AKE protocols that are secure against quantum adversaries, i.e., adversaries equipped with a quantum computer. In particular, the National Institute of Standards and Technology (NIST) announced a competition with the goal to standardize new PKE and signature algorithms [NIS17] with security against quantum adversaries. With the understanding that an AKE protocol can be constructed from low level primitives such as quantum-secure PKE and signature schemes, the NIST did not require the submissions to describe a concrete AKE protocol. Natural PKE and signature candidates base their security on the hardness of certain problems over lattices and codes, which are generally believed to resist quantum adversaries.

The quantum ROM. Quantum computers may execute all "offline primitives" such as hash functions on arbitrary superpositions, which motivated the introduction of the quantum (accessible) random oracle model (QROM) [BDF+11]. 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] We insist that 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.) 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 unfit for usage in all previous constructions that relied on perfect correctness.[2]

Previous Constructions of AKE from PKE. The generic AKE protocol of Fujioka et al. [FSXY12] (itself based on [BCNP08]) transforms a passively secure PKE scheme $\mathsf{PKE}$ and an actively (i.e., $\mathsf{IND\text{-}CCA}$) secure PKE scheme $\mathsf{PKE}_{\mathrm{cca}}$ into an AKE protocol. We will refer to this transformation as $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{PKE}_{\mathrm{cca}}]$. Since the $\mathsf{FSXY}$ transformation is in the standard model, it is likely to be secure with the same proof in the post-quantum setting and thus also in the QROM. The standard way to obtain actively secure encryption from passively secure ones is the Fujisaki-Okamoto transformation $\mathsf{PKE}_{\mathrm{cca}} = \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ [FO99, FO13]. In its "implicit rejection" variant [HHK17], it comes with a recently discovered security proof [SXY18] that models the hash functions $\mathsf{G}$ and $\mathsf{H}$ as quantum random oracles. Indeed, the combined AKE transformation $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]]$ transforms passively secure encryption into AKE that is very likely to be secure in the QROM, without using digital signatures, hence giving a first answer to our above question. It has, however, two main drawbacks.

- **Perfect correctness requirement.** Transformation $\mathsf{FSXY}$ is not known to have a security proof if the underlying scheme does not satisfy perfect correctness. Likewise, the relatively tight QROM proof for $\mathsf{FO}$ that was given in [SXY18] requires the underlying scheme to be perfectly correct, and the generalisation of the proof for schemes with non-perfect correctness is not straightforward. Since there were no results on how non-perfect correctness of $\mathsf{PKE}$ influences the security of $\mathsf{FSXY}[\mathsf{PKE}, \mathsf{FO}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]]$, it was unclear whether it was fit to be used with lattice- or code-based encryption schemes.
- **Overly complicated?** The Fujisaki-Okamoto transformation already involves hashing the key using hash function $\mathsf{H}$, and $\mathsf{FSXY}$ involves even more (potentially redundant) hashing of the (already hashed) session key. Overall, the combined transformation seems overly complicated and hence impractical.

Hence, it seems desirable to provide a simplified transformation that gets rid of unnecessary hashing steps, and that can be proven secure in the quantum random oracle model even if the underlying scheme

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

[2] There exist generic transformations that can immunize against decryption errors (e.g., [DNR04]). Even though they are quite efficient in theory, the induced overhead is still not acceptable for practical purposes.

does not come with perfect correctness. As a motivating example, note that the Kyber AKE protocol [BDK+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, $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ ("Fujisaki-Okamoto for AKE") that converts any passively secure encryption scheme into an actively secure AKE protocol, with provable security in the quantum random oracle model. It can deal with non-perfect correctness and does not use digital signatures. Furthermore, we provide a precise game-based security definition for two-message AKE protocols. As a side result, we 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. We want to stress that a security proof for the Fujisaki-Okamoto transformation in the QROM was already given in the independent work of [JZC+18a], but since we identified some flaws and since our proof structurally differs from the one given [JZC+18a], we decided to include our KEM proof to illustrate our techniques and to keep our AKE proof as comprehensible as possible.

### 1.1.1 Improved bounds and analysis for the Fujisaki-Okamoto transformation $\mathsf{FO}_m^{\not\perp}$.

To simplify the presentation of $\mathsf{FO}_{\mathsf{AKE}}$, we first give some background on the Fujisaki-Okamoto transformation. In its original form [FO99, FO13], FO yields an encryption scheme that is IND-CCA secure in the random oracle model [BR93] 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 [Den03] 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 [CS03].) 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, [HHK17] gave several modularizations of FO-like transformations and proved them robust against correctness errors. The key observation was that FO-like transformations essentially consists of two separate steps and can be dissected into two transformations, as sketched in the introduction of [HHK17]:

- Transformation $\mathsf{T}$ ([BBO07], [BHSV98, Sec. 5]): "Derandomization" and "re-encryption". Starting from an encryption scheme $\mathsf{PKE}$ and a hash function $\mathsf{G}$, encryption of $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is defined by
$$\mathsf{Enc}'(pk, m) := \mathsf{Enc}(pk, m; \mathsf{G}(m)),$$
where $\mathsf{G}(m)$ is used as the random coins for $\mathsf{Enc}$, rendering $\mathsf{Enc}'$ deterministic. $\mathsf{Dec}'(sk, c)$ first decrypts $c$ into $m'$ and rejects if $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$ ("re-encryption").

- Transformation $\mathsf{U}_m^{\not\perp}$: "Hashing". Starting from an encryption scheme $\mathsf{PKE}'$ and a hash function $\mathsf{H}$, key encapsulation mechanism $\mathsf{KEM}_m^{\not\perp} = \mathsf{U}_m^{\not\perp}[\mathsf{PKE}', \mathsf{H}]$ with "implicit rejection" is defined by
$$\mathsf{Encaps}(pk) := (c \leftarrow \mathsf{Enc}'(pk, m), K := \mathsf{H}(m)), \tag{1}$$
where $m$ is picked at random from the message space, and
$$\mathsf{Decaps}(sk, c) = \begin{cases} \mathsf{H}(m) & m \neq \perp \\ \mathsf{H}(s, c) & m = \perp \end{cases},$$
where $m := \mathsf{Dec}(sk, c)$ and $s$ is a random seed which is contained in $sk$. In the context of the FO transformation, implicit rejection was first introduced by Persichetti [Per12, Sec. 5.3].

Transformation $\mathsf{T}$ was proven secure both in the (classical) ROM and the QROM, and $\mathsf{U}_m^{\not\perp}$ was proven secure in the ROM. To achieve QROM security, [HHK17] gave a modification of $\mathsf{U}_m^{\not\perp}$, called $\mathsf{QU}_m^{\not\perp}$, but its security proof in the QROM suffered from a quartic loss in tightness, and most real-world proposals are designed such that they fit the framework of $\mathsf{FO}_m^{\not\perp} = \mathsf{U}_m^{\not\perp} \circ \mathsf{T}$, not $\mathsf{QU}_m^{\not\perp} \circ \mathsf{T}$.

A slightly different modularization was introduced in [SXY18]: they gave transformations TPunc ("Puncturing and Encrypt-with-Hash") and SXY ("Hashing with implicit reject and reencryption"). SXY differs from $\mathsf{U}^{\not\perp}_m$ in that it reencrypts during decryption. Hence, it can only be applied to deterministic schemes. Even in the QROM, its CCA security tightly reduces to an intermediate notion called Disjoint Simulatability (DS) of ciphertexts. Intuitively, disjoint simulatability means that we can efficiently sample "fake ciphertexts" that are computationally indistinguishable from real PKE ciphertexts ("simulatability"), while the set of possible fake ciphertexts is required to be (almost) disjoint from the set of real ciphertexts. DS is naturally satisfied by many code/lattice-based encryption schemes. Additionally, it can be achieved using transformation Punc, i.e., by puncturing the underlying schemes' message space at one point and using this message to sample fake encryptions. Deterministic DS can be achieved by using transformation TPunc, albeit non-tightly (due to the use of the oneway-to-hiding lemma).

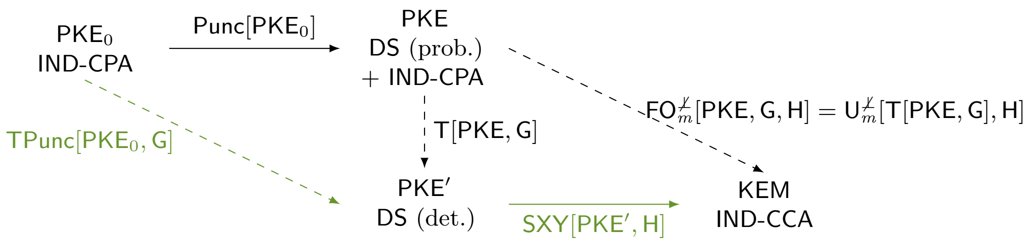



Figure 1: Comparison of [SXY18]'s modular transformation (green) with ours. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions.

However, the reduction that is given in [SXY18] requires the underlying encryption scheme to be perfectly correct. While [JZC+18a, JZC+18b] ([JZC+18b] refers to the full version of [JZC+18a] in its last revision from July 2018) gave security proofs for the non-modular transformations $\mathsf{FO}^{\not\perp}_m$ and $\mathsf{FO}^{\not\perp}$ [JZC+18b, Thms. 1 and 2] as well as a security proof for SXY[3] (see [JZC+18b, Thm. 6]) for schemes with correctness errors. We identified some flaws and drawbacks which we will discuss in Appendix A. In a nutshell, two main issues arise: The first issue is that to prove the non-modular statements, a lemma is used whose formal statement is unclear. One of its requirements might be unsatisfiable, rendering the proof impossible to verify. We structure our proof differently by following [SXY18]'s modular approach as far as possible.[4] For more details on this issue and our strategy to avoid it, we refer to Appendix A.

The second issue is that the security statement given in [JZC+18b, Thm. 6] is based on prerequisites that are not met by most lattice-based encryption schemes. Recall that SXY is only applicable to deterministic schemes since it reencrypts, and the issue stated above is due to the correctness definition for deterministic schemes that is used.[5] It is not straightforward to give a correctness definition for deterministic encryption schemes such that it fits known strategies to prove SXY tightly secure, but also is achievable by most lattice-based schemes. We circumvent this difficulty by resorting to a non-modularized proof that assumes a non-deterministic scheme.[6] Lastly, we want to stress that the statement of [JZC+18b, Thm. 6] is not proven, and it is unclear how it could be proven with the standard notion of IND-CCA security. More details on these issues are also given in Appendix A.

Transformation $\mathsf{FO}^{\not\perp}_m$ can be applied to any PKE scheme that is both IND-CPA and DS secure. The reduction is tighter than the one that results from combining those of TPunc and SXY in [SXY18], and also than the reduction given in [JZC+18b]. This is due to our use of the improved Oneway-to-Hiding lemma [AHU18, Thm. 1: "Semi-classical O2H"]. Furthermore, we achieve a better correctness bound (the square of the bound given in [JZC+18b]) 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:

[3] Note that the papers' nomenclature is misleading: while the KEM discussed in theorem 6 and given in figure 13 is called $\mathsf{U}^{\not\perp}_m$, it is transformation SXY (it reencrypts during decryption, which transformation $\mathsf{U}^{\not\perp}_m$ does not).

[4] We will first prove that $\mathsf{T}[-,\mathsf{G}]$ turns any suitable scheme into a scheme that is deterministically DS, and then plug in this result into [SXY18]'s tight security proof for $\mathsf{U}^{\not\perp}_m$.

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

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

To rely on $\mathsf{IND\text{-}CPA}$ alone, all that has to be done is to plug in transformation $\mathsf{Punc}$. A visualization is given in Figure 1.

#### 1.1.2 Rigorous Security Model for Two-Message Authenticated Key Exchange.

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

Details on our Models. We define two security notions for two-message AKEs: key indistinguishability against active attacks ($\mathsf{IND\text{-}AA}$) and the weaker notion of indistinguishability against active attacks without state reveal in the test session ($\mathsf{IND\text{-}StAA}$). $\mathsf{IND\text{-}AA}$ captures the classical notion of key indistinguishability (as introduced by Bellare and Rogaway [BR94]) as well as security against reflection attacks, key compromise impersonation (KCI) attacks, and weak forward secrecy (wFS) [Kra05]. 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 [BCNP08], this makes our model incomparable to the eCK model [LLM07] but strictly stronger than the CK model. Essentially, the $\mathsf{IND\text{-}AA}$ model states that the session key remains indistinguishable from a random one even if

1. the attacker knows either the long-term secret key or the secret state information (but not both) of both parties involved in the test session, as long as it did not modify the message received by the test session,

2. and also if the attacker modified the message received by the test session, as long as it did not obtain the long-term secret key of the test session's peer.

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

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

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

We remark that $\mathsf{IND\text{-}StAA}$ security is essentially the same notion that was achieved by the $\mathsf{FSXY}$ transformation [FSXY12].[7]

#### 1.1.3 Our Authenticated Key-Exchange Protocol.

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

Transformation $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is described in Figure 2 and uses transform $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ as a building block. (The full construction is given in Figure 18, see Section 5.) Our main security result (Theorem 5.1) states that $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ is an $\mathsf{IND\text{-}StAA}$-secure AKE if the underlying probabilistic $\mathsf{PKE}$ is $\mathsf{DS}$ as well as $\mathsf{IND\text{-}CPA}$ secure and has negligible correctness error, and furthermore $\mathsf{G}$ and $\mathsf{H}$ are modeled as quantum random oracles.

The proof essentially is the AKE analogue to the security proof of $\mathsf{FO}_m^{\not\perp}$ we give in Section 3.2: By definition of our security model, it always holds that at least one of the messages $m_i$, $m_j$ and $\tilde{m}$ is hidden from the adversary (unless it loses trivially). Adapting the simulation technique in [SXY18], we can simulate the session keys even if we do not know the corresponding secret key $sk_i$ ($sk_j$, $\tilde{sk}$). Assuming that $\mathsf{PKE}$ is $\mathsf{DS}$, we can replace the corresponding ciphertext $c_i$ ($c_j$, $\tilde{c}$) of the test session with a fake

[7] The difference is that the model from [FSXY12] furthermore allows a "partial reveal" of the test session's state. For simplicity and due to their little practical relevance, we decided not to include such partial session reveal queries in our model. We remark that, however, our protocol could be proven secure in this slightly stronger model.

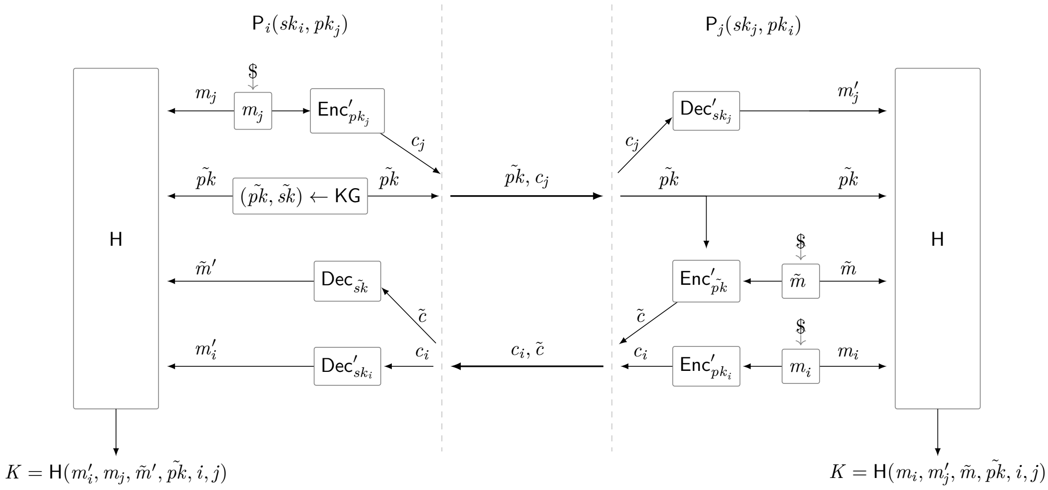



Figure 2: A visualisation of our authenticated key-exchange protocol $\mathsf{FO}_{\mathsf{AKE}}$. We make the convention that, in case any of the $\mathsf{Dec}'$ algorithms returns $\perp$, the session key $K$ is derived deterministically and pseudorandomly from the player's state ("implicit rejection").

ciphertext, rendering the test session's key completely random from the adversary's view due to PKE's disjointness.

Let us add two remarks. Firstly, we cannot prove the security of $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ in the stronger sense of IND-AA and actually, it is not secure against state attacks. Secondly, note that our security statement involves the probabilistic scheme PKE rather than PKE′. Unfortunately, we were not able to provide a modular proof of AKE solely based on reasonable security properties of $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$. The reason for this is indeed the non-perfect correctness of PKE. This difficulty corresponds to the difficulty to generalize [SXY18]'s result for deterministic encryption schemes with correctness errors discussed above.

Concrete Applications. Our transformation can be applied to any DS and IND-CPA secure PKE scheme with post-quantum security, e.g., Frodo [NAB+17], Kyber [BDK+17], and Lizard [BI17]. In fact, applying $\mathsf{FO}_{\mathsf{AKE}}$ to Kyber provides a formal security proof for the AKE protocol described in [BDK+17]. Note that most of the mentioned schemes are already DS secure under the same assumption as it is used for IND-CPA security and as mentioned above, the requirement of DS security can be waived with negligible loss of efficiency.

#### 1.1.4 Open Problems.

In the literature, one can find several Diffie-Hellman based protocols that achieve IND-AA security, for example HMQV [Kra05]. 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. While we were able to generalize (and tighten) the proof of CCA security given in [SXY18] for the *combined* transformation $\mathsf{FO}_m^{\not\perp} := \mathsf{U}_m^{\not\perp} \circ \mathsf{T}$ such that it covers encryption schemes that come with non-perfect correctness, it still remains an open problem to generalize the security proof of $\mathsf{U}_m^{\not\perp}$ such that it is applicable to *any* deterministic encryption scheme that is DS, even if it is not perfectly correct for more than neglibly many key pairs.

# 2 Preliminaries

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

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

Games. Following [Sho04, BR06], we use code-based games. We implicitly assume boolean flags to be initialized to false, numerical types to 0, sets to $\varnothing$, and strings to the empty string $\epsilon$. We make the convention that a procedure terminates once it has returned an output.

## 2.1 Public-key Encryption

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

**Definition 2.1** (Collision probability of key generation.). We define

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

Correctness. [HHK17] We define $\delta := \mathbf{E}[\max_{m \in \mathcal{M}} \Pr[c \leftarrow \mathsf{Enc}(pk, m) : \mathsf{Dec}(sk, c) \neq m]]$, where the expectation is taken over $(pk, sk) \leftarrow \mathsf{KG}$.

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

**Definition 2.2** (IND-CPA). Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. We define game IND-CPA game as in Figure 3, and the IND-CPA advantage function of a quantum adversary $\mathsf{A} = (\mathsf{A}_1, \mathsf{A}_2)$ against PKE (such that $\mathsf{A}_2$ has binary output) as

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

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

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

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

Disjoint simulatability. Following [SXY18], 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 2.3** (DS) [SXY18] Let $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, together with a PPT algorithm $\overline{\mathsf{Enc}}$. For quantum adversaries A, we define the *advantage against* PKE*'s disjoint simulatability* as

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

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

## 2.2 Key Encapsulation

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

We call $\mathsf{KEM}$ $\delta$-*correct* if

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

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

SECURITY. We now define a security notion for key encapsulation: Indistinguishbility under Chosen Ciphertext Attacks (IND-CCA).

**Definition 2.4** (IND-CCA). We define the IND-CCA game as in Figure 3 and the IND-CCA *advantage function of an adversary* A *(with binary output) against* KEM as

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

## 2.3 Quantum Computation

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

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

MEASUREMENTS. Qubits can be measured with respect to a basis. In this paper, we will only consider measurements in the standard orthonormal computational basis, and denote this measurement by MEASURE$(\cdot)$, where the outcome of MEASURE$(|\varphi\rangle)$ for a single qubit $|\varphi\rangle = \alpha \cdot |0\rangle + \beta \cdot |1\rangle$ will be 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$, and the outcome of measuring a qubit register $|\varphi\rangle = \sum_{x \in \mathbb{F}_2^n} \alpha_x \cdot |x\rangle$ will be $x$ with probability $|\alpha_x|^2$. Note that the amplitudes *collapse* during a measurement, this means that by measuring $\alpha \cdot |0\rangle + \beta \cdot |1\rangle$, $\alpha$ and $\beta$ 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 [BDF+11, BBC+98], we view a quantum oracle $|\mathsf{O}\rangle$ as a mapping

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

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

QUANTUM RANDOM ORACLE MODEL. We consider security games in the quantum random oracle model (QROM) as their counterparts in the classical random oracle model, with the difference that we consider quantum adversaries that are given **quantum** access to the (offline) random oracles involved, and **classical** access to all other (online) oracles. For example, in the IND-CPA game, the adversary only obtains a classical encryption, like in [BJ15], and unlike in [BZ13]. In the IND-CCA game, the adversary only has access to a classical decryption oracle, unlike in [GHS16] and [AJOP18].

Zhandry [Zha12] proved that no quantum algorithm $\mathsf{A}^{|\mathsf{O}\rangle}$, issuing at most $q$ quantum queries to $|\mathsf{O}\rangle$, can distinguish between a random function $\mathsf{O}: \mathbb{F}_2^m \to \mathbb{F}_2^n$ and a $2q$-wise independent function $f_{2q}$. For concreteness, we view $f_{2q}: \mathbb{F}_2^m \to \mathbb{F}_2^n$ as a random polynomial of degree $2q$ over the finite field $\mathbb{F}_{2^n}$. The running time to evaluate $f_{2q}$ is linear in $q$. In this article, we will use this observation in the context of security reductions, where quantum adversary $\mathsf{B}$ simulates quantum adversary $\mathsf{A}^{|\mathsf{O}\rangle}$ issuing at most $q$ queries to $|\mathsf{O}\rangle$. Hence, the running time of $\mathsf{B}$ is $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q \cdot \mathrm{Time}(\mathsf{O})$, where $\mathrm{Time}(\mathsf{O})$ denotes the time it takes to simulate $|\mathsf{O}\rangle$. Using the observation above, $\mathsf{B}$ can use a $2q$-wise independent function in order to (information-theoretically) simulate $|\mathsf{O}\rangle$, and we obtain that the running time of $\mathsf{B}$ is $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q \cdot \mathrm{Time}(f_{2q})$, and the time $\mathrm{Time}(f_{2q})$ to evaluate $f_{2q}$ is linear in $q$. Following [SXY18] and [KLS18], 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 $\mathsf{B}$ can simply use another random oracle to simulate $|\mathsf{O}\rangle$. Assuming evaluating the random oracle takes one time unit, we write $\mathrm{Time}(\mathsf{B}) = \mathrm{Time}(\mathsf{A}) + q$, which is approximately $\mathrm{Time}(\mathsf{A})$.

Oneway to Hiding with semi-classical oracles. In [AHU18], Ambainis et al. defined semi-classical oracles that return a state that was measured with respect to one of the input registers. In particular, to any subset $S \subset X$, they associated the following semi-classical oracle $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$: Algorithm $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$, when queried on $|\psi, 0\rangle$, measures with respect to the projectors $M_1$ and $M_0$, where $M_1 := \sum_{x \in S} |x\rangle\langle x|$ and $M_0 := \sum_{x \notin S} |x\rangle\langle x|$. The oracle then initializes the second register to $|b\rangle$ for the measured bit $b$. This means that $|\psi, 0\rangle$ collapses to either a state $|\psi', 0\rangle$ such that $|\psi'\rangle$ only contains elements of $X \setminus S$ or to a state $|\psi', 1\rangle$ such that $|\psi'\rangle$ only contains elements of $S$. Let FIND denote the event that the latter ever is the case, i.e., that $\mathsf{O}_\mathsf{S}^{\mathsf{SC}}$ ever answers with $|\psi', 1\rangle$ for some $\psi'$. To a quantum oracle $|\mathsf{G}\rangle$ and a subset $S \subset X$, Ambainis et al. associate the following punctured oracle $|\mathsf{G} \setminus \mathsf{S}\rangle$ that removes $S$ from the domain of $|\mathsf{G}\rangle$ unless FIND occurs.

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

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

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

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

The following theorem is a simplification of statement (2) given in [AHU18, Thm. 1: "Semi-classical O2H"]. It differs in the following way: While [AHU18] 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 2.5** *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 $\mathsf{A}$ issuing at most $q$ queries that, on input $z$, output either 0 or 1,*

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

**Theorem 2.6** *([AHU18, Cor. 1]) Suppose that $S := \{x\}$ for $x \leftarrow_\$ X$, and that $x$ and $z$ are independent. Then, for all quantum algorithms $\mathsf{A}$ issuing at most $q$ queries,*

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

Generic quantum Distinguishing Problem with bounded probabilities. For $\lambda \in [0, 1]$, let $B_\lambda$ 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 ([ARU14, Lemma 37: "Preimage search in a random function" ], [HRS16, Lem. 3]) is to distinguish quantum access to an oracle $F : X \to \mathbb{F}_2$, such that for each $x \in X$ , $F(x)$ is distributed according to $B_\lambda$, 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 $\lambda_x$ may depend on $x$, but still is upper bounded by a global $\lambda$. The upper bound we give is the same as in [HRS16, Lem. 3].

**Lemma 2.7** (Generic Distinguishing Problem with Bounded Probabilities). *Let $X$ be a finite set, and let $\lambda \in [0,1]$. Then, for any (unbounded, quantum) algorithm $\mathsf{A}$ issuing at most $q$ quantum queries,*

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

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

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

*Proof.* In this proof, let $\mathsf{CGDPB}_\lambda$ denote the game $\mathsf{GDPB}_\lambda$ as defined in [ARU14] and [HRS16], i.e., defined such that $\lambda_x = \lambda$ for all $x$. (Hence, we call it constant GDPB). The bound on $\mathsf{GDPB}_\lambda$ can be reduced to the known bound on $\mathsf{CGDPB}_\lambda$ by coupling the Bernoulli parameter to obtain the dependence on each $x \in X$: Let $\mathsf{A}$ be an adversary against game $\mathsf{GDPB}_\lambda$, issuing at most $q$ queries. Without loss of generality, we can assume that $\lambda > 0$. Consider adversary $\mathsf{B}$ against game $\mathsf{CGDPB}_\lambda$, given in Figure 5.

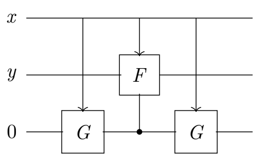
| $\mathsf{B}_1$ | $\mathsf{B}_2^{\|\mathsf{F}\rangle}$ |
|---|---|
| $(\lambda_x)_{x\in X} \leftarrow \mathsf{A}_1$ | $b' \leftarrow \mathsf{A}_2^{\|\mathsf{F}\cdot\mathsf{G}\rangle}$ |
| $\lambda := \max_{x\in X} \lambda_x$ | **return** $b'$ |
| **for** all $x \in X$ | |
| $\mu_x := \frac{\lambda_x}{\lambda}$ | |
| $G(x) \leftarrow B_{\mu_x}$ | |
| **return** $\lambda$ | |

Figure 5: Adversary $\mathsf{B}$ for the proof of Lemma 2.7.

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

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

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




The dot indicates execution of $F(x)$, conditioned on $G(x)$. It's easy to see that $|x,y,0\rangle$ transitions to $|x, y \oplus F(x), 0\rangle$ if $G(x) = 1$, and that $|x,y,0\rangle$ transitions to $|x,y,0\rangle$ if $G(x) = 0$, hence $|x,y,0\rangle$ transitions to $|x, y \oplus (F \cdot G)(x), 0\rangle$, either way, and $\mathsf{B}$ can answer queries to $|\mathsf{F} \cdot \mathsf{G}\rangle$ by querying $|\mathsf{F}\rangle$ just once. Since $\mathsf{B}$ issues at most $q$ queries to $|\mathsf{F}\rangle$, we can apply [HRS16, Lem. 3] and obtain

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

□

# 3 The FO Transformation: QROM security with correctness errors

In Section 3.1, we modularize transformation TPunc that was given in [SXY18] and that turns any public key encryption scheme that is IND-CPA secure into a deterministic one that is DS. We show that TPunc essentially consists of first puncturing the message space at one point (transformation Punc, to achieve DS), and then applying transformation T. Next, in Section 3.2, we show that transformation $\mathsf{U}^{\not\perp}_m$, when applied to T, transforms any encryption scheme that is DS as well as IND-CPA into an IND-CCA secure KEM.

## 3.1 Modularization of TPunc

We modularize transformation TPunc ("Puncturing and Encrypt-with-Hash") that was given in [SXY18], and that turns any IND-CPA secure PKE scheme into a deterministic one that is DS. Note that apart from reencryption, $\mathsf{TPunc}[\mathsf{PKE}_0, \mathsf{G}]$ given in [SXY18] and our modularization $\mathsf{T}[\mathsf{Punc}[\mathsf{PKE}_0], \mathsf{G}]$ are equal. In Section 3.1.1, we show that puncturing turns any IND-CPA secure scheme into a scheme that is both DS and IND-CPA, and in Section 3.1.2, we show that transformation T turns any scheme that is DS as well as IND-CPA secure into a deterministic scheme that is DS. Unfortunately, the latter security proof is nontight due to the use of the oneway-to-hiding lemma.

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

Transformation Punc turns any IND-CPA secure public-key encryption scheme with injective encryption into a DS secure one by puncturing the message space at one message and sampling encryptions of this message as fake encryptions. If $\mathsf{PKE}_0$'s encryption is injective, PKE is statistical disjoint with $\epsilon_{\text{dis}} = 0$.

THE CONSTRUCTION. To a public-key encryption scheme $\mathsf{PKE}_0 = (\mathsf{KG}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ with message space $\mathcal{M}_0$, we associate $\mathsf{PKE} := \mathsf{Punc}[\mathsf{PKE}_0, \hat{m}] := (\mathsf{KG} := \mathsf{KG}_0, \mathsf{Enc}, \mathsf{Dec} := \mathsf{Dec}_0)$ with message space $\mathcal{M} := \mathcal{M}_0 \setminus \{\hat{m}\}$ for some message $\hat{m} \in \mathcal{M}$. Encryption and fake encryption sampling of PKE are defined in Figure 6.

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

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

The following lemma states that IND-CPA security of $\mathsf{PKE}_0$ implies DS security of PKE.

**Lemma 3.1** (DS security of PKE). *If $\mathsf{PKE}_0$ is $\delta$-correct, so is PKE. For all adversaries A, there exists an IND-CPA adversary B such that*

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

*Furthermore, PKE is $\epsilon_{dis}$-statistical disjoint with*

$$\epsilon_{dis} \leq \mathbf{E}[\Pr_{\hat{r} \leftarrow_\$ \mathcal{R}}[\exists\, (m, r) \in \mathcal{M}_0 \times \mathcal{R} \ s.th. \ \mathsf{Enc}_0(pk, \hat{m}; \hat{r}) = \mathsf{Enc}_0(pk, m; r)]] \ ,$$

*where the expectation is taken over $(pk, sk) \leftarrow \mathsf{KG}$. In particular, if $\mathsf{Enc}_0(pk, -; -) : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$ is injective for all public keys $pk$, PKE is statistical disjoint with $\epsilon_{dis} = 0$.*

*Proof.* Let $\mathsf{A}$ be a $\mathsf{DS}$ adversary against $\mathsf{PKE}$. Consider the games given in Figure 7.

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

| Game $G$ | $\mathsf{B}_1(pk)$ |
|---|---|
| $pk \leftarrow \mathsf{KG}_0$ | $m \leftarrow_\$ \mathcal{M}_0 \setminus \{\hat{m}\}$ |
| $m \leftarrow_\$ \mathcal{M}_0 \setminus \{\hat{m}\}$ | **return** $(m, \hat{m})$ |
| $b \leftarrow_\$ \mathbb{F}_2$ | |
| $c_0 \leftarrow \mathsf{Enc}_0(pk, m)$ | $\mathsf{B}_2(c)$ |
| $c_1 \leftarrow \mathsf{Enc}_0(pk, \hat{m})$ | $b' \leftarrow \mathsf{A}(pk, c)$ |
| $b' \leftarrow \mathsf{A}(pk, c_b)$ | **return** $b'$ |
| **return** $[\![b' = b]\!]$ | |

Figure 7: Game $G$ and $\mathsf{IND\text{-}CPA}$ adversary $\mathsf{B} = (\mathsf{B}_1, \mathsf{B}_2)$ for the proof of Lemma 3.1.

Consider the $\mathsf{IND\text{-}CPA}$ adversary $\mathsf{B} := (\mathsf{B}_1, \mathsf{B}_2)$ also given in Figure 7. Since $\mathsf{B}$ perfectly simulates game $G$,

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

The following lemma states that $\mathsf{IND\text{-}CPA}$ security of $\mathsf{PKE}_0$ translates to $\mathsf{IND\text{-}CPA}$ security of $\mathsf{PKE}$. Its proof is straightforward.

**Lemma 3.2** ($\mathsf{IND\text{-}CPA}$ security of $\mathsf{PKE}$). *For all* $\mathsf{IND\text{-}CPA}$ *adversaries* $\mathsf{A}$ *there exists an adversary* $\mathsf{B}$ *such that*

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

### 3.1.2 Transformation $\mathsf{T}$: From probabilistic to deterministic $\mathsf{DS}$ security

Transformation $\mathsf{T}$ [BBO07] turns any probabilistic public-key encryption scheme into a deterministic one. The transformed scheme is $\mathsf{DS}$, given that $\mathsf{PKE}$ is $\mathsf{DS}$ as well as $\mathsf{IND\text{-}CPA}$ secure. Our security proof is tighter than the proof given for $\mathsf{TPunc}$ (see [SXY18, Theorem 3.3]) due to our use of the semi-classical O2H theorem.

The Construction. Take an encryption scheme $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$. Assume $\mathsf{PKE}$ to be additionally endowed with a sampling algorithm $\overline{\mathsf{Enc}}$ (see Definition 2.3). To $\mathsf{PKE}$ and random oracle $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, we associate $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, where the algorithms of $\mathsf{PKE}' = (\mathsf{KG}' := \mathsf{KG}, \mathsf{Enc}', \mathsf{Dec}', \overline{\mathsf{Enc}}' := \overline{\mathsf{Enc}})$ are defined in Figure 8. Note that $\mathsf{Enc}'$ deterministically computes the ciphertext as $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$.

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

Figure 8: Deterministic encryption scheme $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$.

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

**Lemma 3.3** ($\mathsf{DS}$ security of $\mathsf{PKE}'$). *If* $\mathsf{PKE}$ *is $\epsilon$-disjoint, so is* $\mathsf{PKE}'$. *For all adversaries* $\mathsf{A}$ *issuing at* *most* $q_{\mathsf{G}}$ *queries to* $|\mathsf{G}\rangle$*, there exist an adversary* $\mathsf{B}_{\mathsf{IND}}$ *and an adversary* $\mathsf{B}_{\mathsf{DS}}$ *such that*

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

*and the running time of each adversary is about that of* $\mathsf{B}$.

*Proof.* It is straightforward to prove disjointness since $\mathsf{Enc}'(pk, \mathcal{M})$ is subset of $\mathsf{Enc}(pk, \mathcal{M}; \mathcal{R})$. Let $\mathsf{A}$ be a $\mathsf{DS}$ adversary against $\mathsf{PKE}'$. Consider the sequence of games given in Figure 9. Per definition,

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

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

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

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

Figure 9: Games $G_0$ - $G_5$ for the proof of Lemma 3.3.

To upper bound $|\Pr[G_0^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]|$, consider adversary $\mathsf{B}_{\mathsf{DS}}$ against the disjoint simulatability of the underlying scheme $\mathsf{PKE}$, given in Figure 10. $\mathsf{B}_{\mathsf{DS}}$ runs in the time that is required to run $\mathsf{A}$ and to simulate $\mathsf{G}$ for $q_{\mathsf{G}}$ queries. Since $\mathsf{B}_{\mathsf{DS}}$ perfectly simulates game $G_0$ if run with a fake ciphertext as input, and game $G_3$ if run with a random encryption $c \leftarrow \mathsf{Enc}(pk, m^*)$,

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

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

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

```
B_DS(pk, c)
b' <- A^{|G>}(pk, c)
return b'

B_IND,1(pk)
m* <-$ M
return (0, m*, st := m*)

B_IND,2(pk, c*, st := m*)
FIND := false
b' <- A^{|G\{m*}>}(pk, c*)
return FIND

|G \ {m*}>|ψ, φ>
|ψ', b> := O^{SC}_{{m*}} |ψ, 0>
if b = 1
    FIND := true
return |G>|ψ', φ>
```

Figure 10: Adversaries $\mathsf{B}_{\mathsf{DS}}$ and $\mathsf{B}_{\mathsf{IND}\text{-}}$ for the proof of Lemma 3.3.

GAME $G_2$. In game $G_2$, we replace oracle access to $|\mathsf{G}\rangle$ with oracle acess to $|\mathsf{H}\rangle$ in line 08, where $\mathsf{H}$ is defined as follows: we pick a uniformly random $r^*$ in line 05 and let $\mathsf{H}(m) := \mathsf{G}(m)$ for all $m \neq m^*$, and $\mathsf{H}(m^*) := r^*$. Since $\mathsf{G}$ is a random oracle, this change is purely conceptual and

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

GAME $G_3$. In game $G_3$, we switch back to oracle access to $|\mathsf{G}\rangle$. Applying Theorem 2.5 for $S := \{m^*\}$, and $z := (pk, c^* := \mathsf{Enc}(pk, m^*; r^*))$, we obtain

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

GAME $G_5$. In game $G_5$, $c^* \leftarrow \mathsf{Enc}(pk, m^*)$ is replaced with an encryption of 0. Since in game $G_5$, $(pk, c^*)$ is independent of $m^*$, we can apply Theorem 2.6 to obtain

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

To upper bound $|\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - \Pr[G_5^{\mathsf{A}} \Rightarrow 1]|$, consider adversary $\mathsf{B}_{\mathsf{IND}}$ against the IND-CPA security of PKE, also given in Figure 10. $\mathsf{B}_{\mathsf{IND}}$ runs in the time that is required to run A and to measure and simulate G for $q_{\mathsf{G}}$ queries. $\mathsf{B}_{\mathsf{IND}}$ perfectly simulates game $G_4$ if run in game $\mathsf{IND\text{-}CPA}_0$ and game $G_5$ if run in game $\mathsf{IND\text{-}CPA}_1$, therefore,

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

Collecting the probabilities yields

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

□

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

Transformation SXY [SXY18] got rid of the additional hash (sometimes called key confirmation) that was included in [HHK17]'s quantum transformation $\mathsf{QU}_m^{\not\perp}$. SXY is essentially the (classical) transformation $\mathsf{U}_m^{\not\perp}$ that was also given in [HHK17], and apart from doing without the additional hash, it comes with a tight security reduction in the QROM. SXY differs from the (classical) transformation $\mathsf{U}_m^{\not\perp}$ only in the regard that it reencrypts during decapsulation. (In [HHK17], reencryption is done during decryption of T.) The security proof given in [SXY18] 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. What we show instead is that the combined transformation $\mathsf{FO}_m^{\not\perp} = \mathsf{U}_m^{\not\perp}[\mathsf{T}[-, \mathsf{G}], \mathsf{H}]$ turns any encryption scheme that is DS as well as IND-CPA into a KEM that is IND-CCA secure in the QROM, even if the underlying encryption scheme comes with a small probability of decryption failure. This is achieved by modifying random oracle G during the proof such that the encryption scheme is rendered perfectly correct. Our reduction is tighter as the (combined) reduction in [SXY18] due to our tighter security proof for T (see Section 3.1.2).

THE CONSTRUCTION. To $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, and random oracles $\mathsf{H} : \mathcal{M} \to \mathcal{K}$, $\mathsf{G} : \mathcal{M} \to \mathcal{R}$, and an additional internal random oracle $\mathsf{H_r} : \mathcal{C} \to \mathcal{K}$ that can not be directly accessed, we associate $\mathsf{KEM} = \mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] := \mathsf{U}_m^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$, where the algorithms of $\mathsf{KEM} = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps})$ are given in Figure 11.

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

Figure 11: Key encapsulation mechanism $\mathsf{KEM} = \mathsf{FO}_m^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] = \mathsf{U}_m^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$. Oracle $\mathsf{H_r}$ is used to generate random values whenever reencryption fails. This strategy is called implicit reject. Amongst others, it is used in [HHK17], [SXY18], and [JZC+18a]. For simplicity of the proof, $\mathsf{H_r}$ is modelled as an internal random oracle that cannot be accessed directly. For implementation, it would be sufficient to use a PRF.

Security. The following theorem (whose proof is essentially the same as in [SXY18] 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 3.4** (PKE DS+IND-CPA $\overset{\text{QROM}}{\Rightarrow}$ KEM IND-CCA)**.** *Assume* PKE *to come with injective encryption and a fake sampling algorithm* $\overline{\mathsf{Enc}}$ *such that* PKE *is* $\epsilon_{dis}$*-disjoint. Then, for any (quantum)* IND-CCA *adversary* A *issuing at most* $q_D$ *(classical) queries to the decapsulation oracle* Decaps*, at most* $q_{\mathsf{H}}$ *quantum queries to* $|\mathsf{H}\rangle$*, and at most* $q_{\mathsf{G}}$ *quantum queries to* $|\mathsf{G}\rangle$*, there exist (quantum) adversaries* $\mathsf{B}_{\mathsf{DS}}$ *and* $\mathsf{B}_{\mathsf{IND}}$ *such that*

$$\mathrm{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \leq 16 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 2q_D + 1)^2 \cdot \delta + \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{DS}}(\mathsf{B}_{\mathsf{DS}})$$
$$+ 2 \cdot \sqrt{(q_{\mathsf{G}} + q_{\mathsf{H}}) \cdot \mathrm{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}(\mathsf{B}_{\mathsf{IND}}) + \frac{4(q_{\mathsf{G}} + q_{\mathsf{H}})^2}{|\mathcal{M}|}} + \epsilon_{dis} \ ,$$

*and the running time of* $\mathsf{B}_{\mathsf{DS}}$ *and* $\mathsf{B}_{\mathsf{IND}}$ *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 $q_{\mathsf{H}}$ queries to the quantum random oracle $|\mathsf{H}\rangle$, and at most $q_{\mathsf{G}}$ queries to the quantum random oracle $|\mathsf{G}\rangle$. Consider the sequence of games given in Figure 12.

**GAMES** $G_0$ - $G_4$

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

Decaps$(c \neq c^*)$ // $G_0$ - $G_1$

16 $m' := \mathsf{Dec}(sk, c)$
17 **if** $m' = \perp$
  **or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
18   **return** $K := \mathsf{H_r}(c)$
19 **else**
20   **return** $K := \mathsf{H}(m')$ // $G_0$
21   **return** $K := \mathsf{H_q}(c)$ // $G_1$

Decaps$(c \neq c^*)$ // $G_2$ - $G_4$

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

Figure 12: Games $G_0$ - $G_4$ for the proof of Theorem 3.4.

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

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

Game $G_1$. In game $G_1$, we prepare getting rid of the secret key by plugging in encryption into random oracle H: Instead of drawing $\mathsf{H} \leftarrow_\$ \mathcal{K}^{\mathcal{M}}$, we draw $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}}$ in line 03 and define $\mathsf{H} := \mathsf{H_q}(\mathsf{Enc}(pk, -; \mathsf{G}(-)))$ in line 04. For consistency, we also change key $K_0^*$ in line 11 from letting $K_0^* := \mathsf{H}(m^*)$ to letting $K_0^* := \mathsf{H_q}(c^*)$, which is a purely conceptual change since $c^* = \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$. Additionally, we make the change of H explicit in oracle Decaps, i.e., we change oracle Decaps in line 21 such that it returns $K := \mathsf{H_q}(c)$ whenever $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) = c$. Since we assume $\mathsf{Enc}(pk, -; -)$ to be injective, H still is uniformly random, and since we only change Decaps for ciphertexts $c$ where $c = \mathsf{Enc}(pk, m'; \mathsf{G}(m'))$, we maintain consistency of H and Decaps. Hence, A's view is identical in both games and

$$\Pr[G_1^{\mathsf{A}} \Rightarrow 1] = \Pr[G_0^{\mathsf{A}} \Rightarrow 1] \ .$$

Game $G_2$. In game $G_2$, we change oracle Decaps such that it always returns $K := \mathsf{H_q}(c)$, as opposed to returning $\mathsf{H_q}(c)$ only if $c = \mathsf{Enc}(pk, \mathsf{Dec}(sk, c); \mathsf{G}(\mathsf{Dec}(sk, c)))$, and otherwise returning $\mathsf{H_r}(c)$. We claim

$$|\Pr[G_2^{\mathsf{A}} \Rightarrow 1] - \Pr[G_1^{\mathsf{A}} \Rightarrow 1]| \leq 8 \cdot (q_{\mathsf{G}} + q_{\mathsf{H}} + 2q_D + 1)^2 \cdot \delta \ .$$

**GAMES** $G_1$ - $G_2$

01 $(pk, sk) \leftarrow \mathsf{KG}$
02 $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ // $G_1$, $G_2$
03 Pick $2q$-wise hash $f$ // $G_{1^1/_3}$ - $G_{1^2/_3}$
04 $\mathsf{G} := \mathsf{G}_{pk,sk}$ // $G_{1^1/_3}$ - $G_{1^2/_3}$
05 $\mathsf{H_r} \leftarrow_\$ \mathcal{K}^{\mathcal{C}}$
06 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}}$
07 $\mathsf{H} := \mathsf{H_q}(\mathsf{Enc}(pk, -; \mathsf{G}(-)))$
08 $b \leftarrow_\$ \mathbb{F}_2$
09 $m^* \leftarrow \mathcal{M}$
10 $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$
11 $K_0^* := \mathsf{H_q}(c^*)$
12 $K_1^* \leftarrow_\$ \mathcal{K}$
13 $b' \leftarrow \mathsf{A}^{\textsc{Decaps}, |\mathsf{H}\rangle, |\mathsf{G}\rangle}(pk, c^*, K_b^*)$
14 **return** $[\![b' = b]\!]$

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

15 $r := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
16 **return** $r$

$\textsc{Decaps}(c \neq c^*)$ // $G_1$ - $G_{1^1/_3}$

17 $m' := \mathsf{Dec}'(sk, c)$
18 **if** $m' = \bot$
**or** $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$
19 **return** $K := \mathsf{H_r}(c)$
20 **else return** $K := \mathsf{H_q}(c)$

$\textsc{Decaps}(c \neq c^*)$ // $G_{1^2/_3}$ - $G_2$

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

Figure 13: Intermediate games $G_1$ - $G_2$ for the proof of Theorem 3.4 that deal with correctness errors. $f$ (lines 03 and 15) is an internal $2q$-wise independent hash function, where $q := q_\mathsf{G} + q_\mathsf{H} + 2 \cdot q_D + 1$, that cannot be accessed by $\mathsf{A}$. $\mathsf{Sample}(Y)$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_\$ Y$. $\mathsf{Sample}(Y; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(Y)$ using explicitly given randomness $f(m)$.

To verify this upper bound, consider the sequence of games given in Figure 13.

Game $G_{1^1/_3}$. In game $G_{1^1/_3}$, we enforce that no decryption failure will occur: For fixed $(pk, sk)$ and message $m \in \mathcal{M}$, let

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

denote the set of "bad" randomness. We replace random oracle $\mathsf{G}$ in line 04 with $\mathsf{G}_{pk,sk}$ that only samples from good randomness. Further, define

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

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

To upper bound $|\Pr[G^{\mathsf{A}}_{1^1/_3} \Rightarrow 1] - \Pr[G^{\mathsf{A}}_1 \Rightarrow 1]|$, we construct an (unbounded, quantum) adversary $\mathsf{B}$ against the generic distinguishing problem with bounded probabilities $\mathsf{GDPB}$ (see Lemma 2.7) in Figure 14, issuing $q_\mathsf{G} + q_\mathsf{H} + 2 \cdot q_D + 1$ queries to $|\mathsf{F}\rangle$. $\mathsf{B}$ draws a key pair $(pk, sk) \leftarrow \mathsf{KG}$ and computes the parameters $\lambda(m)$ of the generic distinguishing problem as $\lambda(m) := \delta(pk, sk, m)$, which are bounded by $\lambda := \delta(pk, sk)$. To analyze $\mathsf{B}$, we first fix $(pk, sk)$. For each $m \in \mathcal{M}$, by the definition of game $\mathsf{GDPB}_{\lambda,1}$, the random variable $\mathsf{F}(m)$ is bernoulli-distributed according to $B_{\lambda(m)} = B_{\delta(pk,sk,m)}$. By construction, the random variable $\mathsf{G}(m)$ defined in line 06 if $\mathsf{F}(m) = 0$ and in line 08 if $\mathsf{F}(m) = 1$ is uniformly distributed in $\mathcal{R}$, therefore $\mathsf{G}$ is a (quantum) random oracle and $\mathsf{A}^{|\mathsf{F}\rangle}$ perfectly simulates game $G_1$ if executed in game $\mathsf{GDPB}_{\lambda,1}$. Since $\mathsf{A}^{|\mathsf{F}\rangle}$ also perfectly simulates game $G_{1^1/_3}$ if executed in game $\mathsf{GDPB}_{\lambda,0}$,

$$|\Pr[G^{\mathsf{A}}_{1^1/_3} \Rightarrow 1] - \Pr[G^{\mathsf{A}}_1 \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{A}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{A}}_{\lambda,0} = 1]| \ ,$$

and according to Lemma 2.7,

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

Game $G_{1^2/_3}$. In game $G_{1^2/_3}$, we change oracle $\textsc{Decaps}$ such that it always returns $K := \mathsf{H_q}(c)$ (instead of returning $K := \mathsf{H_r}(c)$ if $m' := \mathsf{Dec}(sk, c) = \bot$ or $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) \neq c$). This change does not affect $\mathsf{A}$'s view: If there exists no message $m$ such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, oracle $\textsc{Decaps}(c)$ returns a random value (that can not possibly correlate to any random oracle query to $|\mathsf{H}\rangle$) in both games, therefore $\textsc{Decaps}(c)$ is a random value independent of all other input to $\mathsf{A}$ in both games. But if there

```
B_1 = B'_1
01 (pk, sk) ← KG
02 for m ∈ M
03     λ(m) := δ(pk, sk, m)
04 return (λ(m))_{m∈M}

B_2^{|H_r⟩,|H_q⟩,|F⟩} = B'_2^{|H_r⟩,|H_q⟩,|F⟩}
05 Pick 2q-wise hash f
06 H := H_q(Enc(pk, −; G(−)))
07 b ←$ F_2
08 m* ← M
09 c* := Enc(pk, m*; G(m*))
10 K*_0 := H_q(c*)
11 K*_1 ←$ K
12 b' ← A^{DECAPS,|H⟩,|G⟩}(pk, c*, K*_b)
13 return [[b' = b]]

DECAPS(c ≠ c*)                                    // Adversary B
14 m' := Dec'(sk, c)
15 if m' = ⊥
   or Enc(pk, m'; G(m')) ≠ c
16     return K := H_r(c)
17 else return K := H_q(c)

DECAPS(c ≠ c*)                                    // Adversary B'
18 return K := H_q(c)

G(m)
19 if F(m) = 0
20     G(m) := Sample(R \ R_bad(pk, sk, m); f(m))
21 else G(m) := Sample(R_bad(pk, sk, m); f(m))
22 return G(m)
```

Figure 14: Adversaries $\mathsf{B}$ and $\mathsf{B}'$ executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ with access to $|\mathsf{F}\rangle$ (and additional oracles $|\mathsf{H_r}\rangle$ and $|\mathsf{H_q}\rangle$) for the proof of Theorem 3.4. $\delta(pk, sk, m)$ is defined in Equation (7). $f$ (lines 06 and 08) is an internal $2q$-wise independent hash function, where $q := q_\mathsf{G} + q_\mathsf{H} + 2 \cdot q_D + 1$, that cannot be accessed by $\mathsf{A}$. Note that $\mathsf{B}$ and $\mathsf{B}'$ only differ in their simulation of the decapsulation oracle.

exists some message $m$ such that $c = \mathsf{Enc}(pk, m; \mathsf{G}(m))$, $\textsc{Decaps}(c)$ always returns $\mathsf{H_q}(c)$ in both games: Since $\mathsf{G}(m) \in \mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m)$ for all messages $m$, it holds that $m' := \mathsf{Dec}(sk, c) = m \neq \bot$ and that $\mathsf{Enc}(pk, m'; \mathsf{G}(m')) = c$. Hence $\mathsf{A}$'s view is identical in both games and

$$\Pr[G_{1^2/3}^{\mathsf{A}} \Rightarrow 1] = \Pr[G_{1^1/3}^{\mathsf{A}} \Rightarrow 1] \ .$$

$\textsc{Game}$ $G_2$. In game $G_2$, we switch back to using $\mathsf{G} \leftarrow_\$ \mathcal{R}^\mathcal{M}$ instead of $\mathsf{G}_{pk,sk}$. With the same reasoning as for the gamehop from game $G_1$ to $G_{1^1/3}$,

$$\begin{aligned}|\Pr[G_2^{\mathsf{A}} \Rightarrow 1] - \Pr[G_{1^2/3}^{\mathsf{A}} \Rightarrow 1]| &= |\Pr[\mathsf{GDPB}_{\lambda,1}^{\mathsf{B}'} = 1] - \Pr[\mathsf{GDPB}_{\lambda,0}^{\mathsf{B}'} = 1]| \\ &\leq 8 \cdot (q_\mathsf{G} + q_\mathsf{H} + 2q_D + 1)^2 \cdot \delta \ ,\end{aligned}$$

where adversary $\mathsf{B}'$ is also given in Figure 14.

So far, we established

$$\mathrm{Adv}_{\mathsf{KEM}}^{\mathsf{IND\text{-}CCA}}(\mathsf{A}) \leq |\Pr[G_2^{\mathsf{A}} \Rightarrow 1] - 1/2| + 16 \cdot (q_\mathsf{G} + q_\mathsf{H} + 2q_D + 1)^2 \cdot \delta \ .$$

The rest of the proof proceeds similiar to the proof in [SXY18], aside from the fact that we consider the particular scheme $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ instead of a generic $\mathsf{DS}$ deterministic encryption scheme.

$\textsc{Game}$ $G_3$. In game $G_3$, the challenge ciphertext $c^*$ gets decoupled from message $m^*$ by sampling $c^* \leftarrow \overline{\mathsf{Enc}}(pk)$ in line 09 instead of letting $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$. Consider the adversary $\mathsf{C_{DS}}$ against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ given in Figure 15. Since $\mathsf{C_{DS}}$ perfectly simulates game $G_2$ if run with deterministic encryption $c^* := \mathsf{Enc}(pk, m^*; \mathsf{G}(m^*))$ of a random message $m^*$, and game $G_3$ if run with a fake ciphertext,

$$|\Pr[G_3^{\mathsf{A}} \Rightarrow 1] - \Pr[G_2^{\mathsf{A}} \Rightarrow 1]| = \mathrm{Adv}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}^{\mathsf{DS}}(\mathsf{C_{DS}}), \ ,$$

and according to Lemma 3.3, there exist an adversary $\mathsf{B_{DS}}$ and an adversary $\mathsf{B_{IND}}$ with roughly the same running time such that

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

| $\mathsf{C_{DS}}^{\lvert\mathsf{G}\rangle,\lvert\mathsf{H_r}\rangle\lvert\mathsf{H_q}\rangle}(pk, c^*)$ | $\underline{\textsc{Decaps}(c \neq c^*)}$ |
|---|---|
| $b \leftarrow_\$ \mathbb{F}_2$ | **return** $K := \mathsf{H_q}(c)$ |
| $K_0^* := \mathsf{H_q}(c^*)$ | |
| $K_1^* \leftarrow_\$ \mathcal{K}$ | |
| $b' \leftarrow \mathsf{A}^{\textsc{Decaps},\lvert\mathsf{H}\rangle,\lvert\mathsf{G}\rangle}(pk, c^*, K_b^*)$ | |
| **return** $[\![b' = b]\!]$ | |

Figure 15: Adversary $\mathsf{C_{DS}}$ (with access to additional oracles $\lvert\mathsf{H_r}\rangle$ and $\lvert\mathsf{H_q}\rangle$) against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE},\mathsf{G}]$ for the proof of Theorem 3.4.

GAME $G_4$. In game $G_4$, the game is changed in line 12 such that it always uses a randomly picked challenge key. Since both $K_0^*$ and $K_1^*$ are independent of all other input to $\mathsf{A}$ in game $G_4$,

$$\Pr[G_4^{\mathsf{A}} \Rightarrow 1] = 1/2 \ .$$

It remains to upper bound $|\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]|$. To this end, it is sufficient to upper bound the probability that any of the queries to $\lvert\mathsf{H_q}\rangle$ could possibly contain $c^*$. Each query to $\lvert\mathsf{H_q}\rangle$ is either a classical query triggered by a query to DECAPS on some ciphertext $c$ or triggered by a query to $\lvert\mathsf{H}\rangle$ on a superposition $\lvert m\rangle$. Since queries to DECAPS on $c^*$ are explicitely forbidden, the only possibility would be a a query to $\lvert\mathsf{H_q}\rangle$ of the form $\sum_m \lvert\mathsf{Enc}(pk, m; \mathsf{G}(m))\rangle$. This query cannot contain $c^*$ unless there exists some message $m$ such that $\mathsf{Enc}(pk, m; \mathsf{G}(m)) = c^*$, and since we assume $\mathsf{PKE}$ to be $\epsilon_{\text{dis}}$-disjoint,

$$|\Pr[G_4^{\mathsf{A}} \Rightarrow 1] - \Pr[G_3^{\mathsf{A}} \Rightarrow 1]| \leq \epsilon_{\text{dis}} \ .$$

# 4 Two-Message Authenticated Key Exchange

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

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

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

| Party $\mathsf{P}_i$ $(pk_i, sk_i)$ | | Party $\mathsf{P}_j$ $(pk_j, sk_j)$ |
|---|---|---|
| $(M, \text{st}) \leftarrow \mathsf{Init}(sk_i, pk_j)$ | $\xrightarrow{\quad M \quad}$ | |
| | | $(M', K') \leftarrow \mathsf{Der_{resp}}(sk_j, pk_i, M)$ |
| $K := \mathsf{Der_{init}}(sk_i, pk_j, M', \text{st})$ | $\xleftarrow{\quad M' \quad}$ | |

Note that in contrast to the holder $\mathsf{P}_i$, the peer $\mathsf{P}_j$ will not be required to save any (secret) state information besides the key $K'$. Keys can be derived immediately after receiving the initiator's message.

Correctness. We call a two-message key exchange protocol $\mathsf{AKE}$ $\delta$-*correct* if

$$\Pr[(pk_i, sk_i) \leftarrow \mathsf{KG}, (pk_j, sk_j) \leftarrow \mathsf{KG}, (M, \mathrm{st}) \leftarrow \mathsf{Init}(sk_i, pk_j), \\ (M', K') \leftarrow \mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M), K := \mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \mathrm{st}) : K \neq K'] \leq \delta \ .$$

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

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

In our security model, the adversary $A$ is given black-box access to the set of processes $\mathsf{Init}$, $\mathsf{Der}_{\mathsf{resp}}$ and $\mathsf{Der}_{\mathsf{init}}$ that execute the AKE algorithms. To model the attacker's control of the network, we allow $\mathsf{A}$ to establish new sessions via EST, to call either INIT and $\mathrm{DER}_{\mathrm{init}}$ or $\mathrm{DER}_{\mathrm{resp}}$, each at most once per session (see Figure 16, page 20) and to relay their outputs faithfully as well as modifying the data on transit. Moreover, the attacker is additionally granted queries to reveal both secret process data, namely using REVEAL and REV-STATE queries, and parties' secret keys using CORRUPT queries, see Figure 17, page 21. 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 $\mathsf{IND\text{-}AA}_b$ and $\mathsf{IND\text{-}StAA}_b$ for $b \in \mathbb{F}_2$ as in Figure 16 and Figure 17. We define the $\mathsf{IND\text{-}AA}$ *advantage function of an adversary* $\mathsf{A}$ *against* $\mathsf{AKE}$ as

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

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

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

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

We say that a completed session sID *was recreated* iff there exists a session $\mathrm{sID}' \neq \mathrm{sID}$ such that $(\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}]) = (\mathrm{holder}[\mathrm{sID}'], \mathrm{peer}[\mathrm{sID}'])$, $\mathrm{role}[\mathrm{sID}] = \mathrm{role}[\mathrm{sID}']$, $\mathrm{sent}[\mathrm{sID}] = \mathrm{sent}[\mathrm{sID}']$, $\mathrm{received}[\mathrm{sID}] = \mathrm{received}[\mathrm{sID}']$ and $\mathrm{state}[\mathrm{sID}] = \mathrm{state}[\mathrm{sID}']$.

We say that two completed sessions $\mathrm{sID}_1$ and $\mathrm{sID}_2$ *match* iff $(\mathrm{holder}[\mathrm{sID}_1], \mathrm{peer}[\mathrm{sID}_1]) = (\mathrm{peer}[\mathrm{sID}_2], \mathrm{holder}[\mathrm{sID}_2])$, $(\mathrm{sent}[\mathrm{sID}_1], \mathrm{received}[\mathrm{sID}_1]) = (\mathrm{received}[\mathrm{sID}_2], \mathrm{sent}[\mathrm{sID}_2])$, and $\mathrm{role}[\mathrm{sID}_1] \neq \mathrm{role}[\mathrm{sID}_2]$.

We say that $\mathsf{A}$ *tampered with the test session* $\mathrm{sID}^*$ if at the end of the security game, there exists no matching session for $\mathrm{sID}^*$.

Helper procedure TRIVIAL (Figure 17) is used in all games to exclude the possibility of trivial attacks, and helper procedure ATTACK (also Figure 17) is defined in games $\mathsf{IND\text{-}StAA}_b$ to exclude the possibility of trivial attacks as well as one nontrivial attack that we will discuss below. During execution of TRIVIAL, the game creates list $\mathfrak{M}(\mathrm{sID}^*)$ of all matching sessions that were executed throughout the game (see line 56), and $\mathsf{A}$'s output bit $b'$ only counts in games $\mathsf{IND\text{-}AA}_b$ only if TRIVIAL returns false, i.e., if test session $\mathrm{sID}^*$ was completed and all of the following conditions hold:

1. $\mathsf{A}$ did not obtain the key of $\mathrm{sID}^*$ by querying REVEAL on $\mathrm{sID}^*$ or any matching session, see lines 50 and 57.

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

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

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

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

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

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

Figure 16: Games $\mathsf{IND\text{-}AA}_b$ and $\mathsf{IND\text{-}StAA}_b$ for $\mathsf{AKE}$, where $b \in \mathbb{F}_2$, and the collection of oracles O used in lines 06 and 28 is defined as O := {EST, INIT, $\mathrm{DER}_{\mathrm{resp}}$, $\mathrm{DER}_{\mathrm{init}}$, REVEAL, REV-STATE, CORRUPT, TEST}. Oracles REVEAL, REV-STATE, CORRUPT, and TEST are given in Figure 17. Note that $\mathsf{IND\text{-}StAA}_b$ only differs from $\mathsf{IND\text{-}AA}_b$ in ruling out one more kind of attack: To rule out attacks, we introduce helper methods TRIVIAL and ATTACK in Figure 17. A's bit $b'$ does not count in games $\mathsf{IND\text{-}AA}_b$ if helper procedure TRIVIAL returns **true**, see line 07. In games $\mathsf{IND\text{-}StAA}_b$, A's bit $b'$ does not count already if procedure ATTACK (that includes TRIVIAL and additionally checks for state-attacks on the test session) returns **true**, see line 29.

2. A did not obtain both the holder $i$'s secret key $sk_i$ and the test session's internal state, see line 52. We enforce that $\neg$corrupted[$i$] or $\neg$revState[sID*] since otherwise, A is allowed to obtain all information required to trivially compute $\mathsf{Der}(sk_i, pk_j, \text{received}[\text{sID}^*], \text{state}[\text{sID}^*])$.

3. A did not obtain both the peer's secret key $sk_j$ and the internal state of any matching session, see line 59. We enforce that $\neg$corrupted[$j$] or $\neg$revState[sID] for all sID s. th. sID $\in \mathfrak{M}(\text{sID}^*)$ for the same reason as discussed in 2: A could trivially compute $\mathsf{Der}(sk_j, pk_i, \text{received}[\text{sID}], \text{state}[\text{sID}])$ 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 62. We enforce that $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ or $\neg$corrupted[$j$] to exclude the following trivial attack: A could learn the peer's secret key $sk_j$ via query CORRUPT[$j$] and either
   - receive a message $M$ by querying INIT on sID*, compute $(M', K') \leftarrow \mathsf{Der}_{\mathsf{resp}}(sk_j, pk_i, M)$ without having to call $\mathrm{DER}_{\mathrm{resp}}$, and call $\mathrm{DER}_{\mathrm{init}}(\text{sID}^*, M')$, thereby ensuring that sKey[sID*] $= K'$,
   - or compute $(M, \text{st}) \leftarrow \mathsf{Init}(sk_j, pk_i)$ without having to call INIT, receive a message $M'$ by querying $\mathrm{DER}_{\mathrm{resp}}(\text{sID}^*, M)$, and trivially compute $\mathsf{Der}_{\mathsf{init}}(sk_j, pk_i, M', \text{st})$.

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

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

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

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

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

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

Figure 17: Helper procedures TRIVIAL and ATTACK and oracles REVEAL, REV-STATE, CORRUPT, and TEST of games IND-AA and IND-StAA defined in Figure 16.

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 65. We enforce that $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ or $\neg\text{revState}[\text{sID}^*]$ in game IND-StAA for the following reason: In an active attack, given that the test session's internal state got leaked, it is possible to choose a message $M'$ such that the result of algorithm $\mathsf{Der}_{\mathsf{init}}(sk_i, pk_j, M', \text{st})$ can be computed . For some protocols, this attack is possible even without knowledge of any of the static secret keys. In this setting, an adversary might query INIT on sID$^*$, learn the internal state st by querying REV-STATE on sID$^*$, choose its own message $M'$ without a call to DER$_{\text{resp}}$ and finally call DER$_{\mathsf{init}}$(sID$^*$, $M'$), thereby being enabled to anticipate the resulting key.

# 5 Transformation from PKE to AKE

Transformation $\mathsf{FO}_{\mathsf{AKE}}$ constructs a IND-StAA-secure AKE protocol from a PKE scheme that is both DS and IND-CPA secure.

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

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

The algorithms of AKE are defined in Figure 18.

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

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

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

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

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

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

Figure 18: IND-StAA secure AKE protocol $\mathsf{AKE} = \mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$. Oracles $\mathsf{H}'_{\mathsf{R}}$ and $\mathsf{H}'_{\mathsf{L1}}$, $\mathsf{H}'_{\mathsf{L2}}$ and $\mathsf{H}'_{\mathsf{L3}}$ are used to generate random values whenever reencryption fails. (For encryption, this strategy is called *implicit reject* Amongst others, it is used in [HHK17], [SXY18] and [JZC+18a].) For simplicity of the proof, $\mathsf{H}'_{\mathsf{R}}$ and $\mathsf{H}'_{\mathsf{L1}}$, $\mathsf{H}'_{\mathsf{L2}}$ and $\mathsf{H}'_{\mathsf{L3}}$ are internal random oracles that cannot be accessed directly. For implementation, it would be sufficient to use a PRF.

Security from DS. The following theorem establishes that IND-StAA security of AKE (see Definition 4.1) reduces to DS and IND-CPA security of PKE (see Definition 2.3 and Lemma 3.3).

**Theorem 5.1** (PKE DS + IND-CPA ⇒ AKE IND-StAA)**.** *Assume* PKE *to be injective. Furthermore, assume* PKE *to come with a sampling algorithm* $\overline{\mathsf{Enc}}$ *such that it is* $\epsilon$*-disjoint. Then, for any* IND-StAA *adversary* B *that establishes* $S$ *sessions and issues at most* $q_{\mathsf{R}}$ *(classical) queries to* REVEAL*, at most* $q_{\mathsf{G}}$ *(quantum) queries to random oracle* G *and at most* $q_{\mathsf{H}}$ *(quantum) queries to random oracle* H*, there exists an adversary* $\mathsf{A}_{\mathsf{DS}}$ *against the disjoint simulatability of* $\mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ *issuing at most* $q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S$ *queries to* G *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{B}) \leq\ & 16S^2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}_{\mathsf{DS}}) + 128 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\ & + 4S^2 \cdot \left(\epsilon_{dis} + \frac{S}{|\mathcal{M}|}\right) + 2S^2 \cdot \gamma(\mathsf{KG}) \ ,\end{aligned}$$

*and the running time of* $\mathsf{A}_{\mathsf{DS}}$ *is about that of* B*, and due to Lemma 3.3, there exist adversaries* $\mathsf{C}_{\mathsf{DS}}$ *and* $\mathsf{C}_{\mathsf{IND}}$ *such that*

$$\begin{aligned}\mathrm{Adv}^{\mathsf{IND\text{-}StAA}}_{\mathsf{AKE}}(\mathsf{B}) \leq 16S^2 \cdot & \left(\mathrm{Adv}^{\mathsf{DS}}_{\mathsf{PKE}}(\mathsf{C}_{\mathsf{DS}}) + 2 \cdot \sqrt{(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S) \cdot \mathrm{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathsf{C}_{\mathsf{IND}})}\right) \\ & + 128 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta + \frac{4S^2 \cdot (16(q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 + 1)}{\sqrt{|\mathcal{M}|}} \\ & + 4S^2 \cdot \epsilon_{dis} + 2S^2 \cdot \gamma(\mathsf{KG}) \ ,\end{aligned}$$

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

Proof Sketch. To prove IND-StAA security of $\mathsf{FO}_{\mathsf{AKE}}[\mathsf{PKE},\mathsf{G},\mathsf{H}]$, we consider an adversary B with black-box access to the protocols' algorithms and to oracles that reveal keys of completed sessions, internal states, and long-term secret keys of participating parties as specified in Figure 16. Intuitively, B will always be able to obtain all-but-one of the three secret messages $m_i$, $m_j$ and $\tilde{m}$ that are picked during execution of the test session between $\mathsf{P}_i$ and $\mathsf{P}_j$:

1. We first consider the case that B executed the test session honestly. Note that on the right-hand side of the protocol there exists no state. We assume that B has learned the secret key of party $\mathsf{P}_j$ and hence knows $m_j$. Additionally, B could either learn the secret key of party $\mathsf{P}_i$ 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 [SXY18] to simulate the decryption oracles by patching encryption into the random oracle H. In order to extend their technique to PKE schemes with non-perfect correctness, during the security proof we also need to patch random oracle G in a way that $(\mathsf{Enc}', \mathsf{Dec}')$ (relative to the patched G) provides perfect correctness. This strategy is the AKE analogue to the technique used in our analysis of the Fujisaki-Okamoto transformation given in Section 3, in particular, during our proof of Theorem 3.4.

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 $q_{\mathsf{R}}$ (classical) queries to REVEAL, at most $q_{\mathsf{G}}$ (quantum) queries to random oracle G and at most $q_{\mathsf{H}}$ (quantum) queries to random oracle H. We will first examine the case that B executed the test session honestly (i.e., the case that $\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$, where $\mathfrak{M}(\mathrm{sID}^*)$ is defined in Figure 17 , line 56, 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 $\mathfrak{M}(\mathrm{sID}^*) = \varnothing$).

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

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

$$\begin{aligned}
&|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(sID^*) \neq \varnothing]| \\
&\qquad \leq 4S^2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}) + 64 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\
&\qquad\quad + 2S^2 \cdot \left(\epsilon_{dis} + \frac{N}{|\mathcal{M}|} + \gamma(\mathsf{KG})\right) \ ,
\end{aligned}$$

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

The upper bound is proven in appendix B. 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[PKE, G]'s disjoint simulatability).

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

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

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

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

Folding A and A′ into one adversary $\mathsf{A}_{\mathsf{DS}}$, and assuming that $N << S$, we obtain

$$
\begin{aligned}
&|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1]| \\
&\quad \leq 16S^2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T[PKE,G]}}(\mathsf{A}_{\mathsf{DS}}) + 128 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\
&\quad + 4S^2 \cdot \left(\epsilon_{\mathrm{dis}} + \frac{S}{|\mathcal{M}|}\right) + 2S^2 \cdot \gamma(\mathsf{KG}) \ .
\end{aligned}
$$

# References


[ABS14] Janaka Alawatugoda, Colin Boyd, and Douglas Stebila. Continuous after-the-fact leakage-resilient key exchange. In Willy Susilo and Yi Mu, editors, *ACISP 14: 19th Australasian Conference on Information Security and Privacy*, volume 8544 of *Lecture Notes in Computer Science*, pages 258–273, Wollongong, NSW, Australia, July 7–9, 2014. Springer, Heidelberg, Germany. (Cited on page 1.)

[AHU18] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. Cryptology ePrint Archive, Report 2018/904, 2018. `http://eprint.iacr.org/2018/904`. (Cited on page 4, 9, 28.)

[AJOP18] Gorjan Alagic, Stacey Jeffery, Maris Ozols, and Alexander Poremba. On non-adaptive quantum chosen-ciphertext attacks and learning with errors. *CoRR*, abs/1808.09655, 2018. (Cited on page 9.)

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th Annual Symposium on Foundations of Computer Science*, pages 474–483, Philadelphia, PA, USA, October 18–21, 2014. IEEE Computer Society Press. (Cited on page 9, 10.)

[BBC+98] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *39th Annual Symposium on Foundations of Computer Science*, pages 352–361, Palo Alto, CA, USA, November 8–11, 1998. IEEE Computer Society Press. (Cited on page 8.)

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany. (Cited on page 3, 12.)

[BCK98] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. 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*, pages 419–428, Dallas, TX, USA, May 23–26, 1998. ACM Press. (Cited on page 1.)

[BCNP08] Colin Boyd, Yvonne Cliff, Juan Gonzalez Nieto, and Kenneth G. Paterson. 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.)

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Heidelberg, Germany. (Cited on page 2, 8.)

[BDK+17] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. 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, 6.)

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil P. Vadhan. Many-to-one trapdoor functions and their relation to public-key cryptosystems. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 283–298, Santa Barbara, CA, USA, August 23–27, 1998. Springer, Heidelberg, Germany. (Cited on page 3.)

[BI17] Subhadeep Banik and Takanori Isobe. Some cryptanalytic results on lizard. Cryptology ePrint Archive, Report 2017/346, 2017. http://eprint.iacr.org/2017/346. (Cited on page 6.)

[BJ15] Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 609–629, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany. (Cited on page 9.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 3.)

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Heidelberg, Germany. (Cited on page 1, 5.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Heidelberg, Germany. (Cited on page 7.)

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379, Santa Barbara, CA, USA, August 18–22, 2013. Springer, Heidelberg, Germany. (Cited on page 9.)

[CK01] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology – EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474, Innsbruck, Austria, May 6–10, 2001. Springer, Heidelberg, Germany. (Cited on page 1.)

[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. (Cited on page 3.)

[Den03] Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *Lecture Notes in Computer Science*, pages 133–151, Cirencester, UK, December 16–18, 2003. Springer, Heidelberg, Germany. (Cited on page 3.)

[DNR04] Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 342–360, Interlaken, Switzerland, May 2–6, 2004. Springer, Heidelberg, Germany. (Cited on page 2.)

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Heidelberg, Germany. (Cited on page 2, 3.)

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. (Cited on page 2, 3.)

[FSXY12] Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Conference on Theory and Practice of Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 467–484, Darmstadt, Germany, May 21–23, 2012. Springer, Heidelberg, Germany. (Cited on page 1, 2, 5.)

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. Semantic security and indistinguishability in the quantum world. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 60–89, Santa Barbara, CA, USA, August 14–18, 2016. Springer, Heidelberg, Germany. (Cited on page 9.)

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Heidelberg, Germany. (Cited on page 2, 3, 7, 14, 22, 28.)

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 9614 of *Lecture Notes in Computer Science*, pages 387–416, Taipei, Taiwan, March 6–9, 2016. Springer, Heidelberg, Germany. (Cited on page 9, 10, 11.)

[JKSS12] Tibor Jager, Florian Kohlar, Sven Schäge, and Jörg Schwenk. On the security of TLS-DHE in the standard model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 273–293, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany. (Cited on page 1.)

[JZC+18a] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125, Santa Barbara, CA, USA, August 19–23, 2018. Springer, Heidelberg, Germany. (Cited on page 3, 4, 14, 22, 28.)

[JZC+18b] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. 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 4, 28.)

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. (Cited on page 9.)

[Kra05] Hugo Krawczyk. HMQV: A high-performance secure Diffie-Hellman protocol. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 546–566, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Heidelberg, Germany. (Cited on page 1, 5, 6.)

[LLM07] Brian A. LaMacchia, Kristin Lauter, and Anton Mityagin. Stronger security of authenticated key exchange. In Willy Susilo, Joseph K. Liu, and Yi Mu, editors, *ProvSec 2007: 1st International Conference on Provable Security*, volume 4784 of *Lecture Notes in Computer Science*, pages 1–16, Wollongong, Australia, November 1–2, 2007. Springer, Heidelberg, Germany. (Cited on page 1, 5.)

[LS17] Yong Li and Sven Schäge. No-match attacks and robust partnering definitions: Defining trivial attacks for security protocols is not trivial. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17: 24th Conference on Computer and Communications Security*, pages 1343–1360, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. (Cited on page 1.)

[NAB+17] Michael Naehrig, Erdem Alkim, Joppe Bos, Leo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. Frodokem. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions. (Cited on page 6.)

[NIS17] NIST. National institute for standards and technology. postquantum crypto project, 2017. http://csrc.nist.gov/groups/ST/post-quantum-crypto/. (Cited on page 1.)

[Per12] Edoardo Persichetti. *Improving the efficiency of code-based cryptography*. PhD thesis, 2012. (Cited on page 3.)

[Sch15] Sven Schäge. TOPAS: 2-pass key exchange with full perfect forward secrecy and optimal communication complexity. In Indrajit Ray, Ninghui Li, and Christopher Kruegel:, editors, *ACM CCS 15: 22nd Conference on Computer and Communications Security*, pages 1224–1235, Denver, CO, USA, October 12–16, 2015. ACM Press. (Cited on page 1.)

[Sho04] Victor Shoup. 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 7.)

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part III*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Heidelberg, Germany. (Cited on page 2, 3, 4, 5, 6, 7, 9, 11, 12, 14, 15, 17, 22, 23, 28, 29.)

[Too15] Mohsen Toorani. On continuous after-the-fact leakage-resilient key exchange. In *Proceedings of the Second Workshop on Cryptography and Security in Computing Systems*, CS2 '15, pages 31:31–31:34, New York, NY, USA, 2015. ACM. (Cited on page 1.)

[Unr14] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 129–146, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany. (Cited on page 28.)

[YZ13] Andrew Chi-Chih Yao and Yunlei Zhao. OAKE: a new family of implicitly authenticated Diffie-Hellman protocols. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security*, pages 1113–1128, Berlin, Germany, November 4–8, 2013. ACM Press. (Cited on page 1.)

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Heidelberg, Germany. (Cited on page 9.)

# A Problems in and comparison with the proofs of [JZC+18a].

In this section we will discuss some problems we encountered in the proofs of [JZC+18a]. We refer to its current eprint version [JZC+18b]. Due to the structure of the non-modular proofs of [JZC+18b, Thms. 1 and 2], the original OW2H lemma [Unr14, Lem. 31: "One-way to hiding"] cannot be used to decouple the challenge plaintext from the adversary's view since random oracles $\mathsf{H}$ and $\mathsf{G}$ are not independent of each other. As a consequence, a new lemma called "One-way to hiding with redundant oracle" is introduced (see [JZC+18b, Lem. 3]). Unfortunately, the formal statement of Lemma 3 is unclear, in particular, the precise meaning of the independence requirement in [JZC+18b] is unclear and might be unsatisfiable,[8] rendering the proof impossible to verify.[9] During our proof, we circumvent this difficulty by following [SXY18]'s modular approach as far as we managed to: In [SXY18], the original OW2H lemma only needs to be applied for random oracle $\mathsf{G}$ (to prove that $\mathsf{PKE}'$ is deterministically $\mathsf{DS}$, as reflected in Figure 1). Once deterministic $\mathsf{DS}$ is achieved, oracle $\mathsf{H}$ does not have to be reprogrammed (instead, a fake encryption is sampled) and hence, OW2H does not have to be applied again.

To explain in which sense we followed the modular approach of [SXY18] *as far as we managed to*, we will point out some issues regarding the security claim for $\mathsf{SXY}$[10] [JZC+18b, Thm. 6] in an attempt to illustrate the difficulties in proving $\mathsf{SXY}$ secure if the underlying scheme comes with non-perfect correctness: [JZC+18b, Thm. 6] states that $\mathsf{SXY}$ turns any PKE scheme that is oneway-secure into a KEM that is $\mathsf{IND}$-$\mathsf{CCA}$ secure, with the correctness term $\delta$ being included into the upper bound as a summand $4q_E\sqrt{\delta}$, where $q_E$ is said to denote the number of queries to an encryption oracle.

The first drawback is that for deterministic schemes, the correctness term $\delta$ defined in [HHK17] and used in [JZC+18b, Thm. 6] reduces to the probability that for the sampled key pair, *at least one* message exists that inhibits decryption failure, i.e., the probability that the scheme is not perfectly correct for the sampled key pair. With this definition, the security statements given in the theorem are not meaningful for most lattice-based encryption schemes since in most cases, there exist some messages inducing decryption failure for each key pair, though this fraction might be small. Unfortunately, it is not straightforward to reasonably define correctness for deterministic encryption schemes such that it fits existing proof strategies, but also is being met by lattice-based schemes at the same time. We also would like to mention that the statement of [JZC+18b, Thm. 6], in the case where the underlying scheme is $\mathsf{DS}$, follows trivially (and with a better upper bound) from [SXY18, Thm. 4.2: "Security of $\mathsf{SXY}$ in the

[8] The requirement is that $x$ is uniformly distributed given $\mathcal{O}(x')$ for all $x' \neq x$. The formal meaning of this is hard to pin down, because the requirement says that $x$ is supposed to be uniform given a set of random variables (namely $\{\mathcal{O}(x')\}_{x' \neq x}$), where the choice which random variables are in that set depends in turn on $x$. But $x$ is a random variable itself and thus, it has no fixed value. We can formalize the requirement as "$x$ is uniform given $\mathcal{O}(x := \bot)$" (i.e., we remove the point $x$ from $\mathcal{O}$). But $x$ cannot be uniform given $\mathcal{O}(x := \bot)$ since $\mathcal{O}(x := \bot)$ determines $x$. So, the conditions in the O2H variant from [JZC+18b] may be unsatisfiable.

[9] While we cannot exclude the possibility that this issue could be resolved by applying [AHU18, Thm. 1: "Semi-classical O2H"], this approach would result in structurally different reductions and would require a stronger security assumption for the underlying scheme.

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

QROM"].[11]

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

The problems discussed above reflect why we weren't able to generalize [SXY18]'s modular analysis in a straightforward manner: In fact, we did not manage to define correctness for deterministic encryption schemes such that the definition bridges the gap between what is achievable by most lattice-based schemes and what is needed to fit existing proof strategies. This difficulty is solved by resorting to a non-modularized proof: What we show is that the KEM resulting from applying $\mathsf{FO}_m^{\not\perp} := \mathsf{U}_m^{\not\perp} \circ \mathsf{T}$ is IND-CCA secure in the QROM. To this end, we first prove that $\mathsf{T}[-, \mathsf{G}]$ turns any suitable scheme into a scheme that is deterministically DS, and then plug in this result into [SXY18]'s tight security proof. When plugging in $\mathsf{T}[-, \mathsf{G}]$ into $\mathsf{U}_m^{\not\perp}$, we can change random oracle G during the security proof such that the scheme is rendered perfectly correct, a necessary condition to proceed with the tight security proof. Distinguishing G from its "perfected" version allows for a reduction to a distinguishing problem. To generalize this strategy for *any* scheme, however, one would have to come up with a reduction that distinguishes access to an encryption oracle from access to an oracle that only answers with perfect encryptions, and as mentioned above, it might prove difficult to formalize this indistinguishability property in a meaningful manner such that it is compatible with the standard notion of IND-CCA security. We hope that our proofs achieve better auditability due to their at least somewhat more modular structure.

# B Proof of Lemma 5.2

Faithful execution of the protocol ($\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$). Recall that we are proving an upper bound for $|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) \neq \varnothing]|$. First, we will enforce that indeed, we only need to consider the case where $\mathfrak{M}(\mathrm{sID}^*) \neq \varnothing$, afterwards we ensure that exactly one matching session exists. Consider the sequence of games given in Figure 19.

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

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

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

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

Games $G_{2,b}$. Both games $G_{2,b}$ abort in line 08 if $|\mathfrak{M}(\mathrm{sID}^*)| > 1$, i.e., if more than one matching session exists. Due to the difference lemma,

$$|\Pr[G_{1,b}^{\mathsf{B}} \Rightarrow 1] - \Pr[G_{2,b}^{\mathsf{B}} \Rightarrow 1]| \leq \Pr[\text{Abort in line } 08]$$

for both bits $b$, and due to Lemma B.1 below,

$$\Pr[\text{Abort in line } 08] \leq \frac{S-1}{|\mathcal{M}|} \max\{\frac{1}{|\mathcal{M}|}, \gamma(\mathsf{KG})\} \leq \frac{S}{|\mathcal{M}|} \ .$$

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

**GAMES** $G_{0,b}$ - $G_{2,b}$

$\mathrm{sID}, \mathrm{sID}^* := 0$
**for** $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
**if** $\mathrm{ATTACK}(\mathrm{sID}^*)$
  **return** 0
**if** $\mathfrak{M}(\mathrm{sID}^*) = \varnothing$ ABORT // $G_{1,b}$
**if** $|\mathfrak{M}(\mathrm{sID}^*)| > 1$ ABORT // $G_{2,b}$
**return** $b'$

INIT(sID)

**if** $\mathrm{holder}[\mathrm{sID}] = \bot$
 **or** $\mathrm{sent}[\mathrm{sID}] \neq \bot$ **return** $\bot$
$\mathrm{role}[\mathrm{sID}] :=$ "initiator"
$i := \mathrm{holder}[\mathrm{sID}]$
$j := \mathrm{peer}[\mathrm{sID}]$
$m_j \leftarrow_\$ \mathcal{M}$
$c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
$(\tilde{pk}, \tilde{sk}) \leftarrow \mathsf{KG}$
$M := (\tilde{pk}, c_j)$
$\mathrm{state}[\mathrm{sID}] := (\tilde{sk}, m_j, M)$
$\mathrm{sent}[\mathrm{sID}] := M$
**return** $M$

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

**if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$
 **or** $\mathrm{role}[\mathrm{sID}] =$ "initiator" **return** $\bot$
$\mathrm{role}[\mathrm{sID}] :=$ "responder"
$(j, i) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}]$
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m_j' := \mathsf{Dec}(sk_j, c_j)$
**if** $m_j' = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m_j'; \mathsf{G}(m_j'))$
  $K' := \mathsf{H}_\mathsf{R}'(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
**else** $K' := \mathsf{H}(m_i, m_j', \tilde{m}, \tilde{pk}, i, j)$
$\mathrm{sKey}[\mathrm{sID}] := K'$
$(\mathrm{received}[\mathrm{sID}], \mathrm{sent}[\mathrm{sID}]) := (M, M')$
**return** $M'$

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

**if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{state}[\mathrm{sID}] = \bot$
 **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$ **return** $\bot$
$(i, j) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
$(\tilde{sk}, m_j, \tilde{pk}, c_j) := \mathrm{state}[\mathrm{sID}]$
$m_i' := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
**if** $m_i' = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m_i'; \mathsf{G}(m_i'))$
  **if** $\tilde{m}' = \bot$
    $K := \mathsf{H}_{\mathsf{L1}}'(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
  **else**
    $K := \mathsf{H}_{\mathsf{L2}}'(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
**else if** $\tilde{m}' = \bot$
    $K := \mathsf{H}_{\mathsf{L3}}'(m_i', m_j, \tilde{c}, \tilde{pk}, i, j)$
**else** $K := \mathsf{H}(m_i', m_j, \tilde{m}', \tilde{pk}, i, j)$
$\mathrm{sKey}[\mathrm{sID}] := K$
$\mathrm{received}[\mathrm{sID}] := M'$

Figure 19: Games $G_{0,b}$ - $G_{2,b}$ for case one of the proof of Theorem 5.1. Helper procedure ATTACK and oracles TEST, EST, CORRUPT, REVEAL and REV-STATE remains as in the original IND-StAA game (see Figures 16 and 17).

**Lemma B.1** *Assume* PKE *to be injective. Then, for any execution of* IND-StAA *in which* $S$ *sessions were established, the probability that a particular session* sID *was recreated is upper bounded by*

$$\frac{S-1}{|\mathcal{M}|} \cdot \begin{cases} \frac{1}{|\mathcal{M}|} & \mathrm{role}[\mathrm{sID}] = \text{"responder"} \\ \gamma(\mathsf{KG}) & \mathrm{role}[\mathrm{sID}] = \text{"initiator"} \end{cases} .$$

*Proof.* We first consider the case that $\mathrm{role}[\mathrm{sID}] =$ "responder": Let $j := \mathrm{holder}[\mathrm{sID}]$ and $i := \mathrm{peer}[\mathrm{sID}]$, let $(\tilde{pk}, c_j) := \mathrm{received}[\mathrm{sID}]$ and let $(c_i, \tilde{c}) := \mathrm{sent}[\mathrm{sID}]$, where $c_i := \mathsf{Enc}(pk_i, m_i, \mathsf{G}(m_i))$ and $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}, \mathsf{G}(\tilde{m}))$ for some messages $m_i$ and $\tilde{m}$ that were randomly drawn during execution of $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID})$

To recreate sID, B has to establish another session $\mathrm{sID}' \neq \mathrm{sID}$ with same holder and peer, and to call $\mathrm{DER}_{\mathrm{resp}}$ on $(\mathrm{sID}, (\tilde{pk}, c_j))$. After execution of $\mathrm{DER}_{\mathrm{resp}}$, we have that $\mathrm{sent}[\mathrm{sID}'] = (\mathsf{Enc}(pk_i, m_i', \mathsf{G}(m_i')), \mathsf{Enc}(\tilde{pk}, \tilde{m}', \mathsf{G}(\tilde{m}'))$ for some random messages $m_i'$ and $\tilde{m}'$. Since we assume $\mathsf{Enc}(pk, -; -)$ to be injective, $\mathrm{sent}[\mathrm{sID}] = \mathrm{sent}[\mathrm{sID}']$ iff $m_i = m_i'$ and $\tilde{m} = \tilde{m}'$, happening with probability at most $1/|\mathcal{M}|^2$.

Now we consider the case that $\mathrm{role}[\mathrm{sID}] =$ "initiator": Let $i := \mathrm{holder}[\mathrm{sID}]$ and $i := \mathrm{peer}[\mathrm{sID}]$, and let $(\tilde{sk}, m_j, \tilde{pk}, c_j) := \mathrm{st}[\mathrm{sID}]$ before execution of $\mathrm{DER}_{\mathrm{init}}(\mathrm{sID}, -)$. To recreate sID, B has to establish and initialize another session $\mathrm{sID}' \neq \mathrm{sID}$ with same holder and peer. Let $(\tilde{sk}', m_j', \tilde{pk}', c_j') := \mathrm{st}[\mathrm{sID}']$ before execution of $\mathrm{DER}_{\mathrm{init}}(\mathrm{sID}', -)$. $\mathrm{st}[\mathrm{sID}] = \mathrm{st}[\mathrm{sID}']$ iff $m_j = m_j'$ and $(\tilde{pk}, \tilde{sk}) = (\tilde{pk}', \tilde{sk}')$, happening with probability at most $\gamma(\mathsf{KG})/|\mathcal{M}|$. □

So far, we established

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

Since games $G_{2,b}$ abort unless $|\mathfrak{M}(\mathrm{sID}^*)| = 1$, we can treat the session ID of the matching session as unique from this point on and call it $\mathrm{sID}'$. Let $\mathrm{sID}^*_{\mathrm{init}}$ denote the initialising session, i.e., choose $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ such that $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] =$ "initiator", and let $\mathrm{sID}^*_{\mathrm{resp}}$ denote the other session. B's bit $b'$ only counts in $\mathsf{IND\text{-}StAA}_b$ (and also in $G_{2,b}$) if no trivial attack was executed: ATTACK returns **true** (and hence the game returns 0) if B did obtain both the initialising session's internal state and the secret key of its holder. We will therefore examine

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

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

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

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

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

The proof is given in in Appendix B.1. Its main idea is that since the initialising session's state (in particular, ephemeral secret key $\tilde{sk}^*$) remains unrevealed throughout the game, at least message $\tilde{m}^*$ (that was randomly picked by $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*_{\mathrm{resp}})$ cannot be computed trivially. By patching encryption into the random oracle at the argument where the ephemeral messages go in, we ensure that the game makes no use of $\tilde{sk}^*$ any longer. Since PKE is DS (and hence, so is T[PKE, G], see Lemma 3.3), we can decouple the test session's key from $\tilde{m}^*$ by replacing $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$, and changing the key accordingly. Given that PKE is $\epsilon_{\mathrm{dis}}$-disjoint, the probability that this fake ciphertext is a proper encryption can be upper bounded by $\epsilon_{\mathrm{dis}}$. Since the random oracle now comes with patched-in encryption, $\epsilon_{\mathrm{dis}}$ also serves as an upper bound for the probability that a random oracle query actually hits the session key. Hence the key is indistinguishable from a random key with overwhelming probability.

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

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

The proof of the upper bound is given in in Appendix B.2. Structurally, the proof is the same. It differs in the following way: while in case ($\neg$st), we made use of the fact that B does not obtain ephemeral secret key $\tilde{sk}^*$ and therefore, ciphertext $\tilde{c}$ was indistinguishable from a random fake encryption, in case ($\neg sk$), we can replace ciphertext $c_i$ (since $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]$ is not corrupted). In this setting, we need to patch in encryption at the first two arguments of the random oracle. Note that since B can execute many sessions defined relative to the secret key of $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]$, whereas in case ($\neg$st), the probability that ephemeral key pair $(\tilde{pk}^*, \tilde{sk}^*)$ was drawn in another session was negligibly small. Due to the adversary's capability to implicitly decrypt many encryptions relative to the secret key of $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]$, the proof gets more involved when dealing with correctness errors.

Collecting the probabilities, folding $\mathsf{A}_{\mathsf{DS}}^{\neg\text{st}}$ and $\mathsf{A}_{\mathsf{DS}}^{\neg sk}$ into one adversary $\mathsf{A}$, and assuming that $N << S << |\mathcal{M}|$, we obtain

$$\begin{aligned}
&|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) \neq \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\text{sID}^*) \neq \varnothing]| \\
&\quad \leq 4S^2 \cdot \text{Adv}_{\mathsf{T[PKE,G]}}^{\mathsf{DS}}(\mathsf{A}) + 64 \cdot N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 4S)^2 \cdot \delta \\
&\quad\quad + 2S^2 \cdot \left(\epsilon_{\text{dis}} + \frac{N}{|\mathcal{M}|} + \gamma(\mathsf{KG})\right) \ ,
\end{aligned}$$

the upper bound given in Lemma 5.2.

## B.1 Case $(\neg\text{st})$ of the Proof of Lemma 5.2

Case $(\neg\text{st})$ (Initialising session's state was not revealed). Consider the sequence of games given in Figures 20, 21 and 22: 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 $G_{9,b}^{\neg\text{st}}$, see Figure 20 and 21). Next, during execution of $\text{DER}_{\text{resp}}(\text{sID}_{\text{resp}}^*)$, we replace $\tilde{c} = \mathsf{Enc}(\tilde{pk}^*, \tilde{m}^*; \mathsf{G}(\tilde{m}^*))$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G_{10,b}^{\neg\text{st}}$ to $G_{11,b}^{\neg\text{st}}$, Figure 22, see line 28 ). We show that after those changes, $\mathsf{B}$'s view does not change with overwhelming probability if we change TEST such that it always returns a random value (game $G_{12,0}^{\neg\text{st}}$, also Figure 22).

**GAMES $G_{2,b}^{\neg\text{st}}$ - $G_{6,b}^{\neg\text{st}}$**

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

INIT(sID)

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

Figure 20: Games $G_{2,b}^{\neg\text{st}}$ - $G_{6,b}^{\neg\text{st}}$ for case $(\neg\text{st})$ of the proof of Lemma 5.2. Oracles $\text{DER}_{\text{resp}}$, $\text{DER}_{\text{init}}$ and TEST remain as in games $G_{0,b}^{\neg\text{st}}$ (see Figure 19, page 30), and helper procedure ATTACK and oracles EST, REVEAL and REV-STATE remain as in the original $\mathsf{IND\text{-}StAA}$ game (see Figure 16 and Figure 17, pages 20 and 21).

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 $b$,

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

Games $G_{3,b}^{\neg\text{st}}$. Both 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 $\Pr[G_{3,b}^{\mathsf{B}} \Rightarrow 1] = \Pr[G_{2,b}^{\mathsf{B}} \Rightarrow 1 \wedge \neg\text{st}]$,

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

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

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

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

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

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

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

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

So far, we established

$$|\Pr[G_{2,1}^{\mathsf{B}} \Rightarrow 1 \wedge \neg st] - \Pr[G_{2,0}^{\mathsf{B}} \Rightarrow 1 \wedge \neg st]| \leq S \cdot |\Pr[G_{6,1}^{\neg st\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{6,0}^{\neg st\,\mathsf{B}} \Rightarrow 1]| + 2S^2 \cdot \gamma(\mathsf{KG}) \ .$$

To upper bound $|\Pr[G_{6,1}^{\neg st\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{6,0}^{\neg st\,\mathsf{B}} \Rightarrow 1]|$, consider the sequence of games given in Figure 21.

To prepare getting rid of $\tilde{sk}^*$, we first change $\text{DER}_{\text{init}}$ such that whenever ciphertext $c_i$ induces decryption failure, $\tilde{sk}^*$ is not used anymore.

Games $G_{7,b}^{\neg st}$. In games $G_{7,b}^{\neg st}$, oracle $\text{DER}_{\text{init}}$ is changed in line 43 such that whenever $c_i$ fails to decrypt (i.e., $c_i$ does not decrypt to a message $m'_i$ s. th. $c_i = \mathsf{Enc}(pk_i, m'_i, \mathsf{G}(m'_i))$), the session key is always defined as $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$. (Before this change we let $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$ in the case that $c_i$ fails to decrypt, but $\tilde{c}$ decrypts correctly.) Since both $\mathsf{H}'_{\mathsf{L1}}$ and $\mathsf{H}'_{\mathsf{L2}}$ are not directly accessible and we assume $\mathsf{Enc}(\tilde{pk}, -)$ to be injective, $\mathsf{B}$'s view does not change and

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

The next preparation step is to rule out the possibility that the test session's ephemeral ciphertext fails to decrypt.

Game $G_{8,b}^{\neg st}$. In games $G_{8,b}^{\neg st}$, $\text{DER}_{\text{init}}(s'_{\text{init}}, (c_i, \tilde{c}))$ is changed such that it aborts in line 46 if $\tilde{c}$ does not decrypt to some message $\tilde{m}'$ such that $\tilde{c} = \mathsf{Enc}(\tilde{pk}^*, \tilde{m}'; \mathsf{G}(\tilde{m}'))$. Since the unique matching session $\text{sID}^*_{\text{resp}}$ exists, $\tilde{c}$ is the encryption of some message that was picked at random by $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}}, \text{sent}[\text{sID}^*_{\text{init}}])$ and

$$|\Pr[G_{7,b}^{\neg st} \Rightarrow 1] - \Pr[G_{8,b}^{\neg st} \Rightarrow 1]| \leq \delta \ .$$

We finally get rid of $\tilde{sk}^*$ by changing $\text{DER}_{\text{init}}$ for $s'_{\text{init}}$ such that if ciphertext $c_i$ decrypts correctly, the key is defined not using $\tilde{sk}^*$ anymore. This is achieved as follows: If ciphertext $c_i$ decrypts correctly, we do note use the decryption of $\tilde{c}$, but $\tilde{c}$ itself. To this end, we "patch in" encryption into random oracle $\mathsf{H}$ whenever ephemeral public key $\tilde{pk}^*$ is used. Due to the need for key consistency, we have to change $\text{DER}_{\text{resp}}$ accordingly.

Games $G_{9,b}^{\neg st}$. In game $G_{9,b}^{\neg st}$, random oracle $\mathsf{H}$ is changed as follows: Instead of picking $\mathsf{H}$ uniformly random, we pick two random oracles $\mathsf{H_q}$ and $\mathsf{H}'$ in lines 01 and 02, and define

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

see line 55. Since we assume $\mathsf{Enc}$ to be injective, $\mathsf{H}$ still is uniformly random.

We make the change of $\mathsf{H}$ explicit in the derivation oracles:

We change $\text{DER}_{\text{init}}$ in line 47 such that for $\text{sID} = s'_{\text{init}}$, the session key is defined as $K := \mathsf{H_q}(m'_i, m_j, \tilde{c}, \tilde{pk}^*, i, j)$, given that $c_i$ decrypts correctly. Since we enforced in game $G_{6,b}^{\neg st}$ that no other

**GAMES** $G_{6,b}^{\neg\text{st}}$ - $G_{9,b}^{\neg\text{st}}$

01 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$ // $G_{9,b}^{\neg\text{st}}$-$G_{9,b}^{\neg\text{st}}$
02 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{M}^2 \times \mathcal{C} \times \mathcal{PK} \times [N]^2}$ // $G_{9,b}^{\neg\text{st}}$-$G_{9,b}^{\neg\text{st}}$
03 $\text{cnt}, \text{sID}^* := 0$
04 $s'_{\text{init}} \leftarrow_\$ [S]$
05 **for** $n \in [N]$
06 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
07 $(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \mathsf{KG}$
08 $b' \leftarrow \mathsf{B}^{\text{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
09 **if** ATTACK$(\text{sID}^*)$
10 $\quad$ **return** 0
11 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
12 **if** revState$[\text{sID}^*_{\text{init}}]$ ABORT
13 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. role$[\text{sID}^*_{\text{init}}]$ = "initiator"
14 **if** $\text{sID}^*_{\text{init}} \neq s'_{\text{init}}$ **return** 0
15 **return** $b'$

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

16 **if** holder[sID] $= \bot$ **or** sKey[sID] $\neq \bot$ **or** role[sID] = "initiator" **return** $\bot$
17 role[sID] := "responder"
18 $(j, i) :=$ (holder[sID], peer[sID])
19 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
20 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
21 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
22 $M' := (c_i, \tilde{c})$
23 $m'_j := \mathsf{Dec}(sk_j, c_j)$
24 **if** $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
25 $\quad K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
26 **else**
27 $\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}, i, j)$
28 $\quad$ **if** $\tilde{pk} = \tilde{pk}^*$
29 $\quad\quad K' := \mathsf{H_q}(m_i, m'_j, \tilde{c}, \tilde{pk}, i, j)$ // $G_{9,b}^{\neg\text{st}}$-$G_{9,b}^{\neg\text{st}}$
30 sKey[sID] := $K'$
31 (received[sID], sent[sID]) := $(M, M')$
32 **return** $M'$

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

33 **if** holder[sID] $= \bot$ **or** state[sID] $= \bot$ **or** sKey[sID] $\neq \bot$ **return** $\bot$
34 $(i, j) :=$ (holder[sID], peer[sID])
35 $(\tilde{sk}, m_j, \tilde{pk}, c_j) :=$ state[sID]
36 $m'_i := \mathsf{Dec}(sk_i, c_i)$
37 $\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
38 **if** $m'_i = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
39 $\quad$ **if** $\tilde{m}' = \bot$
40 $\quad\quad K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
41 $\quad$ **else**
42 $\quad\quad K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$ // $G_{6,b}^{\neg\text{st}}$
43 $\quad\quad K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$ // $G_{7,b}^{\neg\text{st}}$-$G_{9,b}^{\neg\text{st}}$
44 **else if** $\text{sID} = s'_{\text{init}}$
45 $\quad$ **if** $\tilde{m}' = \bot$ **or** $\tilde{c} \neq \mathsf{Enc}(\tilde{pk}, \tilde{m}'; \mathsf{G}(\tilde{m}'))$
46 $\quad\quad$ ABORT // $G_{8,b}^{\neg\text{st}}$-$G_{9,b}^{\neg\text{st}}$
47 $\quad K := \mathsf{H_q}(m'_i, m_j, \tilde{c}, \tilde{pk}, i, j)$ // $G_{9,b}^{\neg\text{st}}$
48 **else if** $\tilde{m}' = \bot$
49 $\quad\quad K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
50 **else**
51 $\quad K := \mathsf{H}(m'_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
52 sKey[sID] := $K$
53 received[sID] := $M'$

$\mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j)$ // $G_{9,b}^{\neg\text{st}}$

54 **if** $\tilde{pk} = \tilde{pk}^*$
55 $\quad$ **return** $\mathsf{H_q}(m_1, m_2, \mathsf{Enc}(\tilde{pk}, m_3; \mathsf{G}(m_3)), \tilde{pk}, i, j)$
56 **return** $\mathsf{H}'(m_1, m_2, m_3, \tilde{pk}, i, j)$

Figure 21: Games $G_{6,b}^{\neg\text{st}}$ - $G_{9,b}^{\neg\text{st}}$ for case ($\neg$st) of the proof of Lemma 5.2. Oracle $\mathsf{Init}$ remains as in games $G_{4,b}^{\neg\text{st}}$ (see Figure 20, page 32), (see Figure 16, page 20), and helper procedure ATTACK and oracles TEST, EST, REVEAL and REV-STATE remain as in the original $\mathsf{IND\text{-}StAA}$ games.

session than $s'_{\text{init}}$ could possibly use public key $\tilde{pk}^*$, this indeed is the only session where we have to change the definition of $K$. Furthermore, we enforced in game $G_{8,b}^{\neg\text{st}}$ that $\tilde{c}$ decrypts correctly, i.e., we enforce that $\tilde{m}' := \mathsf{Dec}(\tilde{sk}^*, \tilde{c}) \neq \bot$ and that $\tilde{c} = \mathsf{Enc}(\tilde{pk}^*, \tilde{m}'; \mathsf{G}(\tilde{m}'))$, hence we have key consistency:

$$\mathsf{H}(m'_i, m_j, \tilde{m}', \tilde{pk}^*, i, j) = \mathsf{H_q}(m'_i, m_j, \mathsf{Enc}(\tilde{pk}^*, \tilde{m}'; \mathsf{G}(\tilde{m}')), \tilde{pk}^*, i, j)$$
$$= \mathsf{H_q}(m'_i, m_j, \tilde{c}, \tilde{pk}^*, i, j) \ .$$

Likewise, make the change of $\mathsf{H}$ explicit in $\text{DER}_{\text{resp}}$: we change $\text{DER}_{\text{resp}}$ in line 29 such that if $\tilde{pk} = \tilde{pk}^*$, the session keys are defined as $K' := \mathsf{H_q}(m_i, m'_j, \tilde{c}, \tilde{pk}^*, i, j)$ whenever $c_j$ decrypts correctly. This change is purely conceptual since $\tilde{c}$ is defined as $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$:

$$\mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}^*, i, j) = \mathsf{H_q}(m_i, m'_j, \mathsf{Enc}(\tilde{pk}^*, \tilde{m}; \mathsf{G}(\tilde{m})), \tilde{pk}, i, j) = \mathsf{H_q}(m_i, m'_j, \tilde{c}, \tilde{pk}^*, i, j) \ .$$

We conclude

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

So far, we established

$$|\Pr[G_{6,1}^{\neg\mathrm{st}} \Rightarrow 1] - \Pr[G_{6,0}^{\neg\mathrm{st}} \Rightarrow 1]| \leq |\Pr[G_{9,1}^{\neg\mathrm{st}} \Rightarrow 1] - \Pr[G_{9,0}^{\neg\mathrm{st}} \Rightarrow 1]| + 2 \cdot \delta \ ,$$

hence

$$|\Pr[G_{2,1}^{\mathsf{B}} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G_{2,0}^{\mathsf{B}} \Rightarrow 1 \wedge \neg\mathrm{st}]| \leq S \cdot |\Pr[G_{9,1}^{\neg\mathrm{st}} \Rightarrow 1] - \Pr[G_{9,0}^{\neg\mathrm{st}} \Rightarrow 1]| + 2S \cdot \delta + 2S^2 \cdot \gamma(\mathsf{KG}) \ .$$

We stress that from game $G_{9,b}^{\neg\mathrm{st}}$ on, none of the oracles use ephemeral secret key $\tilde{sk}^*$ any longer. To upper bound $|\Pr[G_{9,1}^{\neg\mathrm{st}\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{9,0}^{\neg\mathrm{st}\,\mathsf{B}} \Rightarrow 1]|$, consider the sequence of games given in Figure 22, where we replace $\mathrm{sID}^*_{\mathrm{resp}}$'s ciphertext $\tilde{c}$ with a fake encryption. To replace $\tilde{c}$, we first have to add a guess for $\mathrm{sID}^*_{\mathrm{resp}}$.

**GAMES** $G_{9,b}^{\neg\mathrm{st}}$ - $G_{12,b}^{\neg\mathrm{st}}$

01 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$
02 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{M}^2 \times \mathcal{C} \times \mathcal{PK} \times [N]^2}$
03 $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$
04 $\mathrm{cnt}, \mathrm{sID}^* := 0$
05 $s'_{\mathrm{init}} \leftarrow_\$ [S]$
06 $s'_{\mathrm{resp}} \leftarrow_\$ [S]$ // $G_{10,b}^{\neg\mathrm{st}}$ - $G_{12,b}^{\neg\mathrm{st}}$
07 **for** $n \in [N]$
08 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
09 $(\tilde{pk}^*, \tilde{sk}^*) \leftarrow \mathsf{KG}$
10 $b' \leftarrow \mathsf{B}^{\mathrm{O}, |\mathsf{G}\rangle, |\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
11 **if** ATTACK($\mathrm{sID}^*$)
12 $\quad$ **return** 0
13 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
14 **if** revState[$\mathrm{sID}^*_{\mathrm{init}}$] ABORT
15 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. role[$\mathrm{sID}^*_{\mathrm{init}}$] = "initiator"
16 **if** $\mathrm{sID}^*_{\mathrm{init}} \neq s'_{\mathrm{init}}$ **return** 0
17 Pick $\mathrm{sID}^*_{\mathrm{resp}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$ s. th. role[$\mathrm{sID}^*_{\mathrm{resp}}$] = "responder" // $G_{10,b}^{\neg\mathrm{st}}$-$G_{12,b}^{\neg\mathrm{st}}$
18 **if** $\mathrm{sID}^*_{\mathrm{resp}} \neq s'_{\mathrm{resp}}$
19 $\quad$ **return** 0 // $G_{10,b}^{\neg\mathrm{st}}$-$G_{12,b}^{\neg\mathrm{st}}$
20 **return** b'

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

21 **if** holder[sID] = $\perp$ **or** sKey[sID] $\neq \perp$ **or** role[sID] = "initiator" **return** $\perp$
22 role[sID] := "responder"
23 $(j, i)$ := (holder[sID], peer[sID]
24 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
25 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
26 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
27 **if** $\mathrm{sID} = s'_{\mathrm{resp}}$
28 $\quad \tilde{c} \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$ // $G_{11,b}^{\neg\mathrm{st}}$-$G_{12,b}^{\neg\mathrm{st}}$
29 $M' := (c_i, \tilde{c})$
30 $m'_j := \mathsf{Dec}(sk_j, c_j)$
31 **if** $m'_j = \perp$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
32 $\quad K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
33 **else**
34 $\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}, i, j)$
35 $\quad$ **if** $\tilde{pk} = \tilde{pk}^*$
36 $\quad\quad K' := \mathsf{H_q}(m_i, m'_j, \tilde{c}, \tilde{pk}, i, j)$
37 sKey[sID] := $K'$
38 (received[sID], sent[sID]) := $(M, M')$
39 **return** $M'$

TEST(sID) //only one query

40 $\mathrm{sID}^* := \mathrm{sID}$
41 **if** sKey[$\mathrm{sID}^*$] = $\perp$ **return** $\perp$
42 $K_0^* :=$ sKey[$\mathrm{sID}^*$] // $G_{9,b}^{\neg\mathrm{st}}$-$G_{11,b}^{\neg\mathrm{st}}$
43 $K_0^* \leftarrow_\$ \mathcal{K}$ // $G_{12,0}^{\neg\mathrm{st}}$
44 $K_1^* \leftarrow_\$ \mathcal{K}$
45 **return** $K_b^*$

Figure 22: Games $G_{9,b}^{\neg\mathrm{st}}$ - $G_{12,b}^{\neg\mathrm{st}}$ for case ($\neg$st) of the proof of Lemma 5.2. All oracles except for TEST and $\mathrm{DER}_{\mathrm{resp}}$ remain as in game $G_{9,b}^{\neg\mathrm{st}}$ (see Figure 21, page 34).

**Games** $G_{10,b}^{\neg\mathrm{st}}$. In game $G_{10,b}^{\neg\mathrm{st}}$, one of the sessions that get established during execution of $\mathsf{B}$ is picked at random in line 06, and the game returns 0 in line 19 if any other session $s'_{\mathrm{resp}}$ was picked than session $\mathrm{sID}^*_{\mathrm{resp}}$. Again,

$$\Pr[G_{9,b}^{\neg\mathrm{st}} \Rightarrow 1] = S \cdot \Pr[G_{10,b}^{\neg\mathrm{st}} \Rightarrow 1] \ .$$

Games $G^{\neg\mathrm{st}}_{11,b}$. In game $G^{\neg\mathrm{st}}_{11,b}$, $\mathrm{DER}_{\mathrm{resp}}$ is changed in line 28 such that for $s'_{\mathrm{resp}}$, $\tilde{c}$ is no longer an encryption of a randomly drawn message $\tilde{m}$, but a fake encryption $\tilde{c} \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$. Consider the adversaries $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}$ against the disjoint simulatability of $\mathsf{T}[\mathsf{PKE},\mathsf{G}]$ given in Figure 23. Each adversary $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}$ needs to generate ephemeral key pairs (at most $S$ times), to (deterministically) encrypt or reencrypt (at most $3S$ times), to decrypt (at most $2S$ times), to evaluate the random oracles $\mathsf{H_q}$ and $\mathsf{H}'$ (at most $q_{\mathsf{H}} + S$ times) as well as $\mathsf{G}$ (at most $q_{\mathsf{G}} + 3S$ times), and to lazy sample (at most $S$ times). Hence the total running time is upper bounded as follows:

$$\begin{aligned}\mathrm{Time}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}) &\leq \mathrm{Time}(\mathsf{B}) + S \cdot (\mathrm{Time}(\mathsf{KG}) + 3 \cdot \mathrm{Time}(\mathsf{Enc}) + 2 \cdot \mathrm{Time}(\mathsf{Dec})) + q_{\mathsf{H}} + q_{\mathsf{G}} + 4S \\ &\approx \mathrm{Time}(\mathsf{B}) \ . \end{aligned} \quad (5)$$

Since $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}$ perfectly simulates game $G^{\neg\mathrm{st}}_{10,b}$ if its input $c^*$ was generated by $c := \mathsf{Enc}(\tilde{pk}^*, m, \mathsf{G}(m))$ for some randomly picked message $m$, and game $G^{\neg\mathrm{st}}_{11,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(\tilde{pk}^*)$,

$$|\Pr[G^{\neg\mathrm{st}}_{10,b} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{11,b} \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}) \ ,$$

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

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

```
A_DS,b^¬st |H'⟩,|Hq⟩,|G⟩ (p̃k*, c*)
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]

DER_resp(sID, M = (p̃k, 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(p̃k, 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, p̃k, i, j)
29 else
30     K' := H(m_i, m'_j, m̃, p̃k, i, j)
31     if p̃k = p̃k*
32         K' := H_q(m_i, m'_j, c̃, p̃k, i, j)
33 sKey[sID] := K'
34 (received[sID], sent[sID]) := (M, M')
35 return M'
```

Figure 23: Adversaries $\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS,b}}$ for case ($\neg$st) of the proof of Lemma 5.2, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H_q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for $\mathrm{DER}_{\mathrm{resp}}$ and REV-STATE are defined as in game $G^{\neg\mathrm{st}}_{10,b}$ (see Figure 22, page 35). Note that the internal random oracles ($\mathsf{H}'_{\mathsf{R}}$, and $\mathsf{H}'_{\mathsf{L1}}$ to $\mathsf{H}'_{\mathsf{L3}}$) can be simulated via lazy sampling since they are only accessible indirectly via $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$, which are queried classically.

So far, we established

$$|\Pr[G^{\neg\mathrm{st}}_{9,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{9,0} \Rightarrow 1]| \leq S \cdot |\Pr[G^{\neg\mathrm{st}}_{11,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{11,0} \Rightarrow 1]| + 2S \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}) \ ,$$

hence

$$\begin{aligned}|\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg\mathrm{st}] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg\mathrm{st}]| \leq S^2 \cdot |\Pr[G^{\neg\mathrm{st}}_{11,1} \Rightarrow 1] - \Pr[G^{\neg\mathrm{st}}_{11,0} \Rightarrow 1]| \\ + 2S^2 \cdot \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T}[\mathsf{PKE},\mathsf{G}]}(\mathsf{A}^{\neg\mathrm{st}}_{\mathsf{DS}}) + 2S \cdot \delta + 2S^2 \cdot \gamma(\mathsf{KG}) \ .\end{aligned}$$

Game $G^{\neg\text{st}}_{12,0}$. In game $G^{\neg\text{st}}_{12,0}$, we change oracle TEST in line 43 such that it returns a random value instead of returning sKey[sID$^*$]. Since this change renders games $G^{\neg\text{st}}_{12,0}$ and $G^{\neg\text{st}}_{12,1}$ equal, and since game $G^{\neg\text{st}}_{12,1}$ is equal to game $G^{\neg\text{st}}_{11,1}$,

$$|\Pr[G^{\neg\text{st}}_{11,1} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{11,0} \Rightarrow 1]| = |\Pr[G^{\neg\text{st}}_{12,0} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{11,0} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G^{\neg\text{st}}_{12,0} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{11,0} \Rightarrow 1]|$. $\mathsf{B}$ cannot distinguish $K_0^* = \text{sKey}[\text{sID}^*]$ from random in game $G^{\neg\text{st}}_{11,0}$ unless it obtains $K_0^*$ (either classically or contained in a quantum answer) at some point other than during the calling of TEST. It's easy to verify that $\mathsf{B}$ can only obtain keys (and in particular, $K_0^*$) by queries to REVEAL or to $\mathsf{H}$.

Let $(i^*, j^*) := (\text{holder}[\text{sID}^*_{\text{init}}], \text{peer}[\text{sID}^*_{\text{init}}])$. $\tilde{pk}^*$ denotes the ephemeral key that was chosen in the beginning of the game (see Figure 20, line 05) and used during execution of INIT(sID$^*_{\text{init}}$) (line 25, also Figure 20). Let $m_j^*$ denote the randomly chosen message with encryption $c_j^* := \mathsf{Enc}(pk_{j^*}, m_j^*; \mathsf{G}(m_j^*))$ that was sampled during execution of INIT(sID$^*_{\text{init}}$), furthermore let $\tilde{c}^*$ denote the fake ciphertext that was sampled under $\tilde{pk}^*$ (Figure 22, line 28) and let $m_i^*$ denote the randomly chosen message with encryption $c_i^* := \mathsf{Enc}(pk_{i^*}, m_i^*; \mathsf{G}(m_i^*))$ that was picked during execution of $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}})$. We changed the key derivation such that since $\tilde{pk}^*$ is used (and $\mathsf{Enc}$ is injective), in the case that $\text{sID}^* = \text{sID}^*_{\text{init}}$, we have

$$K_0^* = \begin{cases} \mathsf{H}'_{\mathsf{L1}}(c_i^*, m_j^*, \tilde{c}^*, \tilde{pk}^*, i^*, j^*) & \mathsf{Dec}(sk_{i^*}, c_i^*) \neq m_i^* \\ \mathsf{H}_{\mathsf{q}}(m_i^*, m_j^*, \tilde{c}^*, \tilde{pk}^*, i^*, j^*) & \text{o.w.} \end{cases} \ ,$$

and in the case that $\text{sID}^* = \text{sID}^*_{\text{resp}}$, we have

$$K_0^* = \begin{cases} \mathsf{H}'_{\mathsf{R}}(m_i^*, c_j^*, \tilde{m}^*, \tilde{pk}^*, i^*, j^*) & \mathsf{Dec}(sk_{j^*}, c_j^*) \neq m_j^* \\ \mathsf{H}_{\mathsf{q}}(m_i^*, m_j^*, \tilde{c}^*, \tilde{pk}^*, i^*, j^*) & \text{o.w.} \end{cases} \ .$$

We claim that $\mathsf{B}$ obtains $K_0^*$ by a query to REVEAL with probability 0 if role[sID$^*$] = "initiator" and with probability at most $^{S-2}/_{|\mathcal{M}|^2} \cdot \delta$ if role[sID$^*$] = "responder":

Recall that $\mathsf{B}$ trivially loses if revealed[sID$^*_{\text{init}}$] or revealed[sID$^*_{\text{resp}}$], hence, to obtain $K_0^*$ (without losing trivially) via some query to REVEAL, $\mathsf{B}$ would have to derive the same session key by recreating the test session. (Creation of an additional matching session would result in an abort.) We first consider the case that $\text{sID}^* = \text{sID}^*_{\text{init}}$: To obtain $K_0^*$ via recreation, $\mathsf{B}$ would have to establish and initialize session $\text{sID} \neq \text{sID}^*_{\text{init}}$ with holder $i^*$ and peer $j^*$. INIT(sID) randomly picks some message $m_j$ and a key pair $(\tilde{pk}, \tilde{sk})$ and outputs $\tilde{pk}$ and $c_j := \mathsf{Enc}(pk_{j^*}, m_j; \mathsf{G}(m_j))$. The subsequent call to $\text{DER}_{\text{init}}$ only results in the same key if $m_j^* = m_j$ and $\tilde{pk} = \tilde{pk}^*$, which is impossible since we enforced in game $G^{\neg\text{st}}_{6,b}$ that no other session uses $\tilde{pk}^*$. Using the same reasoning, it is straightforward to argue that if $\text{sID}^* = \text{sID}^*_{\text{resp}}$, $\mathsf{B}$ can only obtain $K_0$ (without losing trivially) with probability at most $^{S-2}/_{|\mathcal{M}|^2} \cdot \delta$.

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

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

Hence, to trigger a query to $|\mathsf{H}_{\mathsf{q}}\rangle$ containing the classical query $(m_i^*, m_j^*, \tilde{c}^*, \tilde{pk}^*, i^*, j^*)$, $\mathsf{B}$ would need to come up with a message $m$ such that $\mathsf{Enc}(\tilde{pk}^*, m; \mathsf{G}(m)) = \tilde{c}^*$. Since $\tilde{c}^*$ was sampled by $\overline{\mathsf{Enc}}$ and $\mathsf{PKE}$ is $\epsilon_{\text{dis}}$-disjoint, this is possible with probability at most $\epsilon_{\text{dis}}$ and

$$|\Pr[G^{\neg\text{st}}_{11,0} \Rightarrow 1] - \Pr[G^{\neg\text{st}}_{12,0} \Rightarrow 1]| \leq \frac{S-2}{|\mathcal{M}|^2} \cdot \delta + \epsilon_{\text{dis}} \ \leq \ \frac{S}{|\mathcal{M}|^2} + \epsilon_{\text{dis}} \ .$$

Collecting the probabilities yields

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

the upper bound we claimed in equation (3).

## B.2 Case $(\neg sk)$ of the Proof of Lemma 5.2

Case $(\neg sk)$ (Initialising session's owner was not corrupted). Intuition is as follows: While $\mathsf{B}$ 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 24 and 27: First, we will enforce that indeed, we only need to consider the case where $\neg$corrupted[holder[$\text{sID}^*_{\text{init}}$]]. Afterwards, we ensure that the game makes no use of $sk_{i^*}$ any longer by patching encryption into the random oracle (in games $G^{\neg sk}_{2,b}$ to $G^{\neg sk}_{7,b}$, see Figure 24, line 35). This is the only part of the proof where we need to consider the adversary's capability to come up with encryptions that decrypt incorrectly. Next, during execution of $\text{DER}_{\text{resp}}(\text{sID}^*_{\text{resp}})$, we replace $c_i = \mathsf{Enc}(pk_{i^*}, m_i^*)$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G^{\neg sk}_{8,b}$ to $G^{\neg sk}_{9,b}$, see Figure 27). We show that after those changes, $\mathsf{B}$'s view does not change with overwhelming probability if we finally change TEST such that it always returns a random value (game $G^{\neg sk}_{10,b}$, also Figure 27).

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

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

Games $G^{\neg sk}_{3,b}$. Both games $G^{\neg sk}_{3,b}$ abort in line 14 if corrupted[holder[$\text{sID}^*_{\text{init}}$]]. Since for both bits $b$ it holds that $\Pr[{G^{\neg sk}_{3,b}}^{\mathsf{B}} \Rightarrow 1] = \Pr[{G^{\neg sk}_{2,b}}^{\mathsf{B}} \Rightarrow 1 \wedge \neg sk]$,

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

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

Games $G^{\neg sk}_{4,b}$. In both games $G^{\neg sk}_{4,b}$, one of the parties is picked at random in line 05, and the games return 0 in line 16 if any other party $i'$ was picked than the holder of $\text{sID}^*_{\text{init}}$.

Since for both bits $b$ it holds that games $G^{\neg sk}_{4,b}$ and $G^{\neg sk}_{3,b}$ proceed identically if holder[$\text{sID}^*_{\text{init}}$] $= i'$, and since games $G^{\neg sk}_{4,b}$ output 0 if holder[$\text{sID}^*_{\text{init}}$] $\neq i'$,

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

To prepare getting rid of $sk_{i'}$, we first change $\text{DER}_{\text{init}}$ such that whenever ciphertext $\tilde{c}$ induces decryption failure, $sk_{i'}$ is not used anymore.

Games $G^{\neg sk}_{5,b}$. In both games $G^{\neg sk}_{5,b}$, we change oracle $\text{DER}_{\text{init}}$ in line 34 such that whenever the session's holder is $i'$ and $\tilde{c}$ does not decrypt to a message $\tilde{m}'$ s. th. $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}', \mathsf{G}(\tilde{m}'))$, the session key is defined as $K := \mathsf{H}'_{\mathsf{L}1}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$. (Before this change we let $K := \mathsf{H}'_{\mathsf{L}3}(m_i', m_j, \tilde{c}, \tilde{pk}, i, j)$ in the case that $\tilde{c}$ fails to decrypt, but $c_i$ decrypts correctly.) Since both $\mathsf{H}'_{\mathsf{L}1}$ and $\mathsf{H}'_{\mathsf{L}3}$ are not directly accessible and we assume $\mathsf{Enc}(pk_{i'}, -)$ to be injective, $\mathsf{B}$'s view does not change and

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

The next two game-hops are done to achieve that $\text{DER}_{\text{init}}$ and $\text{DER}_{\text{resp}}$ do not use $sk_{i'}$ any more. In the next game, we only change key definition of $\text{DER}_{\text{init}}$ if both ciphertexts decrypt correctly, and key definition of $\text{DER}_{\text{resp}}$ if $c_j$ decrypts correctly. In these cases, we do note use the decryptions under $sk_{i'}$, but the ciphertexts themself. Similar to case ($\neg$st), we "patch in" encryption into random oracle $\mathsf{H}$ whenever $i'$ appears as one of the involved parties. Due to the need for key consistency, we have to change patch encryption into the first *two* arguments.

**GAMES** $G_{2,b}^{\neg sk}$ - $G_{7,b}^{\neg sk}$

$\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$ // $G_{6,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
$\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$ // $G_{6,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
$\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$
$\mathrm{cnt}, \mathrm{sID}^* := 0$
$i' \leftarrow_\$ [N]$ // $G_{4,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
**for** $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
**if** $\mathrm{ATTACK}(\mathrm{sID}^*)$
  **return** 0
**if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$
 s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] =$ "initiator" // $G_{3,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
**if** $\mathrm{corrupted}[\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]]$
  ABORT // $G_{3,b}^{\neg sk}$-$G_{6,b}^{\neg \mathrm{st}}$
**if** $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}] \neq i'$
  **return** 0 // $G_{4,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
**return** b'

$\mathrm{DER_{init}}(\mathrm{sID}, M' = (c_i, \tilde{c}))$

**if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{state}[\mathrm{sID}] = \bot$
 **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$ **return** $\bot$
$(i, j) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
$(\tilde{sk}, m_j, \tilde{pk}, c_j) := \mathrm{state}[\mathrm{sID}]$
$m_i' := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
**if** $m_i' = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m_i'; \mathsf{G}(m_i'))$
  **if** $\tilde{m}' = \bot$
    $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
  **else**
    $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $i = i'$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{7,b}^{\neg sk}$
**else**
  **if** $\tilde{m} = \bot$
    $K := \mathsf{H}'_{\mathsf{L3}}(m_i', m_j, \tilde{c}, \tilde{pk}, i, j)$
    **if** $i = i'$
      $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$ // $G_{5,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
  **else**
    $K := \mathsf{H}(m_i', m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $i' \in \{i, j\}$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{6,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
$\mathrm{sKey}[\mathrm{sID}] := K$
$\mathrm{received}[\mathrm{sID}] := M'$

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

**if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$
 **or** $\mathrm{role}[\mathrm{sID}] =$ "initiator" **return** $\bot$
$\mathrm{role}[\mathrm{sID}] :=$ "responder"
$(j, i) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m_j' := \mathsf{Dec}(sk_j, c_j)$
**if** $m_j' = \bot$
 **or** $c_j \neq \mathsf{Enc}(pk_j, m_j'; \mathsf{G}(m_j'))$
  $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
  **if** $j = i'$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{7,b}^{\neg sk}$
**else**
  $K' := \mathsf{H}(m_i, m_j', \tilde{m}, \tilde{pk}, i, j)$
  **if** $i' \in \{i, j\}$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{6,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$
$\mathrm{sKey}[\mathrm{sID}] := K'$
$(\mathrm{received}[\mathrm{sID}], \mathrm{sent}[\mathrm{sID}]) := (M, M')$
**return** $M'$

$\mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j)$ // $G_{6,b}^{\neg sk}$-$G_{7,b}^{\neg sk}$

**if** $i' \in \{i, j\}$
  **return** $\mathsf{H_q}(\mathsf{Enc}(pk_i, m_1; \mathsf{G}(m_1)), \mathsf{Enc}(pk_j, m_2; \mathsf{G}(m_2)), m_3, \tilde{pk}, i, j)$
**return** $\mathsf{H}'(m_1, m_2, m_3, \tilde{pk}, i, j)$

Figure 24: Games $G_{2,b}^{\neg sk}$ - $G_{7,b}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5.2. Helper procedure ATTACK and oracles TEST, Init, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 16 and Figure 17, pages 20 and 21).

Games $G_{6,b}^{\neg sk}$. In games $G_{6,b}^{\neg sk}$, the random oracle is changed as follows: Instead of picking H uniformly random, we pick two random oracles $\mathsf{H_q}$ and $\mathsf{H}'$ and define

$$\mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j) := \begin{cases} \mathsf{H_q}(\mathsf{Enc}(pk_i, m_1; \mathsf{G}(m_1)), \mathsf{Enc}(pk_j, m_2; \mathsf{G}(m_2)), m_3, \tilde{pk}, i, j) & i' \in \{i, j\} \\ \mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j) & \text{o.w.} \end{cases} ,$$

see line 61. Again, $\mathsf{H}$ still is uniformly random since we assume $\mathsf{Enc}(pk, -; -)$ to be injective.

We make the change of $\mathsf{H}$ explicit in oracles $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$: We change $\mathrm{DER}_{\mathrm{init}}$ in line 38 such that if the session's peer or holder is $i'$, the session key is defined as $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ whenever both $c_i$ and $\tilde{c}$ decrypt correctly. This change is purely conceptual since $c_i = \mathsf{Enc}(pk, m_i'; \mathsf{G}(m_i'))$ and $c_j = \mathsf{Enc}(pk, m_j; \mathsf{G}(m_j))$.

Likewise, we change oracle $\mathrm{DER}_{\mathrm{resp}}$ in line 56 such that if the session's peer or holder is $i'$, the session key is defined as $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ whenever $c_j$ decrypts correctly. Again, this change is purely conceptual, and

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

So far, we established

$$|\Pr[G_{2,1}^{\mathsf{B}} \Rightarrow 1 \wedge \neg sk] - \Pr[G_{2,0}^{\mathsf{B}} \Rightarrow 1 \wedge \neg sk]| = N \cdot |\Pr[G_{6,1}^{\neg sk} \Rightarrow 1] - \Pr[G_{6,0}^{\neg sk} \Rightarrow 1]| \ .$$

The final step to get rid of $sk_{i'}$ is to even out the key derivation for problematic ciphertexts: To this end, we also use $\mathsf{H_q}$ if a ciphertext fails to decrypt under $sk_{i'}$, instead of using the implicit reject.

Games $G_{7,b}^{\neg sk}$. In games $G_{7,b}^{\neg sk}$, we change $\mathrm{DER}_{\mathrm{resp}}$ in line 52 such that whenever the session's holder is $i'$ and $c_j$ fails to decrypt, the session key is defined as $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ instead of letting $K' := \mathsf{H}_{\mathsf{R}}'(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$.

Likewise, we change $\mathrm{DER}_{\mathrm{init}}$ in line 29 such that whenever the session's holder is $i'$ and ciphertext $\tilde{c}$ decrypts correctly, the session key is defined as $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$. (Before this change, we let $K := \mathsf{H}_{\mathsf{L2}}'(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$ if $\tilde{c}$ decrypts correctly, but ciphertext $c_i$ fails to decrypt.) We claim that for both bits $b$ it holds that

$$|\Pr[G_{6,b}^{\neg sk} \Rightarrow 1] - \Pr[G_{7,b}^{\neg sk} \Rightarrow 1]| \leq 16 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ . \tag{6}$$

To verify this upper bound, consider the sequence of intermediate games given in Figure 25. Intuitively, removing the implicit rejects can only affect $\mathsf{B}$'s view if keys were derived using error-inducing encryptions. We show that we can replace random oracle $\mathsf{G}$ with an oracle $\mathsf{G}_{pk_{i'}, sk_{i'}}$ that makes error-inducing encryptions impossible, while distinguishing $\mathsf{G}$ from $\mathsf{G}_{pk_{i'}, sk_{i'}}$ is reducable to winning $\mathsf{GDPB}$.

**GAMES** $G^{\neg sk}_{6,b}$ - $G^{\neg sk}_{7,b}$

01 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$
02 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$
03 $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ // $G^{\neg sk}_{6,b}$, $G^{\neg sk}_{7,b}$
04 Pick $2q$-wise hash $f$ // $G^{\neg sk}_{61/3,b}$-$G^{\neg sk}_{62/3,b}$
05 $\mathrm{cnt}, \mathrm{sID}^* := 0$
06 $i' \leftarrow_\$ [N]$
07 **for** $n \in [N]$
08 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
09 $\mathsf{G} := \mathsf{G}_{pk_{i'}, sk_{i'}}$ // $G^{\neg sk}_{61/3,b}$-$G^{\neg sk}_{62/3,b}$
10 $b' \leftarrow \mathsf{B}^{\mathrm{O}, |\mathsf{G}\rangle, |\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
11 **if** ATTACK(sID*)
12 $\quad$ **return** 0
13 **if** $|\mathfrak{M}(\mathrm{sID}^*)| \neq 1$ ABORT
14 Pick $\mathrm{sID}^*_{\mathrm{init}} \in \{\mathrm{sID}^*, \mathrm{sID}'\}$
$\quad$ s. th. $\mathrm{role}[\mathrm{sID}^*_{\mathrm{init}}] =$ "initiator"
15 **if** $\mathrm{corrupted}[\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}]]$ ABORT
16 **if** $\mathrm{holder}[\mathrm{sID}^*_{\mathrm{init}}] \neq i'$
17 $\quad$ **return** 0
18 **return** b'

$\mathsf{G}_{pk_{i'}, sk_{i'}}(m)$

19 $r := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk_{i'}, sk_{i'}, m); f(m))$
20 **return** $r$

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

21 **if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$
$\quad$ **or** $\mathrm{role}[\mathrm{sID}] =$ "initiator" **return** $\bot$
22 $\mathrm{role}[\mathrm{sID}] :=$ "responder"
23 $(j, i) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
24 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
25 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
26 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
27 $M' := (c_i, \tilde{c})$
28 $m'_j := \mathsf{Dec}(sk_j, c_j)$
29 **if** $m'_j = \bot$
$\quad$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
30 $\quad K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
31 $\quad$ **if** $j = i'$
32 $\quad\quad K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G^{\neg sk}_{62/3,b}$-$G^{\neg sk}_{7,b}$
33 **else**
34 $\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}, i, j)$
35 $\quad$ **if** $i' \in \{i, j\}$
36 $\quad\quad K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
37 $\mathrm{sKey}[\mathrm{sID}] := K'$
38 $(\mathrm{received}[\mathrm{sID}], \mathrm{sent}[\mathrm{sID}]) := (M, M')$
39 **return** $M'$

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

40 **if** $\mathrm{holder}[\mathrm{sID}] = \bot$ **or** $\mathrm{state}[\mathrm{sID}] = \bot$
$\quad$ **or** $\mathrm{sKey}[\mathrm{sID}] \neq \bot$ **return** $\bot$
41 $(i, j) := (\mathrm{holder}[\mathrm{sID}], \mathrm{peer}[\mathrm{sID}])$
42 $(\tilde{sk}, m_j, \tilde{pk}, c_j) := \mathrm{state}[\mathrm{sID}]$
43 $m'_i := \mathsf{Dec}(sk_i, c_i)$
44 $\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
45 **if** $m'_i = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
46 $\quad$ **if** $\tilde{m}' = \bot$
47 $\quad\quad K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
48 $\quad$ **else**
49 $\quad\quad K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
50 $\quad\quad$ **if** $i = i'$
51 $\quad\quad\quad K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G^{\neg sk}_{62/3,b}$-$G^{\neg sk}_{7,b}$
52 **else**
53 $\quad$ **if** $\tilde{m} = \bot$
54 $\quad\quad K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
55 $\quad\quad$ **if** $i = i'$
56 $\quad\quad\quad K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
57 $\quad$ **else**
58 $\quad\quad K := \mathsf{H}(m'_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
59 $\quad\quad$ **if** $i' \in \{i, j\}$
60 $\quad\quad\quad K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
61 $\mathrm{sKey}[\mathrm{sID}] := K$
62 $\mathrm{received}[\mathrm{sID} := M'$

Figure 25: Intermediate games $G^{\neg sk}_{6,b}$ - $G^{\neg sk}_{7,b}$ for case $(\neg sk)$ of the proof of Lemma 5.2. All oracles except for $\mathsf{G}$, $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$ remain as in game $G^{\neg sk}_{6,b}$. $f$ (lines 04 and 19) is an internal $2q$-wise independent hash function, where $q := q_{\mathsf{G}} + 2 \cdot q_{\mathsf{H}} + 3 \cdot S$, that cannot be accessed by $\mathsf{B}$. $\mathsf{Sample}(Y)$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_\$ Y$. $\mathsf{Sample}(Y; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(Y)$ using explicitly given randomness $f(m)$.

GAME $G^{\neg sk}_{61/3,b}$. In game $G^{\neg sk}_{61/3,b}$, we enforce that no decryption failure with respect to key pair $(pk_{i'}, sk_{i'})$ will occur by replacing random oracle $\mathsf{G}$ with $\mathsf{G}_{pk_{i'}, sk_{i'}}(m)$ in line 09, where $\mathsf{G}_{pk_{i'}, sk_{i'}}(m)$ is defined in line

19 by

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

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

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

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

To upper bound $|\Pr[G^{\neg sk}_{6^1/_3,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{6,b} \Rightarrow 1]|$ for each bit $b$, we construct (unbounded, quantum) adversaries $\mathsf{C^b}$ against the generic distinguishing problem with bounded probabilities $\mathsf{GDPB}_\lambda$ (see Lemma 2.7) in Figure 26, issuing at most $q_\mathsf{G} + 2q_\mathsf{H} + 3 \cdot S$ queries to $|\mathsf{F}\rangle$:

Each $\mathsf{C^b}$ runs $(pk, sk) \leftarrow \mathsf{KG}$ and uses this key pair as $(pk_{i'}, sk_{i'})$ when simulating game $G^{\neg sk}_{6,b}$ to $\mathsf{B}$. $\mathsf{C^b}$ computes the parameters $\lambda(m)$ of the generic distinguishing problem as $\lambda(m) := \delta(pk_{i'}, sk_{i'}, m)$, which are bounded by $\lambda := \delta(pk_{i'}, sk_{i'})$.

To analyze $\mathsf{C^b}$, we first fix $(pk_{i'}, sk_{i'})$. For each $m \in \mathcal{M}$, by the definition of game $\mathsf{GDPB}_{\lambda,1}$, the random variable $\mathsf{F}(m)$ is distributed according to $B_{\lambda(m)} = B_{\delta(pk_{i'},sk_{i'},m)}$. 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 $\mathsf{G}$ is a (quantum) random oracle and $\mathsf{C^b}$ perfectly simulates game $G^{\neg sk}_{6,b}$ if executed in game $\mathsf{GDPB}_{\lambda,1}$. Since adversary $\mathsf{C^b}$ also perfectly simulates game $G^{\neg sk}_{6^1/_3,b}$ if executed in game $\mathsf{GDPB}_{\lambda,0}$,

$$|\Pr[G^{\neg sk}_{6,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{6^1/_3,b} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}^{\mathsf{C^b}}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C^b}}_{\lambda,0} = 1]| \ ,$$

and according to Lemma 2.7,

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

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

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

$\underline{\text{CORRUPT}(i \in [N] \setminus \{i'\})}$
10 **if** corrupted$[i]$ **return** $\perp$
11 corrupted$[i]$ := **true**
12 **return** $sk_i$

$\underline{\mathsf{C^b}_2^{|\mathsf{F}\rangle}, \mathsf{C^{b'}}_2^{|\mathsf{F}\rangle}}$
13 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3\times\mathcal{PK}\times[N]^2}$
14 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2\times\mathcal{M}\times\mathcal{PK}\times[N]^2}$
15 Pick $2q$-wise hash $f$
16 cnt, sID* := 0
17 $i' \leftarrow_\$ [N]$
18 **for** $n \in [N] \setminus \{i'\}$
19 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
20 $(pk_{i'}, sk_{i'}) := (pk, sk)$
21 $b' \leftarrow \mathsf{B}^{\mathsf{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
22 **if** ATTACK(sID*)
23 $\quad$ **return** 0
24 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
25 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$
$\quad$ s. th. role$[\text{sID}^*_{\text{init}}]$ = "initiator"
26 **if** corrupted[holder$[\text{sID}^*_{\text{init}}]$] ABORT
27 **if** holder$[\text{sID}^*_{\text{init}}] \neq i'$
28 $\quad$ **return** 0
29 **return** b'

Figure 26: Adversaries $\mathsf{C^b} = (\mathsf{C^b}_1, \mathsf{C^b}_2)$ and $\mathsf{C^{b'}} = (\mathsf{C^{b'}}_1, \mathsf{C^{b'}}_2)$ for $b \in \mathbb{F}_2$ executed in game $\mathsf{GDPB}_{\delta(pk_{i'},sk_{i'})}$ with access to $|\mathsf{F}\rangle$, for case $(\neg sk)$ of the proof of Lemma 5.2. $\delta(pk_{i'}, sk_{i'})$ is defined in Equation (7). The adversaries only differ in their definition of $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$: For the adversaries $\mathsf{C^b}$, $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ are defined as in game $G^{\neg sk}_{6,b}$, see Figure 25, while for adversaries $\mathsf{C^{b'}}$, $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ are defined as in game $G^{\neg \text{st}}_{6^2/_3,b}$ (also Figure 25).

Games $G^{\neg sk}_{6^2/_3,b}$. In games $G^{\neg sk}_{6^2/_3,b}$, we change $\text{DER}_{\text{init}}$ in line 51 such that for holder $i'$, the session key is defined as $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ whenever ciphertext $\tilde{c}$ decrypts correctly. (Before this change, we let

$K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$ if $\tilde{c}$ decrypts correctly, but ciphertext $c_i$ fails to decrypt.) Likewise, we change $\mathrm{DER}_{\mathrm{resp}}$ in line 32 such that for holder $i'$, the session key is always defined as $K' := \mathsf{H}_{\mathsf{q}}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ instead of letting $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ if $c_j$ fails to decrypt.

We argue that this change does not affect $\mathsf{B}$'s view for both bits $b$: Let $(\mathrm{sID}, (c_i, \tilde{c}))$ be any of the queries to $\mathrm{DER}_{\mathrm{init}}$ such that $\mathrm{holder}[\mathrm{sID}] = i'$. If there exists no message $m_i$ such that $c_i = \mathsf{Enc}(pk_i, m_i; \mathsf{G}_{pk_{i'}, sk_{i'}}(m_i))$, the key $K$ is a random value that can not possibly correlate to any random oracle query to $|\mathsf{H}\rangle$ in both game and hence is independent of all other input to $\mathsf{B}$ in both games. But if there exists some message $m_i$ such that $c_i = \mathsf{Enc}(pk_i, m; \mathsf{G}_{pk_{i'}, sk_{i'}}(m_i))$, the respective key $K$ is defined as $\mathsf{H}(m'_i, m_j, \tilde{m}, pk^*, i, j)$ in both games: We have that $\mathsf{G}_{pk_{i'}, sk_{i'}}(m) \in \mathcal{R} \setminus \mathcal{R}_{\mathrm{bad}}(pk^*, sk^*, m)$ for all messages $m$. Therefore, it holds in particular for for $m'_i := \mathsf{Dec}(sk_{i'}, c_i)$ that $m'_i = m_i \neq \perp$, and hence, also that $\mathsf{Enc}(pk_i, m; \mathsf{G}_{pk_{i'}, sk_{i'}}(m'_i)) = c_i$. The same reasoning applies to all queries to $\mathrm{DER}_{\mathrm{resp}}$. For both bits it holds that $\mathsf{B}$'s view is identical in both games and

$$\Pr[G^{\neg sk}_{6^1/_3, b} \Rightarrow 1] = \Pr[G^{\neg sk}_{6^2/_3, b} \Rightarrow 1] \ .$$

Game $G^{\neg sk}_{7,b}$. In game $G^{\neg sk}_{7,b}$, we switch back to using $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{pk_{i'}, sk_{i'}}$. With the same reasoning as for the gamehop from game $\Pr[G^{\neg sk}_{6,b} \Rightarrow 1]$ to $\Pr[G^{\neg sk}_{6^1/_3, b} \Rightarrow 1]$, for both bits $b$ it holds that

$$\begin{aligned} |\Pr[G^{\neg sk}_{6^2/_3, b} \Rightarrow 1] - \Pr[G^{\neg sk}_{7,b} \Rightarrow 1]| &= |\Pr[\mathsf{GDPB}^{\mathsf{C}'}_{\lambda,1} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{C}'}_{\lambda,0} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S)^2 \cdot \delta \ , \end{aligned}$$

where adversary $\mathsf{C^{b}}'$ also is given in Figure 26.

Collecting the probabilities of the intermediate games yields the upper bound of equation (6), i.e., for both bits $b$ it holds that

$$|\Pr[G^{\neg sk}_{6,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{7,b} \Rightarrow 1]| \leq 16 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ ,$$

hence

$$\begin{aligned} |\Pr[G^{\mathsf{B}}_{2,1} \Rightarrow 1 \wedge \neg sk] - \Pr[G^{\mathsf{B}}_{2,0} \Rightarrow 1 \wedge \neg sk]| &= N \cdot |\Pr[G^{\neg sk}_{6,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{6,0} \Rightarrow 1]| \\ &\leq N \cdot |\Pr[G^{\neg sk}_{7,1} \Rightarrow 1] - \Pr[G^{\neg sk}_{7,0} \Rightarrow 1]| + 32 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ . \end{aligned}$$

We stress that from game $G^{\neg sk}_{7,b}$ on, none of the oracles uses $sk_{i'}$ any longer. To upper bound $|\Pr[G^{\neg sk}_{7,b} \Rightarrow 1] - 1/2|$, consider the sequence of games given in Figure 27, where we replace $\mathrm{sID}^*_{\mathrm{resp}}$'s ciphertext $c_i$ with a fake encryption. Like in case $(\neg\mathrm{st})$, we first have to add a guess for $\mathrm{sID}^*_{\mathrm{resp}}$.

Games $G^{\neg sk}_{8,b}$. In games $G^{\neg sk}_{8,b}$, one of the sessions that get established during execution of $\mathsf{B}$ is picked at random in line 06, and the games return 0 in line 19 if any other session $s'_{\mathrm{resp}}$ was picked than session $\mathrm{sID}^*_{\mathrm{resp}}$. Since for both bits $b$ it holds that both games $G^{\neg sk}_{8,b}$ and $G^{\neg sk}_{7,b}$ proceed identically unless $s'_{\mathrm{resp}} \neq \mathrm{sID}^*_{\mathrm{resp}}$, and since games $G^{\neg sk}_{8,b}$ output 0 if $s'_{\mathrm{resp}} \neq \mathrm{sID}^*_{\mathrm{resp}}$,

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

Games $G^{\neg sk}_{9,b}$. In games $G^{\neg sk}_{9,b}$, oracle $\mathrm{DER}_{\mathrm{resp}}$ is changed in line 35 such that for $\mathrm{sID}^*_{\mathrm{resp}}$, $c_i$ is no longer a ciphertext of the form $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$ for some randomly drawn message $m_i$, but a fake encryption $c_i \leftarrow \overline{\mathsf{Enc}}(pk_{i'})$. Consider the adversaries $\mathsf{A}^{\neg sk}_{\mathsf{DS,b}}$ given in Figure 28. The running times are the same as in case $(\neg\mathrm{st})$, see Equation (5), page 36:

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

and since $\mathsf{A}^{\neg sk}_{\mathsf{DS,b}}$ perfectly simulates game $G^{\neg sk}_{9,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(pk)$, and game $G^{\neg sk}_{8,b}$ if its input $c$ was generated by $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$ for some randomly picked message $m$,

$$|\Pr[G^{\neg sk}_{8,b} \Rightarrow 1] - \Pr[G^{\neg sk}_{9,b} \Rightarrow 1]| = \mathrm{Adv}^{\mathsf{DS}}_{\mathsf{T[PKE,G]}}(\mathsf{A}^{\neg sk}_{\mathsf{DS,b}}) \ ,$$

**GAMES** $G_{7,b}^{\neg sk}$-$G_{10,b}^{\neg sk}$

01 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$
02 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$
03 $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$
04 $\text{cnt}, \text{sID}^* := 0$
05 $i' \leftarrow_\$ [N]$
06 $s'_{\text{resp}} \leftarrow_\$ [S]$ // $G_{8,b}^{\neg sk}$-$G_{10,b}^{\neg sk}$
07 **for** $n \in [N]$
08 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
09 $b' \leftarrow \mathsf{B}^{\mathsf{O}, |\mathsf{G}\rangle, |\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
10 **if** ATTACK($\text{sID}^*$)
11 $\quad$ **return** 0
12 **if** $|\mathfrak{M}(\text{sID}^*)| \neq 1$ ABORT
13 Pick $\text{sID}^*_{\text{init}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. role[$\text{sID}^*_{\text{init}}$] = "initiator"
14 **if** corrupted[holder[$\text{sID}^*_{\text{init}}$]] ABORT
15 Pick $\text{sID}^*_{\text{resp}} \in \{\text{sID}^*, \text{sID}'\}$ s. th. role[$\text{sID}^*_{\text{resp}}$] = "responder" // $G_{8,b}^{\neg sk}$-$G_{10,b}^{\neg sk}$
16 **if** holder[$\text{sID}^*_{\text{init}}$] $\neq i'$
17 $\quad$ **return** 0
18 **if** $\text{sID}^*_{\text{resp}} \neq s'_{\text{resp}}$
19 $\quad$ **return** 0 // $G_{8,b}^{\neg sk}$-$G_{10,b}^{\neg sk}$
20 **return** b'

TEST(sID) //only one query

21 $\text{sID}^* := \text{sID}$
22 **if** sKey[$\text{sID}^*$] $= \bot$
23 $\quad$ **return** $\bot$
24 $K_0^* := \text{sKey}[\text{sID}^*]$ // $G_{7,b}^{\neg sk}$-$G_{9,b}^{\neg sk}$
25 $K_0^* \leftarrow_\$ \mathcal{K}$ // $G_{10,0}^{\neg sk}$
26 $K_1^* \leftarrow_\$ \mathcal{K}$
27 **return** $K_b^*$

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

28 **if** holder[sID] $= \bot$ **or** sKey[sID] $\neq \bot$ **or** role[sID] = "initiator"
29 $\quad$ **return** $\bot$
30 role[sID] := "responder"
31 $(j, i) :=$ (holder[sID], peer[sID])
32 $m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
33 $c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
34 **if** $\text{sID} = s'_{\text{resp}}$
35 $\quad c_i \leftarrow \overline{\mathsf{Enc}}(pk_{i'})$ // $G_{9,b}^{\neg sk}$-$G_{10,b}^{\neg sk}$
36 $\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
37 $M' := (c_i, \tilde{c})$
38 **if** $j = i'$
39 $\quad K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
40 **else**
41 $\quad m'_j := \mathsf{Dec}(sk_j, c_j)$
42 $\quad$ **if** $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
43 $\quad\quad K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
44 $\quad$ **else**
45 $\quad\quad$ **if** $i = i'$
46 $\quad\quad\quad K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
47 $\quad\quad$ **else**
48 $\quad\quad\quad K' := \mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}, i, j)$
49 sKey[sID] := $K'$
50 (received[sID], sent[sID]) := $(M, M')$
51 **return** $M'$

Figure 27: Games $G_{7,b}^{\neg sk}$ - $G_{10,b}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5.2.

and folding $\mathsf{A}_{\mathsf{DS},0}^{\neg\text{st}}$ and $\mathsf{A}_{\mathsf{DS},1}^{\neg\text{st}}$ into one adversary $\mathsf{A}_{\mathsf{DS}}^{\neg\text{st}}$ yields

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

So far, we established

$$|\Pr[G_{7,1}^{\neg sk} \Rightarrow 1] - \Pr[G_{7,0}^{\neg sk} \Rightarrow 1]| \leq S \cdot |\Pr[G_{9,1}^{\neg sk} \Rightarrow 1] - \Pr[G_{9,0}^{\neg sk} \Rightarrow 1]| + 2S \cdot \text{Adv}_{\mathsf{T[PKE,G]}}^{\mathsf{DS}}(\mathsf{A}_{\mathsf{DS}}^{\neg sk}) \ .$$

Game $G_{10,0}^{\neg sk}$. In game $G_{10,0}^{\neg sk}$, we change oracle TEST in line 25 such that it returns a random value instead of returning sKey[$\text{sID}^*$]. Since games $G_{9,1}^{\neg sk}$ and $G_{10,0}^{\neg sk}$ are equal,

$$|\Pr[G_{9,1}^{\neg sk} \Rightarrow 1] - \Pr[G_{9,0}^{\neg sk} \Rightarrow 1]| = |\Pr[G_{10,0}^{\neg sk} \Rightarrow 1] - \Pr[G_{9,0}^{\neg sk} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G_{10,0}^{\neg sk} \Rightarrow 1] - \Pr[G_{9,0}^{\neg sk} \Rightarrow 1]|$, which means upper bounding the probability that $\mathsf{B}$ obtains sKey[$\text{sID}^*$] in game $G_{9,0}^{\neg sk}$ by a classical query to any of the oracles included in $\mathsf{O}$ (except for TEST), and the probability that any quantum answer of the random oracle contains sKey[$\text{sID}^*$]. With the same reasoning as in case $(\neg\text{st})$,

$$|\Pr[G_{10,0}^{\neg sk} \Rightarrow 1] - \Pr[G_{9,0}^{\neg sk} \Rightarrow 1]| \leq \frac{S-2}{|\mathcal{M}|} \cdot \max\{\gamma(\mathsf{KG}), \frac{\delta}{|\mathcal{M}|}\} + \epsilon_{\text{dis}} \leq \frac{S}{|\mathcal{M}|} + \epsilon_{\text{dis}} \ .$$

```
A_{DS,b}^{¬sk |H'⟩,|H_q⟩,|G⟩}(pk, c)
cnt, sID* := 0
i' ←$ [N]
s'_resp ←$ [S]
for n ∈ [N] \ {i'}
    (pk_n, sk_n) ← KG
pk_{i'} := pk
b' ← B^{O,|G⟩,|H⟩}((pk_n)_{n∈[N]})
if ATTACK(sID*)
    return 0
if |𝔐(sID*)| ≠ 1 ABORT
Pick sID*_init ∈ {sID*, sID'} s. th.
    role[sID*_init] = "initiator"
if corrupted[holder[sID*_init]] ABORT
Pick sID*_resp ∈ {sID*, sID'} s. th.
    role[sID*_resp] = "responder"
if holder[sID*_init] ≠ i'
    return 0
if sID*_resp ≠ s'_resp
    return 0
return b'

CORRUPT(i ∈ [N] \ {i'})
if corrupted[i] return ⊥
corrupted[i] := true
return sk_i

DER_resp(sID, M = (p̃k, c_j))
if holder[sID] = ⊥ or sKey[sID] ≠ ⊥
  or role[sID] = "initiator"
    return ⊥
role[sID] := "responder"
(j, i) := (holder[sID], peer[sID])
m_i, m̃ ←$ 𝓜
c_i := Enc(pk_i, m_i; G(m_i))
if sID = s'_resp
    c_i := c
c̃ := Enc(p̃k, m̃; G(m̃))
M' := (c_i, c̃)
if j = i'
    K' := H_q(c_i, c_j, m̃, p̃k, i, j)
else
    m'_j := Dec(sk_j, c_j)
    if m'_j = ⊥
      or c_j ≠ Enc(pk_j, m'_j; G(m'_j))
        K' := H'_R(m_i, c_j, m̃, p̃k, i, j)
    else
        if i = i'
            K' := H_q(c_i, c_j, m̃, p̃k, i, j)
        else K' := H(m_i, m'_j, m̃, p̃k, i, j)
sKey[sID] := K'
(received[sID], sent[sID]) := (M, M')
return M'
```

Figure 28: Adversaries $\mathsf{A}_{\mathsf{DS,b}}^{\neg sk}$ for case $(\neg sk)$ of the proof of Lemma 5.2, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H_q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for $\mathrm{DER}_{\mathrm{resp}}$ and CORRUPT are defined as in game $G_{8,b}^{\neg sk}$ (see Figure 27). Again, internal random oracles ($\mathsf{H}'_\mathsf{R}$, and $\mathsf{H}'_{\mathsf{L1}}$ to $\mathsf{H}'_{\mathsf{L3}}$) can be simulated via lazy sampling since they are only accessible indirectly via $\mathrm{DER}_{\mathrm{resp}}$ and $\mathrm{DER}_{\mathrm{init}}$ which are queried classically.

Collecting the probabilities, we obtain

$$|\Pr[G_{2,1}^{\mathsf{B}} \Rightarrow 1 \wedge \neg sk] - \Pr[G_{2,0}^{\mathsf{B}} \Rightarrow 1 \wedge \neg sk]| \leq 2SN \cdot \mathrm{Adv}_{\mathsf{T[PKE,G]}}^{\mathsf{DS}}(\mathsf{A}_{\mathsf{DS}}^{\neg sk}) + 32N \cdot (q_\mathsf{G} + 2q_\mathsf{H} + 3S)^2 \cdot \delta + SN \cdot \epsilon_{\mathrm{dis}} + \frac{S^2 \cdot N}{|\mathcal{M}|} \enspace ,$$

the upper bound we claimed in equation (4).

# C Proof of Lemma 5.3

Tampering with the protocol ($\mathfrak{M}(\mathrm{sID}^*) = \varnothing$). Recall that we are proving an upper bound for $|\Pr[\mathsf{IND\text{-}StAA}_1^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing] - \Pr[\mathsf{IND\text{-}StAA}_0^{\mathsf{B}} \Rightarrow 1 \wedge \mathfrak{M}(\mathrm{sID}^*) = \varnothing]|$. Therefore, we will first enforce that indeed, we only need to consider the case where $\mathfrak{M}(\mathrm{sID}^*) = \varnothing$. Consider the sequence of games given in Figure 29.

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

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

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

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

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

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

INIT(sID)

**if** holder[sID] = ⊥
**or** sent[sID] ≠ ⊥ **return** ⊥
role[sID] := "initiator"
$i$ := holder[sID]
$j$ := peer[sID]
$m_j \leftarrow_\$ \mathcal{M}$
$c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
$(\tilde{pk}, \tilde{sk}) \leftarrow \mathsf{KG}$
$M := (\tilde{pk}, c_j)$
state[sID] := $(\tilde{sk}, m_j, M)$
sent[sID] := $M$
**return** $M$

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

**if** holder[sID] = ⊥ **or** sKey[sID] ≠ ⊥
**or** role[sID] = "initiator" **return** ⊥
role[sID] := "responder"
$(j, i)$ := (holder[sID], peer[sID]
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m'_j := \mathsf{Dec}(sk_j, c_j)$
**if** $m'_j = \bot$ **or** $c_j \neq \mathsf{Enc}(pk_j, m'_j; \mathsf{G}(m'_j))$
    $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
**else** $K' := \mathsf{H}(m_i, m'_j, \tilde{m}, \tilde{pk}, i, j)$
sKey[sID] := $K'$
(received[sID], sent[sID]) := $(M, M')$
**return** $M'$

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

**if** holder[sID] = ⊥ **or** state[sID] = ⊥
**or** sKey[sID] ≠ ⊥ **return** ⊥
$(i, j)$ := (holder[sID], peer[sID])
$(\tilde{sk}, m_j, \tilde{pk}, c_j)$ := state[sID]
$m'_i := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
**if** $m'_i = \bot$ **or** $c_i \neq \mathsf{Enc}(pk_i, m'_i; \mathsf{G}(m'_i))$
    **if** $\tilde{m}' = \bot$
        $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
    **else**
        $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
**else if** $\tilde{m}' = \bot$
        $K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
**else** $K := \mathsf{H}(m'_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
sKey[sID] := $K$
received[sID] := $M'$

Figure 29: Games $G_{0,b}$ - $G_{1,b}$ for case two of the proof of Theorem 5.1. Helper procedure ATTACK and oracles TEST, EST, CORRUPT, REVEAL and REV-STATE remains as in the original IND-StAA game (see Figures 16 and 17).

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

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

As discussed below Definition 4.1, 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 could pick message $(c_i, \tilde{c})$ on its own (thereby being able to control both $m_i^*$ and $\tilde{m}^*$), peer[sID*] (henceforth called $j^*$) remains uncorrupted throughout the game, and also the internal state state[sID*] remains unrevealed. Therefore, message $m_j^*$ can not be obtained trivially and ciphertext $c_j^*$ can be replaced.

Consider the sequence of games given in Figures 30 and 33: First, we will enforce that indeed, we only need to consider the case where role[sID*] = "initiator". Afterwards, we ensure that the game makes no use of $sk_{j^*}$ any longer by patching encryption into the random oracle (in games $G_{2,b}^{\text{init}}$ to $G_{6,b}^{\text{init}}$, see Figure 30). Again, this is the only part of the proof where the correctness error comes into play. Next, during execution of INIT(sID*), we replace ciphertext $c_j$ with a fake ciphertext that gets sampled using $\overline{\mathsf{Enc}}$ (games $G_{7,b}^{\text{init}}$ to $G_{8,b}^{\text{init}}$, see Figure 33, line 28). 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 $G_{9,b}^{\text{init}}$, also Figure 33).

**GAMES** $G_{1,b}^{\text{init}}$ - $G_{6,b}^{\text{init}}$

$\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$ // $G_{5,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
$\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$ // $G_{5,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
$\text{cnt}, \text{sID}^* := 0$
$j' \leftarrow_\$ [N]$ // $G_{3,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
**for** $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$b' \leftarrow \mathsf{B}^{\mathsf{O}, |\mathsf{G}\rangle, |\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
**if** ATTACK(sID*)
  **return** 0
**if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT
**if** role[sID*] = "responder"
  ABORT // $G_{2,b}^{\text{init}}$-$G_{5,b}^{\text{init}}$
**if** peer[sID*] $\neq j'$
  **return** 0 // $G_{3,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
**return** $b'$

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

**if** holder[sID] = $\perp$ **or** state[sID] = $\perp$
  **or** sKey[sID] $\neq \perp$ **return** $\perp$
$(i, j) :=$ (holder[sID], peer[sID])
$(\tilde{sk}, m_j, \tilde{pk}, c_j) :=$ state[sID]
$m_i' := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
**if** $m_i' = \perp$ **or** $c_i \neq \mathsf{Enc}(pk_i, m_i'; \mathsf{G}(m_i'))$
  **if** $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
  **else**
    $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $i = j'$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{6,b}^{\text{init}}$
**else if** $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L3}}(m_i', m_j, \tilde{c}, \tilde{pk}, i, j)$
    **if** $i = j'$
      $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$ // $G_{4,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
**else** $K := \mathsf{H}(m_i', m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $j' \in \{i, j\}$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{5,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
sKey[sID] := $K$
received[sID] := $M'$

INIT(sID)

**if** holder[sID] = $\perp$ **or** sent[sID] $\neq \perp$
  **return** $\perp$
role[sID] := "initiator"
$i :=$ holder[sID]
$j :=$ peer[sID]
$m_j \leftarrow_\$ \mathcal{M}\}$
$c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
$(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{KG}$
$M := (\tilde{pk}, c_j)$
state[sID] := $(\tilde{sk}, m_j, M)$
sent[sID] := $M$
**return** $M$

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

**if** holder[sID] = $\perp$ **or** sKey[sID] $\neq \perp$
  **or** role[sID] = "initiator"
  **return** $\perp$
role[sID] := "responder"
$(j, i) :=$ (holder[sID], peer[sID]
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m_j' := \mathsf{Dec}(sk_j, c_j)$
**if** $m_j' = \perp$ **or** $c_j \neq \mathsf{Enc}(pk_j, m_j'; \mathsf{G}(m_j'))$
  $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
  **if** $j = j'$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{6,b}^{\text{init}}$
**else** $K' := \mathsf{H}(m_i, m_j', \tilde{m}, \tilde{pk}, i, j)$
  **if** $j' \in \{i, j\}$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G_{5,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$
sKey[sID] := $K'$
(received[sID], sent[sID]) := $(M, M')$
**return** $M'$

$\mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j)$ // $G_{5,b}^{\text{init}}$-$G_{6,b}^{\text{init}}$

**if** $j' \in \{i, j\}$
  **return** $\mathsf{H_q}(\mathsf{Enc}(pk_i, m_1; \mathsf{G}(m_1)), \mathsf{Enc}(pk_j, m_2; \mathsf{G}(m_2)), m_3, \tilde{pk}, i, j)$
**return** $\mathsf{H}'(m_1, m_2, m_3, \tilde{pk}, i, j)$

Figure 30: Games $G_{1,b}^{\text{init}}$ - $G_{6,b}^{\text{init}}$ for case (init) of the proof of Lemma 5.3. Helper procedure ATTACK and oracles TEST, EST, REVEAL and REV-STATE remain as in the original IND-StAA game (see Figure 16 and Figure 17, pages 20 and 21).

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

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

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

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

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

Games $G_{3,b}^{\text{init}}$. In games $G_{3,b}^{\text{init}}$, one of the parties is picked at random in line 04, and the game returns 0 in line 14 if any other party $j'$ was picked than the test session's peer.

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

To prepare getting rid of $sk_{j'}$, we first change $\text{DER}_{\text{init}}$ such that whenever ciphertext $\tilde{c}$ induces decryption failure, $sk_{j'}$ is not used anymore.

Games $G_{4,b}^{\text{init}}$. In games $G_{4,b}^{\text{init}}$, we change oracle $\text{DER}_{\text{init}}$ in line 31 such that if the session's holder is $j'$ and $\tilde{c}$ does not decrypt to a message $\tilde{m}'$ s. th. $\tilde{c} = \mathsf{Enc}(\tilde{pk}, \tilde{m}', \mathsf{G}(\tilde{m}'))$, the session key is defined as $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$. (Before this change we let $K := \mathsf{H}'_{\mathsf{L3}}(m'_i, m_j, \tilde{c}, \tilde{pk}, i, j)$ in the case that $\tilde{c}$ fails to decrypt, but $c_i$ decrypts correctly). Since both $\mathsf{H}'_{\mathsf{L1}}$ and $\mathsf{H}'_{\mathsf{L3}}$ are not directly accessible and $\mathsf{Enc}(pk_{j'}, -)$ is injective, B's view does not change and

$$\Pr[G_{3,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1] = \Pr[G_{4,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1] \ .$$

The next two game-hops are done to achieve that $\text{DER}_{\text{init}}$ and $\text{DER}_{\text{resp}}$ do not use $sk_{j'}$ any more. In the next game, we only change key definition of $\text{DER}_{\text{init}}$ if both ciphertexts decrypt correctly, and key definition of $\text{DER}_{\text{resp}}$ if $c_j$ decrypts correctly. In these cases, we do note use the decryptions under $sk_{j'}$, but the ciphertexts themself. Similar to case ($\neg sk$), we "patch in" encryption into random oracle H whenever $j'$ appears as one of the involved parties. Due to the need for key consistency, we have to change patch encryption into the first *two* arguments.

Games $G_{5,b}^{\text{init}}$. In games $G_{5,b}^{\text{init}}$, the random oracle is changed as follows: Instead of picking H uniformly random, we pick two random oracles $\mathsf{H_q}$ and $\mathsf{H}'$ and define

$$
\begin{aligned}
&\mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j) \\
&\quad := \begin{cases} \mathsf{H_q}(\mathsf{Enc}(pk_i, m_1), \mathsf{Enc}(pk_j, m_2), m_3, \tilde{pk}, i, j) & j' \in \{i, j\} \\ \mathsf{H}(m_1, m_2, m_3, \tilde{pk}, i, j) & \text{o.w.} \end{cases} ,
\end{aligned}
$$

see line 69. Again, H remains truly random under the assumption that encryption is injective. The change of H is made explicit in oracles $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ in lines 64 and 34. Using the same analysis as in game $G_{6,b}^{\neg sk}$ of case ($\neg sk$), it is straightforward to see that

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

So far, we established

$$|\Pr[G_{2,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{2,0}^{\text{init}} \Rightarrow 1]| = N \cdot |\Pr[G_{5,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{5,0}^{\text{init}} \Rightarrow 1]| \ .$$

The final step to get rid of $sk_{j'}$ is to even out the key derivation for problematic ciphertexts: To this end, we also use $\mathsf{H_q}$ if a ciphertext fails to decrypt under $sk_{j'}$, instead of using the implicit reject.

GAMES $G^{\text{init}}_{6,b}$. In games $G^{\text{init}}_{6,b}$, we remove the implicit reject for ciphertexts with decryption failure under the secret key of $j'$ in lines 61 and 27. We claim

$$|\Pr[G^{\text{init}}_{5,b} \Rightarrow 1] - \Pr[G^{\text{init}}_{6,b} \Rightarrow 1]| \leq 16 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ . \tag{8}$$

The proof strategy is completely similar to case $(\neg sk)$: Intuitively, removing the implicit rejects can only affect B's view if keys were derived using error-inducing encryptions. We show that we can replace random oracle G with an oracle $\mathsf{G}_{pk_{i'},sk_{i'}}$ that makes error-inducing encryptions impossible, while distinguishing G from $\mathsf{G}_{pk_{i'},sk_{i'}}$ is reducable to winning GDPB. To verify this upper bound, consider the sequence of intermediate games given in Figure 31.

GAMES $G^{\text{init}}_{5^1/3,b}$. In games $G^{\text{init}}_{5^1/3,b}$, we enforce that no decryption failure with respect to key pair $(pk_{j'}, sk_{j'})$ will occur by replacing random oracle G with $\mathsf{G}_{pk_{j'},sk_{j'}}(m)$ in line 09, where $\mathsf{G}_{pk_{j'},sk_{j'}}(m)$ is defined by

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

To upper bound $|\Pr[G^{\text{init}\,\mathsf{B}}_{5,b} \Rightarrow 1] - \Pr[{G^{\text{init}}_{5^1/3,b}}^{\mathsf{B}} \Rightarrow 1]|$ for each bit $b$, we construct quantum adversaries $\mathsf{D^b}$ against $\mathsf{GDPB}_\lambda$ in Figure 32, issuing at most $q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S$ queries to $|\mathsf{F}\rangle$. With the same reasoning as for case $(\neg\text{st})$ (see page 42),

$$\begin{aligned} |\Pr[G^{\text{init}\,\mathsf{B}}_{5,b} \Rightarrow 1]| - \Pr[{G^{\text{init}}_{5^1/3,b}}^{\mathsf{B}} \Rightarrow 1]| =& |\Pr[\mathsf{GDPB}^{\mathsf{D^b}}_{\lambda,0} = 1] - \Pr[\mathsf{GDPB}^{\mathsf{D^b}}_{\lambda,1} = 1]| \\ &\leq 8 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S)^2 \cdot \delta \ . \end{aligned}$$

GAMES $G^{\text{init}}_{5^2/3,b}$. In games $G^{\text{init}}_{5^2/3,b}$, we change $\text{DER}_{\text{resp}}$ in line 32 such that whenever the session's holder is $j'$, the session key is defined as $K' := \mathsf{H_q}(c_i, c'_j, \tilde{m}, pk^*, i, j)$ instead of letting $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ if $c_j$ fails to decrypt. Likewise, we change $\text{DER}_{\text{init}}$ in line 51 such that if the session's holder is $j'$, whenever $\tilde{c}$ decrypts correctly, the session key is defined as $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ instead of letting $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$ if $c_i$ fails to decrypt. With the same reasoning as in case $(\neg sk)$, this change does not affect B's view and

$$\Pr[{G^{\text{init}}_{5^1/3,b}}^{\mathsf{B}} \Rightarrow 1] = \Pr[{G^{\text{init}}_{5^2/3,b}}^{\mathsf{B}} \Rightarrow 1] \ .$$

**GAMES** $G^{\text{init}}_{5,b}$-$G^{\text{init}}_{6,b}$

$\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3\times\mathcal{PK}\times[N]^2}$
$\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2\times\mathcal{M}\times\mathcal{PK}\times[N]^2}$
$\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ // $G^{\text{init}}_{5,b}$, $G^{\text{init}}_{6,b}$
Pick $2q$-wise hash $f$ // $G^{\text{init}}_{51/3,b}$-$G^{\text{init}}_{52/3,b}$
$\text{cnt}, \text{sID}^* := 0$
$j' \leftarrow_\$ [N]$
**for** $n \in [N]$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$\mathsf{G} := \mathsf{G}_{pk_{j'}, sk_{j'}}$ // $G^{\text{init}}_{51/3,b}$-$G^{\text{init}}_{52/3,b}$
$b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
**if** ATTACK$(\text{sID}^*)$
  **return** 0
**if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT
**if** role$[\text{sID}^*]$ = "responder"
  ABORT
**if** peer$[\text{sID}^*] \neq j'$
  **return** 0
**return** b'

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

**if** holder[sID] $= \perp$ **or** state[sID] $= \perp$
  **or** sKey[sID] $\neq \perp$ **return** $\perp$
$(i, j) :=$ (holder[sID], peer[sID])
$(\tilde{sk}, m_j, \tilde{pk}, c_j) :=$ state[sID]
$m_i' := \mathsf{Dec}(sk_i, c_i)$
$\tilde{m}' := \mathsf{Dec}(\tilde{sk}, \tilde{c})$
**if** $m_i' = \perp$ **or** $c_i \neq \mathsf{Enc}(pk_i, m_i'; \mathsf{G}(m_i'))$
  **if** $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
  **else**
    $K := \mathsf{H}'_{\mathsf{L2}}(c_i, m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $i = j'$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G^{\text{init}}_{52/3,b}$-$G^{\text{init}}_{6,b}$
**else if** $\tilde{m}' = \perp$
    $K := \mathsf{H}'_{\mathsf{L3}}(m_i', m_j, \tilde{c}, \tilde{pk}, i, j)$
    **if** $i = j'$
      $K := \mathsf{H}'_{\mathsf{L1}}(c_i, m_j, \tilde{c}, \tilde{pk}, i, j)$
**else** $K := \mathsf{H}(m_i', m_j, \tilde{m}', \tilde{pk}, i, j)$
    **if** $j' \in \{i, j\}$
      $K := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
sKey[sID] $:= K$
received[sID] $:= M'$

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

**if** holder[sID] $= \perp$ **or** sKey[sID] $\neq \perp$
  **or** role[sID] = "initiator"**return** $\perp$
role[sID] := "responder"
$(j, i) :=$ (holder[sID], peer[sID]
$m_i, \tilde{m} \leftarrow_\$ \mathcal{M}$
$c_i := \mathsf{Enc}(pk_i, m_i; \mathsf{G}(m_i))$
$\tilde{c} := \mathsf{Enc}(\tilde{pk}, \tilde{m}; \mathsf{G}(\tilde{m}))$
$M' := (c_i, \tilde{c})$
$m_j' := \mathsf{Dec}(sk_j, c_j)$
**if** $m_j' = \perp$ **or** $c_j \neq \mathsf{Enc}(pk_j, m_j'; \mathsf{G}(m_j'))$
  $K' := \mathsf{H}'_{\mathsf{R}}(m_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
  **if** $j = j'$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$ // $G^{\text{init}}_{52/3,b}$-$G^{\text{init}}_{6,b}$
**else** $K' := \mathsf{H}(m_i, m_j', \tilde{m}, \tilde{pk}, i, j)$
  **if** $j' \in \{i, j\}$
    $K' := \mathsf{H_q}(c_i, c_j, \tilde{m}, \tilde{pk}, i, j)$
sKey[sID] $:= K'$
(received[sID], sent[sID]) $:= (M, M')$
**return** $M'$

$\mathsf{G}_{pk_{j'}, sk_{j'}}(m)$

$r := \mathsf{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk_{j'}, sk_{j'}, m); f(m))$
**return** $r$

Figure 31: Intermediate games $G^{\text{init}}_{5,b}$ - $G^{\text{init}}_{6,b}$ for case (init) of the proof of Lemma 5.3. All oracles except for $\mathsf{G}$, $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ remain as in game $G^{\text{init}}_{5,b}$. $f$ is an internal $2q$-wise independent hash function (like in games $G^{\neg sk}_{6,b}$ - $G^{\neg sk}_{7,b}$ of case $(\neg sk)$, see Figure 25), where $q := q_{\mathsf{G}} + 2 \cdot q_{\mathsf{H}} + 3 \cdot S$. $\mathsf{Sample}(Y; f(m))$ (again) denotes the deterministic execution of $\mathsf{Sample}(Y)$ using explicitly given randomness $f(m)$.

$\underline{\mathsf{D}_1^{\mathsf{b}} = \mathsf{D}_1^{\mathsf{b}'}}$

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

$\underline{\mathsf{G}(m)}$

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

$\underline{\text{CORRUPT}(i \in [N] \setminus \{j'\})}$

**if** corrupted$[i]$ **return** $\bot$
corrupted$[i]$ := **true**
**return** $sk_i$

$\underline{\mathsf{D}_2^{\mathsf{b}|\mathsf{F}\rangle}, \mathsf{D}_2^{\mathsf{b}'|\mathsf{F}\rangle}}$

$\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$
$\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$
Pick $2q$-wise hash $f$
cnt, sID$^*$ := 0
$j' \leftarrow_\$ [N]$
**for** $n \in [N] \setminus \{j'\}$
  $(pk_n, sk_n) \leftarrow \mathsf{KG}$
$(pk_{j'}, sk_{j'}) := (pk, sk)$
$b' \leftarrow \mathsf{B}^{\mathrm{O}, |\mathsf{G}\rangle, |\mathsf{H}\rangle}((pk_n)_{n \in [N]})$
ATTACK(sID$^*$)
  **return** 0
**if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT
**if** role[sID$^*$] = "responder"
  ABORT
**if** peer[sID$^*$] $\neq j'$
  **return** 0
**return** b'

Figure 32: Adversaries $\mathsf{D}^{\mathsf{b}} = (\mathsf{D}_1^{\mathsf{b}}, \mathsf{D}_2^{\mathsf{b}})$ and $\mathsf{D}^{\mathsf{b}'} = (\mathsf{D}_1^{\mathsf{b}'}, \mathsf{D}_2^{\mathsf{b}'})$ executed in game $\mathsf{GDPB}_{\delta(pk,sk)}$ with access to $|\mathsf{F}\rangle$ for case (init) of the proof of Lemma 5.3. Similar to case (¬st), the adversaries only differ in their definition of $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$: For adversaries $\mathsf{D}^{\mathsf{b}}$, $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ are defined as in game $G_{5,b}^{\text{init}}$, see Figure 31, and for adversaries $\mathsf{D}^{\mathsf{b}'}$, $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ are defined as in game $G_{5^2/3,b}^{\neg\text{st}}$ (also Figure 31).

Game $G_{6,b}^{\text{init}}$. In game $G_{6,b}^{\text{init}}$, we switch back to using $\mathsf{G} \leftarrow_\$ \mathcal{R}^{\mathcal{M}}$ instead of $\mathsf{G}_{pk_{j'},sk_{j'}}$. With the same reasoning as for the gamehop from game $\Pr[G_{5,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1]$ to $\Pr[G_{5^1/_3,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1]$,

$$|\Pr[G_{5^2/_3,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{6,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1]| = |\Pr[\mathsf{GDPB}_{\lambda,0}^{\mathsf{D}'} = 1] - \Pr[\mathsf{GDPB}_{\lambda,1}^{\mathsf{D}'} = 1]|$$
$$\leq 8 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3 \cdot S)^2 \cdot \delta \ ,$$

where adversaries $\mathsf{D}^{\mathsf{b}'}$ also are given in Figure 32.

Collecting the probabilities of the intermediate games yields the upper bound of equation (8), i.e., for both bits it holds that

$$|\Pr[G_{5,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1] - \Pr[G_{6,b}^{\text{init}\,\mathsf{B}} \Rightarrow 1]| \leq 16 \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ ,$$

hence

$$|\Pr[G_{2,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{2,0}^{\text{init}} \Rightarrow 1]| = N \cdot |\Pr[G_{5,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{5,0}^{\text{init}} \Rightarrow 1]|$$
$$\leq N \cdot |\Pr[G_{6,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{6,0}^{\text{init}} \Rightarrow 1]| + 32N \cdot (q_{\mathsf{G}} + 2q_{\mathsf{H}} + 3S)^2 \cdot \delta \ .$$

We stress that from games $G_{6,b}^{\text{init}}$ on, none of the oracles uses $sk_{j'}$ any longer. To upper bound $|\Pr[G_{6,1}^{\text{init}} \Rightarrow 1] - \Pr[G_{6,0}^{\text{init}} \Rightarrow 1]|$, consider the sequence of games given in Figure 33, where we replace sID*'s ciphertext $c_j$ with a fake encryption.

**GAMES $G_{6,b}^{\text{init}}$-$G_{9,b}^{\text{init}}$**

01 $\mathsf{H}' \leftarrow_\$ \mathcal{K}^{\mathcal{M}^3 \times \mathcal{PK} \times [N]^2}$
02 $\mathsf{H_q} \leftarrow_\$ \mathcal{K}^{\mathcal{C}^2 \times \mathcal{M} \times \mathcal{PK} \times [N]^2}$
03 $\text{cnt}, \text{sID}^* := 0$
04 $j' \leftarrow_\$ [N]$
05 **for** $n \in [N]$
06 $\quad (pk_n, sk_n) \leftarrow \mathsf{KG}$
07 $s' \leftarrow_\$ [S]$ // $G_{7,b}^{\text{init}}$-$G_{9,b}^{\text{init}}$
08 $b' \leftarrow \mathsf{B}^{\mathrm{O},|\mathsf{G}\rangle,|\mathsf{H}\rangle}((pk_n)_{n\in[N]})$
09 **if** ATTACK(sID*)
10 $\quad$ **return** 0
11 **if** $\mathfrak{M}(\text{sID}^*) \neq \varnothing$ ABORT
12 **if** role[sID*] = "responder"
13 $\quad$ ABORT
14 **if** peer[sID*] $\neq j'$
15 $\quad$ **return** 0
16 **if** peer[sID*] $\neq j'$
17 $\quad$ **return** 0
18 **if** $\text{sID}^* \neq s'$
19 $\quad$ **return** 0 // $G_{7,b}^{\text{init}}$-$G_{9,b}^{\text{init}}$
20 **return** b'

INIT(sID)

21 **if** holder[sID] $= \perp$ **or** sent[sID] $\neq \perp$
22 $\quad$ **return** $\perp$
23 role[sID] := "initiator"
24 $i :=$ holder[sID], $j :=$ peer[sID]
25 $m_j \leftarrow_\$ \mathcal{M}$
26 $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$
27 **if** sID $= s'$
28 $\quad c_j \leftarrow \overline{\mathsf{Enc}}(pk_{j'})$ // $G_{8,b}^{\text{init}}$-$G_{9,b}^{\text{init}}$
29 $(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{KG}$
30 $M := (\tilde{pk}, c_j)$
31 state[sID] := $(\tilde{sk}, m_j, M)$
32 sent[sID] := $M$
33 **return** $M$

TEST(sID) //only one query

34 sID* := sID
35 **if** sKey[sID*] $= \perp$ **return** $\perp$
36 $K_0^* :=$ sKey[sID*] // $G_{6,b}^{\text{init}}$-$G_{8,b}^{\text{init}}$
37 $K_0^* \leftarrow_\$ \mathcal{K}$ // $G_{9,0}^{\text{init}}$
38 $K_1^* \leftarrow_\$ \mathcal{K}$
39 **return** $K_b^*$

Figure 33: Games $G_{6,b}^{\text{init}}$ - $G_{9,b}^{\text{init}}$ for case (init) of the proof of Lemma 5.3. All oracles except for INIT and TEST remain as in game $G_{6,b}^{\text{init}}$ (see Figure 30).

Games $G_{7,b}^{\text{init}}$. In games $G_{7,b}^{\text{init}}$, one of the sessions that gets established during execution of B is picked at random in line 07, and the game returns 0 in line 19 if any other session $s'$ was picked than test session sID*.

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

Games $G_{8,b}^{\text{init}}$. In games $G_{8,b}^{\text{init}}$, oracle INIT is changed in line 28 such that for $s'$, $c_j$ is no longer a ciphertext of the form $c_j := \mathsf{Enc}(pk_j, m_j; \mathsf{G}(m_j))$ for some randomly drawn message $m_j$, but a fake encryption $c_j \leftarrow \overline{\mathsf{Enc}}(pk_{j'})$. Consider the adversaries $\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}$ given in Figure 34. The running time is the same as in case (¬st), see Equation (5):

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

and since $\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}$ perfectly simulates game $G^{\text{init}}_{8,b}$ if its input was generated by $c \leftarrow \overline{\mathsf{Enc}}(pk)$, and game $G^{\text{init}}_{7,b}$ if its input $c$ was generated by $c := \mathsf{Enc}(pk, m; \mathsf{G}(m))$ for some randomly picked message $m$,

$$|\Pr[G^{\text{init}}_{7,b} \Rightarrow 1] - \Pr[G^{\text{init}}_{8,b} \Rightarrow 1]| = \text{Adv}^{\mathsf{DS}}_{\mathsf{T[PKE,G]}}(\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}) \ ,$$

and folding $\mathsf{A}^{\text{init}}_{\mathsf{DS,0}}$ and $\mathsf{A}^{\text{init}}_{\mathsf{DS,1}}$ into one adversary $\mathsf{A}^{\text{init}}_{\mathsf{DS}}$ yields

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

```
A_DS,b^init |H'⟩,|H_q⟩,|G⟩ (pk, c)
cnt, sID* := 0
j' ←$ [N]
s' ←$ [S]
for n ∈ [N] \ {j'}
   (pk_n, sk_n) ← KG
pk_j' := pk
b' ← B^{O,|RO⟩}(pk_1, ··· , pk_N)
if ATTACK(sID*)
   return 0
if 𝔐(sID*) ≠ ∅ ABORT
if role[sID*] = "responder"
   ABORT
if peer[sID*] ≠ j' return 0
if peer[sID*] ≠ j' return 0
if sID* ≠ s' return 0
return b'

CORRUPT(i ∈ [N] \ {j'})
if corrupted[i] return ⊥
corrupted[i] := true
return sk_i

INIT(sID)
if holder[sID] = ⊥
   return ⊥
if sent[sID] ≠ ⊥
   return ⊥
role[sID] := "initiator"
i := holder[sID]
j := peer[sID]
m_j ←$ M
c_j := Enc(pk_j, m_j; G(m_j))
if sID = s'
   c_j := c
(s̃k, p̃k) ← KG
M := (p̃k, c_j)
state[sID] := (s̃k, m_j, M)
sent[sID] := M
return M
```

Figure 34: Adversaries $\mathsf{A}^{\text{init}}_{\mathsf{DS,b}}$ for case (init) of the proof of Lemma 5.3, with oracle access to $|\mathsf{H}'\rangle$, $|\mathsf{H_q}\rangle$ and $|\mathsf{G}\rangle$. All oracles except for INIT and CORRUPT are defined as in game $G^{\text{init}}_{7,b}$ (see Figure 33). Again, internal random oracles ($\mathsf{H}'_{\mathsf{R}}$, and $\mathsf{H}'_{\mathsf{L1}}$ to $\mathsf{H}'_{\mathsf{L3}}$) can be simulated via lazy sampling since they are only accessible indirectly via $\text{DER}_{\text{resp}}$ and $\text{DER}_{\text{init}}$ which are queried classically.

So far, we established

$$|\Pr[G^{\text{init}}_{6,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{6,0} \Rightarrow 1]| \leq S \cdot |\Pr[G^{\text{init}}_{8,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{8,0} \Rightarrow 1]| + 2S \cdot \text{Adv}^{\mathsf{DS}}_{\mathsf{T[PKE,G]}}(\mathsf{A}^{\text{init}}_{\mathsf{DS}}) \ .$$

Game $G^{\text{init}}_{9,0}$. In game $G^{\text{init}}_{9,0}$, we change oracle TEST in line 37 such that it returns a random value instead of sKey[sID*]. Since games $G^{\text{init}}_{8,1}$ and $G^{\text{init}}_{9,0}$ are equal,

$$|\Pr[G^{\text{init}}_{8,1} \Rightarrow 1] - \Pr[G^{\text{init}}_{8,0} \Rightarrow 1]| = |\Pr[G^{\text{init}\,\mathsf{B}}_{9,0} \Rightarrow 1] - \Pr[G^{\text{init}\,\mathsf{B}}_{8,0} \Rightarrow 1]| \ .$$

It remains to upper bound $|\Pr[G^{\text{init}\,\mathsf{B}}_{9,0} \Rightarrow 1] - \Pr[G^{\text{init}\,\mathsf{B}}_{8,0} \Rightarrow 1]|$, which means upper bounding the probability that $\mathsf{B}$ obtains sKey[sID*] in game $G^{\text{init}}_{8,0}$ by a query to any of the oracles included in O (except for

TEST), and the probability that any answer of the random oracle contains sKey[sID$^*$]. With the same reasoning as in case (¬st),

$$|\Pr[G_{9,0}^{\mathrm{init}^{\mathsf{B}}} \Rightarrow 1] - \Pr[G_{8,0}^{\mathrm{init}^{\mathsf{B}}} \Rightarrow 1]| \leq \frac{S-2}{|\mathcal{M}|} \cdot \delta \cdot \gamma(\mathsf{KG}) + \epsilon_{\mathrm{dis}} \leq \frac{S}{|\mathcal{M}|} + \epsilon_{\mathrm{dis}} \ .$$

Collecting the probabilities, we obtain

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

Case (resp). Intuition is as follows: While $\mathsf{B}$ could pick message $(c_j, \tilde{pk})$ on its own (thereby being able to control both $m_j$ and $\tilde{m}$), peer[sID$^*$] remains uncorrupted throughout the game, therefore, at least message $m_i$ (that was randomly picked by $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*, (c_j, \tilde{pk}))$) cannot be computed trivially. The proof differs from case (init) only in the following way: instead of changing INIT(sID$^*$) such that it outputs a fake encryption $c_j$, we change $\mathrm{DER}_{\mathrm{resp}}(\mathrm{sID}^*, m)$ such that it outputs a fake encryption $c_i$. We obtain a similar upper bound: there exists an adversary $\mathsf{A}_{\mathsf{DS}}^{\mathrm{resp}}$ such that

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

Collecting the probabilities, folding $\mathsf{A}_{\mathsf{DS}}^{\mathrm{init}}$ and $\mathsf{A}_{\mathsf{DS}}^{\mathrm{resp}}$ into one adversary $\mathsf{A}'$, and assuming that $N << S << |\mathcal{M}|$, we obtain

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

the upper bound bound given in Lemma 5.3.