Anonymous, Robust Post-Quantum Public Key Encryption

Anonymous, Robust Post-Quantum Public Key Encryption

Paul Grubbs[1], Varun Maram[2], and Kenneth G. Paterson[2]

[1] University of Michigan, USA.
[2] Department of Computer Science,
ETH Zurich, Switzerland.
paulgrubbs12@gmail.com,
vmaram@inf.ethz.ch,
kenny.paterson@inf.ethz.ch

Abstract. A core goal of the NIST PQC competition is to produce public-key encryption (PKE) schemes which, even if attacked with a large-scale quantum computer, maintain the security guarantees needed by applications. The main security focus in the NIST PQC context has been IND-CCA security, but other applications demand that PKE schemes provide *anonymity* (Bellare *et al.*, ASIACRYPT 2001), and *robustness* (Abdalla *et al.*, TCC 2010). Examples of such applications include anonymous communication systems, cryptocurrencies, anonymous credentials, searchable encryption, and auction protocols. Almost nothing is known about how to build post-quantum PKE schemes offering these security properties. In particular, the status of the NIST PQC candidates with respect to anonymity and robustness is unknown.

This paper initiates a systematic study of anonymity and robustness for post-quantum PKE schemes. Firstly, we identify implicit rejection as a crucial design choice shared by most post-quantum KEMs, show that implicit rejection renders prior results on anonymity and robustness for KEM-DEM PKEs inapplicable, and transfer prior results to the implicit-rejection setting where possible. Secondly, since they are widely used to build post-quantum PKEs, we examine how the Fujisaki-Okamoto (FO) transforms (Fujisaki and Okamoto, Journal of Cryptology 2013) confer robustness and enhance weak anonymity of a base PKE.

We then leverage our theoretical results to study the anonymity and robustness of three NIST KEM finalists—Saber, Kyber, and Classic McEliece—and one alternate, FrodoKEM. Overall, our findings for robustness are definitive: we provide positive robustness results for Saber, Kyber, and FrodoKEM, and a negative result for Classic McEliece. Our negative result stems from a striking property of KEM-DEM PKE schemes built with the Classic McEliece KEM: for any message m, we can construct a single hybrid ciphertext c which decrypts to the chosen m under *any* Classic McEliece private key.

Our findings for anonymity are more mixed: we identify barriers to proving anonymity for Saber, Kyber, and Classic McEliece. We also found that in the case of Saber and Kyber, these barriers lead to issues with their IND-CCA security claims. We have worked with the Saber and Kyber teams to fix these issues, but they remain unresolved. On the positive side, we were able to prove anonymity for FrodoKEM and a variant of Saber introduced by D'Anvers *et al.* (AFRICACRYPT 2018). Our analyses of these two schemes also identified technical gaps in their IND-CCA security claims, but we were able to fix them.

1 Introduction

The increasingly real threat of quantum computers breaking all widely-deployed public-key cryptography has driven research in new paradigms for building core public-key primitives like signatures, public-key encryption (PKE), and key encapsulation mechanisms (KEMs) from problems that are computationally intractable even for quantum computers. An umbrella term for this is *Post-Quantum Cryptography* (PQC). The US National Institute of Standards and Technology (NIST) is in the process of selecting new standards which will be used for decades to come. The process has reached its third round with four finalist candidates and five alternate candidates in the KEM/PKE category. The main security target of evaluation for these schemes until now has been IND-CCA security. This was appropriate as a starting point because it suffices for many important use cases. But we argue that the time has now come for a broader study of the candidates' fitness for emerging applications where security properties other than IND-CCA are required.

Two important security properties that go beyond IND-CCA security are *anonymity* (or key privacy) and *robustness*. Anonymity was first formalised in the public key setting by [9]. Roughly, a PKE scheme is

anonymous if a ciphertext does not leak anything about which public key was used to create it; strong forms of anonymity equip the adversary with a decryption oracle. Anonymous PKE is a fundamental component of several deployed anonymity systems, most notably anonymous cryptocurrencies like Zcash [11]. It is also important in building anonymous broadcast encryption schemes [7, 30], anonymous credential systems [13] and auction protocols [36]. Robustness for PKE, first formalised in [2], goes hand-in-hand with anonymity. Suppose a party equipped with a private key receives a ciphertext for an anonymous PKE scheme. In the absence of other information, how does a party decide that it is the intended receiver of that ciphertext? The standard approach is to perform trial decryption. Robustness provides an assurance that this process does not go wrong – that the receiver is not fooled into accepting a plaintext intended for someone else. Robustness is also important for maintaining consistency in searchable encryption [1] and ensuring auction bid correctness [36]. Various robustness notions for PKE were studied in [2], while stronger notions were introduced in [17]; the symmetric setting was treated in [18, 22, 16, 29].

To date, there is almost no work that shows how to build anonymous, robust post-quantum PKE schemes. Nor is it known whether the NIST candidates meet these extended notions. The only directly relevant work is by Mohassel [33], who showed a number of foundational results on anonymity and robustness of hybrid PKEs built via the KEM-DEM paradigm ("DEM" being an abbreviation for "data encapsulation mechanism"). Our work is influenced by Mohassel's general approach; however, Mohassel only considers KEMs that are directly constructed from strongly-secure PKEs via sampling a random message from the PKE scheme's message space and then PKE-encrypting it. This makes the results of [33] inapplicable to NIST candidates, for a few reasons. First, the NIST candidates are all KEMs, not PKEs, so there is a basic syntactic mismatch. Second, the base PKEs used within the candidate KEMs are only weakly (e.g. OW-CPA) secure, but [33] relies on the starting PKE having (e.g.) IND-CCA security. Finally, [33] only analyzes explicit-rejection KEMs, for which decapsulation can fail, but all the NIST candidates except the alternate candidate HQC [32] are actually implicit-rejection KEMs that never output $\perp$. This means, e.g., the NIST *finalist* KEMs cannot be even weakly robust, while the constructions of [33] all start from robust KEMs.

One of the negative results of [33] is that even if a KEM enjoys a strong anonymity property, the hybrid PKE scheme that results from applying the standard KEM-DEM construction may not be anonymous. This is concerning, since it indicates that if one only focuses on KEMs in the NIST competition, rather than the PKE schemes that will inevitably be built from them using the standard KEM-DEM approach, then there is no guarantee that desired security properties will actually carry over. Thus, one must dig into a KEM's internals if the target is to achieve anonymous hybrid PKE.

In fact, all the NIST candidates in the KEM/PKE category are constructed using variants of the Fujisaki-Okamato (FO) transform [19–21]. The FO transform takes a weakly secure PKE scheme (e.g. one that is OW-CPA or IND-CPA secure) and elevates it to a KEM that is IND-CCA secure. The FO transform and variants of it have recently been heavily analysed, [24, 35, 27, 38, 25], in the Random Oracle Model (ROM) and the Quantum ROM (QROM) [12], but insofar as we are aware, only with a view to establishing IND-CCA security of the resulting KEMs. Only one prior work [23] studies the relationship between FO transforms and anonymity; it shows that the original FO transform enhances anonymity in the ROM. But this result does not tell us whether the modern FO variants used by the NIST finalists also enhance (or even preserve) robustness and anonymity properties; notably, the results of [23] are not in the QROM.

Anonymity and robustness for the KEM-DEM paradigm. Our first main contribution is a modular theory of anonymity and robustness for PKE schemes built via the KEM-DEM paradigm. This extends the work of [33] to general KEMs (instead of those built only from PKEs). An interesting aspect that emerges is a fundamental separation between our results for implicit- and explicit-rejection KEMs. At a high level, KEMs that perform implicit rejection do not in general transfer anonymity and robustness to PKEs obtained via the KEM-DEM paradigm from the KEM component, whilst KEMs that offer explicit rejection, and that also satisfy a mild robustness property, do. Our positive result for explicit rejection KEMs relies on a relatively weak anonymity notion for KEMs which we introduce here, wANO-CCA security. Our negative results for the implicit rejection case are proved through the construction of specific counterexamples and are surprisingly strong. For example, an implicit rejection KEM cannot be robust, but can achieve a strong form of collision freeness (SCFR-CCA, that we define here). This is in some sense the next best thing to robustness. We show that even this property is not sufficient, by exhibiting an implicit rejection KEM that is ANO-CCA, IND-CCA and SCFR-CCA secure, and a DEM that is AE (authenticated encryption) secure and satisfies a

strong robustness property (XROB, from [18]), but where the PKE scheme resulting from composing this KEM and DEM is not ANO-CCA secure.

Anonymity and robustness from FO transforms. Since all the NIST finalists are KEMs of the implicit rejection type and we have a strong negative result there, we must dig deeper if we wish to assure ourselves that anonymity and robustness will be obtained for PKEs built from those KEMs. This introduces our second main contribution, wherein we analyse how the FO transform (and its variants) lift anonymity and robustness properties from a starting weakly-secure PKE scheme, first to the strongly-secure KEM built by the FO transform, and then to the hybrid PKE scheme constructed using the KEM-DEM paradigm.

For explicit-rejection KEMs, we show that for a slight variant of the $\mathrm{HFO}^{\perp}$ transform of [24], the base PKE's weak anonymity and robustness are enhanced to strong (ANO-CCA) anonymity and strong (SROB-CCA) robustness, as long as an intermediate deterministic PKE used in the transform is collision-free. For implicit-rejection KEMs, we show that the $\mathrm{FO}^{\not\perp}$ transform of [24] similarly enhances anonymity and collision-freeness. The culmination of this analysis is showing that KEMs and PKEs built via FO-type transforms can bypass our negative result for implicit rejection KEMs.

Application to NIST candidates. We then apply our above generic analysis for implicit-rejection KEMs to specific schemes related to the NIST PQC competition which employ a transform close to $\mathrm{FO}^{\not\perp}$. In particular, we focus on the NIST finalist Classic McEliece [3], a simplified version of the NIST finalist Saber [8] from [15] that we call "proto-Saber", and the NIST alternate candidate FrodoKEM [4]. The reason we consider proto-Saber instead of the actual Saber scheme is that the IND-CCA security claims made for Saber in its NIST third round specification [8] seem to have been taken from those of proto-Saber in [15] *without modification*. However, the actual technical specification of Saber in [8, Section 8] and the reference implementation of Saber differ from proto-Saber in crucial ways that impact on its formal security analysis. We return to this issue in more detail below and in Section 5.

For Classic McEliece, we show that the hybrid PKE resulting from applying the standard KEM-DEM construction is not strongly robust (in the sense defined in [2]). In fact, we can show that, for any plaintext m, it is possible to construct a single ciphertext c such that c always decrypts to m under *any* Classic McEliece private key. The construction of c does not even need the public key! We stress that this property does not indicate any problem with IND-CCA security of Classic McEliece, but it does expose its limitations as a general-purpose KEM for the broad set of applications that can be envisaged for NIST public key algorithms. Since our $\mathrm{FO}^{\not\perp}$-related results on anonymity of KEMs and PKEs built from them depend on robustness properties, Classic McEliece's limitations in this regard present a barrier to establishing its anonymity using our techniques (but do not preclude a direct proof).

For proto-Saber, the news is better. We provide positive results on anonymity and robustness properties of its KEM and the hybrid PKE schemes derived from it. Towards these results, we have to adapt our analysis on $\mathrm{FO}^{\not\perp}$ to the actual transform used by proto-Saber. In doing so, we were also able to obtain an explicit proof of IND-CCA security for proto-Saber in the QROM that matches the tightness claimed in [15]. This is relevant because despite claims to the contrary in [15], we find that even the IND-CCA security of proto-Saber cannot be directly proved using any of the known results concerning the $\mathrm{FO}^{\not\perp}$ transform. This is due to low-level details of how proto-Saber applies hash functions to intermediate values in its internal computations. These details are crucial given the delicate nature of QROM proofs and invalidate the direct application of known results on "standard" FO transforms in the QROM.

FrodoKEM uses an FO-type transform that is *identical* to that of proto-Saber. Hence, our positive results on tight IND-CCA security, anonymity and robustness of proto-Saber also apply to FrodoKEM in a similar fashion.

Saber and Kyber [6] both implement the same transform, one which hashes even more intermediate values than proto-Saber does. This creates barriers in applying the proof strategies that we used for proto-Saber when trying to establish anonymity of Saber and Kyber. Interestingly, as we explain in detail, these extra hashes also act as barriers in proving even the IND-CCA security of these two finalists in the QROM with the bounds as claimed in their respective specifications. We consider this an important finding given the centrality of IND-CCA security as the design target in the NIST competition. On a positive note, we show that our robustness analysis of proto-Saber can be extended to Saber and Kyber, which implies that these two NIST finalists lead to strongly robust hybrid PKE schemes. Finally, we suggest small modifications to

Saber and Kyber that would bring their FO-type transforms closer to that of proto-Saber and allow us to overcome the aforementioned problems.

Subsequent Work. The NIST finalist NTRU [14] uses altogether a different transform, namely $\mathrm{FO}_m^{\not\perp}$ [24], that differs from $\mathrm{FO}^{\not\perp}$ in a way which makes it difficult to extend our analysis of $\mathrm{FO}^{\not\perp}$ to NTRU. However, in subsequent work to ours, Xagawa [41] has established the anonymity and robustness properties of NTRU by utilizing a stronger property of its base PKE scheme, namely the so-called *strong disjoint-simulatability*.

Paper organisation. Section 2 contains preliminary definitions. Section 3 contains our anonymity and robustness definitions for KEMs, and analysis of generic KEM-DEM composition. Section 4 contains our study of anonymity and robustness enhancement for FO-type transforms, and the security of hybrid PKE built from FO-type KEMs. Section 5 contains our study of the NIST candidate KEMs.

2 Preliminaries

In this section, we briefly define the preliminaries necessary for the main body. We begin with defining the syntax of primitives of interest.

Primitives. A key encapsulation mechanism (KEM) $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ is a tuple of algorithms. The randomized key generation algorithm KGen takes no input and outputs a pair $(\mathsf{pk}, \mathsf{sk})$ of a public encapsulation key pk and a private decapsulation key sk. The randomized encapsulation algorithm Encap takes as input the encapsulation key pk, and outputs a pair (C, k) where C is a ciphertext and k is a bit string. The determinstic decapsulation algorithm Decap takes as input the encapsulation key pk, the decapsulation key sk, and the ciphertext C. If decapsulation can output either a key k or an error symbol $\perp$, we call the KEM an *explicit-rejection* KEM. If decapsulation can only output a key k, we call the KEM an *implicit-rejection* KEM.

A public-key encryption (PKE) scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is a tuple of algorithms. The algorithm KGen is the same as above for KEMs. (It is conventional to call KGen's outputs the encryption/public and decryption/private key, respectively, instead of "encapsulation"/"decapsulation" keys.) The randomized encryption algorithm Enc takes as input the public key pk, and message m, and outputs a ciphertext C. Below, we will sometimes use a modified syntax for encryption, where instead of sampling internal randomness, the algorithm is deterministic and takes random coins as an additional input. Letting r be a string of random bits, we will write $\mathsf{Enc}(\mathsf{pk}, m; r)$ to denote the output of Enc when run with randomness r. Finally, the deterministic decryption algorithm Dec takes as input the public key pk, the secret key sk, and a ciphertext C, and outputs a message m or an error symbol $\perp$.

In Appendix A, we define the syntax for authenticated encryption with associated data (AEAD or AE) schemes and message authentication codes (MACs), along with the *correctness* and γ-*spreadness* properties of PKE schemes and KEMs.

Associated to each algorithm that comprises a primitive above is one or more input spaces (e.g. sets of possible keys $\mathcal{K}$ and messages $\mathcal{M}$) and an output space (e.g. the set of possible ciphertexts $\mathcal{C}$). We assume each algorithm checks that each of inputs is in this set, and aborts if not. To reduce notational clutter, we will not make these input/output spaces explicit below, except where necessary.

The KEM-DEM framework. Composing a KEM and a data encapsulation mechanism (DEM) is a standard way to build PKE. Schemes built this way are often called "hybrid" PKE. For completeness, we describe the hybrid PKE built via KEM-DEM composition. Let KEM be a KEM, and DEM be an authenticated encryption scheme. (Below, we will use "DEM" and "AEAD" synonymously.) The hybrid PKE $\mathsf{PKE}^{\mathrm{hy}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is built as follows. The algorithm $\mathsf{PKE}^{\mathrm{hy}}.\mathsf{KGen}$ is the same as $\mathsf{KEM}.\mathsf{KGen}$. The algorithm $\mathsf{PKE}^{\mathrm{hy}}.\mathsf{Enc}$ takes as input the encapsulation key pk and a message m. It first runs $(C_0, \mathsf{k}) \leftarrow\!\!\$\ \mathsf{KEM}.\mathsf{Encap}(\mathsf{pk})$, the computes $C_1 \leftarrow\!\!\$\ \mathrm{AEAD}.\mathsf{Enc}(\mathsf{k}, m)$ and outputs ciphertext (C_0, C_1). The algorithm $\mathsf{PKE}^{\mathrm{hy}}.\mathsf{Dec}$ first uses sk to decapsulate C_0 and get k or possibly an error symbol $\perp$. Unless decapsulation failed, the algorithm completes by running $\mathrm{AEAD}.\mathsf{Dec}(\mathsf{k}, C_1)$, outputting either m or an error symbol $\perp$.

The Fujisaki-Okamoto transform. Classical results of Fujisaki and Okamoto [19–21] show how to amplify (in the random oracle model, or ROM) the security of public-key encryption, from one-wayness (OW) or indistinguishability (IND) under chosen-plaintext attack (CPA) to indistinguishability under chosen-ciphertext attack (IND-CCA). In this work we will mostly be interested in modern variants of this so-called "FO transform" studied first by Hofheinz et al. [24] in the classical ROM and QROM; extensions in the QROM were then given by [27, 38, 35]. Details of these transforms can be found in Section 4.

2.1 Security Definitions

Next we state several standard security notions which we will use below. In this work we use the "concrete" security paradigm, which explicitly measures the success probability and resource usage of specific adversaries, which we specify using the code-based game-playing framework of Bellare and Rogaway [10]. We will not relate quantities of interest, such as runtime or oracle queries, to a security parameter. We define relevant security notions for PKE (upper box), AEAD and MAC (lower box) in Figure 1.

PKE security notions are given for chosen-ciphertext attacks. All adversaries have access to a decryption oracle D that takes a ciphertext and (where relevant) a bit that selects which secret key to use. In ANO-CCA and IND-CCA games, the decryption oracle $D_{\not C}$ disallows queries for the challenge ciphertext. For each PKE notion, the corresponding definition for chosen-plaintext attacks can be obtained by simply removing the decryption oracle. In INT-CTXT, the adversary has an encryption (resp., decryption) oracle that takes associated data and a message (resp., ciphertext); flag win is set to true if the adversary submits a query to its decryption oracle that returns non-$\perp$, but was not returned from an encryption query. In SUF-CMA, the oracle TagO's inputs and outputs are stored in the table $\mathbf{T}$ after each query. In otROR-CCA, the oracles $E_1, \$_1$ are one-time encryption and random-bits oracles, respectively. The many-time security definition ROR-CCA is identical to otROR-CCA, but without this restriction. As for PKE above, CPA variants can be obtained by removing decryption oracles.

For any game G in Figure 1, we define an associated advantage measure for an adversary $\mathcal{A}$ and primitive P, denoted $\mathbf{Adv}^{G}_{P}(\mathcal{A})$, to be either $\Pr\left[G^{\mathcal{A}}_{P} \Rightarrow \mathsf{true}\right]$ or the absolute difference between that quantity and 1/2, if the game G is a bit-guessing game like IND-CCA.

3 Anonymity and Robustness of KEMs

In [33], Mohassel studied the anonymity and robustness of KEMs. However, all of his definitions and results apply only to the special case of KEMs that are constructed from PKE schemes in a restricted way, namely KEMs in which the encapsulation algorithm selects a random message for the PKE scheme and encrypts it using the PKE scheme's encryption algorithm. With this limitation, Mohassel provided a number of interesting results (positive and negative) concerning the anonymity and robustness of KEMs and of PKEs constructed from them via the KEM-DEM framework.

In this section, we bridge the definitional gap left by Mohassel's work by first considering fully general definitions for KEM anonymity and robustness, and then revisiting his results on these properties in the context of the KEM-DEM framework. As we shall see, how much can be recovered depends in a critical way on the KEM's behaviour with respect to rejection of invalid encapsulations.

We first define ANO-CCA security of a KEM $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ via the security game between an adversary and a challenger, as described in Figure 2. Note that the security game differs from the AI-ATK game defined for so-called *general encryption schemes* in [2], where in the latter, an adversary can have access to multiple public-keys (and some corresponding secret keys which will not result in a trivial win for the adversary). Since we are only considering PKE schemes and KEMs in this paper, it is not hard to show that the two security notions are equivalent up to a factor depending on the number of secret key queries an adversary could make (as already discussed in [2]).

An analogous ANO-CPA definition can be obtained simply by removing decapsulation queries in the above game. An adversary $\mathcal{A}$'s advantage in the ANO-{CPA,CCA} game is then defined to be:

$$\mathbf{Adv}^{\mathrm{ANO}-\{\mathrm{CPA},\mathrm{CCA}\}}_{\mathsf{KEM}}(\mathcal{A}) = |\Pr[\mathsf{G}^{\mathcal{A}} = 1] - 1/2|$$

where $\mathsf{G}^{\mathcal{A}}$ refers to $\mathcal{A}$ playing in the appropriate version of the anonymity game,

$\underline{\text{SROB-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen}$
$C \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$m_0 \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_0, \mathsf{sk}_0, C)$
$m_1 \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_1, \mathsf{sk}_1, C)$
return $m_0 \neq \perp$ AND $m_1 \neq \perp$

$\underline{\text{WROB-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen}$
$(m, b) \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$C \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}_b, m)$
$m_1 \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}, C)$
return $m_1 \neq \perp$

$\underline{\text{ANO-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen}$
$b \leftarrow\!\!\$\ \{0,1\}$
$(m, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$C \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}_b, m)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{D \not\in (\cdot,\cdot)}(C, \mathsf{st})$
return $b = b'$

$\underline{\text{SCFR-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen}$
$C \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$m_0 \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_0, \mathsf{sk}_0, C)$
$m_1 \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_1, \mathsf{sk}_1, C)$
return $m_0 = m_1 \neq \perp$

$\underline{\text{WCFR-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen}$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen}$
$(m, b) \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$C \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}_b, m)$
$m' \leftarrow \mathsf{PKE.Dec}(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}, C)$
return $m' = m \neq \perp$

$\underline{\text{IND-CCA}^{\mathcal{A}}_{\mathsf{PKE}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{KGen}$
$b \leftarrow\!\!\$\ \{0,1\}$
$(m_0, m_1, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}^{D(\cdot)}(\mathsf{pk})$
$C \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow\!\!\$\ \mathcal{A}^{D \not\in (\cdot)}(C, \mathsf{st})$
return $b = b'$

$\underline{\text{FROB}^{\mathcal{A}}_{\text{AEAD}}}$

$(C, \text{AD}, \mathsf{k}_0, \mathsf{k}_1) \leftarrow\!\!\$\ \mathcal{A}$
$m_0 \leftarrow \text{AEAD.Dec}(\mathsf{k}_0, \text{AD}, C)$
$m_1 \leftarrow \text{AEAD.Dec}(\mathsf{k}_1, \text{AD}, C)$
$b \leftarrow m_0 \neq \perp \wedge m_1 \neq \perp$
return $(b \wedge (\mathsf{k}_0 \neq \mathsf{k}_1))$

$\underline{\text{otROR-CCA}^{\mathcal{A}}_{\text{AEAD}}}$

$\mathsf{k} \leftarrow\!\!\$\ \text{AEAD.KGen}\ ;\ b \leftarrow\!\!\$\ \{0,1\}$
if $b = 0$ **then** $b' \leftarrow\!\!\$\ \mathcal{A}^{E_1(\cdot,\cdot), D(\cdot,\cdot)}$
else $b' \leftarrow\!\!\$\ \mathcal{A}^{\$_1(\cdot,\cdot), \perp(\cdot,\cdot)}$
return $b = b'$

$\underline{\text{XROB}^{\mathcal{A}}_{\text{AEAD}}}$

$(m_0, \mathsf{k}_0, R_0, \text{AD}_0, \mathsf{k}_1, \text{AD}_1, C_1) \leftarrow\!\!\$\ \mathcal{A}$
$C_0 \leftarrow \text{AEAD.Enc}(\mathsf{k}_0, m_0; R_0)$
$m_1 \leftarrow \text{AEAD.Dec}(\mathsf{k}_1, \text{AD}_1, C_1)$
$b \leftarrow m_0 \neq \perp \wedge m_1 \neq \perp$
$b_k \leftarrow \mathsf{k}_0 \neq \mathsf{k}_1$
$b_c \leftarrow C_0 = C_1 \neq \perp$
$b_a \leftarrow \text{AD}_0 = \text{AD}_1 \neq \perp$
return $(b \wedge b_k \wedge b_c \wedge b_a)$

$\underline{\text{INT-CTXT}^{\mathcal{A}}_{\text{AEAD}}}$

$\mathsf{k} \leftarrow\!\!\$\ \text{AEAD.KGen}$
$\mathsf{win} \leftarrow \mathsf{false}$
$\mathcal{A}^{E(\cdot,\cdot), D(\cdot,\cdot)}$
return win

$\underline{\text{SUF-CMA}^{\mathcal{A}}_{\mathsf{MAC}}}$

$\mathsf{k} \leftarrow\!\!\$\ \mathsf{MAC.KGen}$
$\mathbf{T} \leftarrow [\,]$
$(m, T) \leftarrow \mathcal{A}^{\text{TagO}(\cdot)}$
$b \leftarrow \mathsf{MAC.Vf}(\mathsf{k}, m, T)$
$b_t \leftarrow (m, T) \notin \mathbf{T}$
return $b \wedge b_t$

Fig. 1. Security games used in this paper. See Section 2.1 for details.

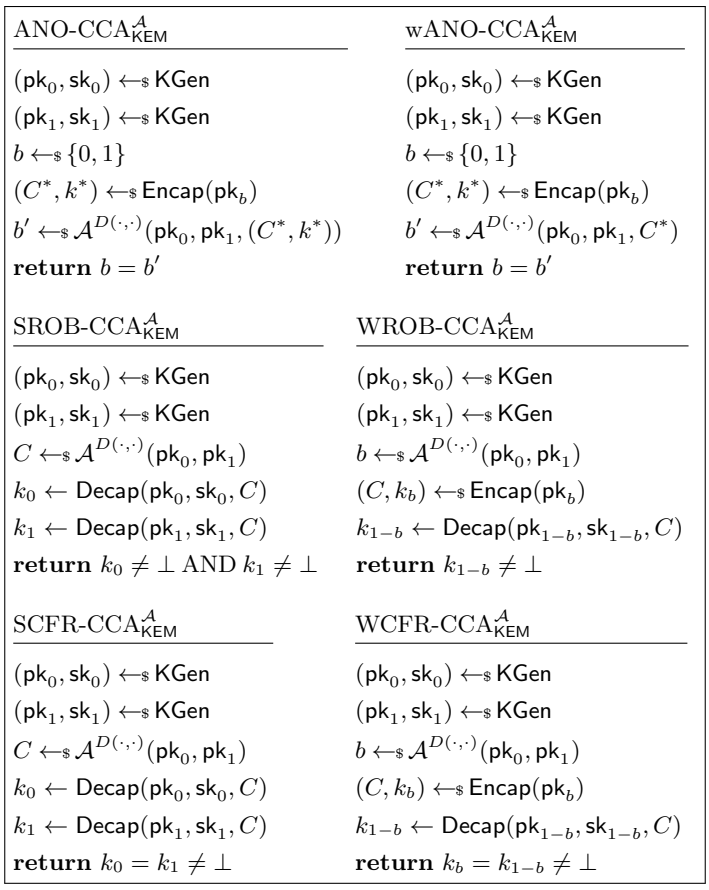


Fig. 2. KEM security notions for chosen-ciphertext attacks. All adversaries have access to a decryption oracle D that takes a ciphertext and (where relevant) a bit that selects which secret key to use. In ANO-CCA and wANO-CCA games, the decryption oracle disallows queries for the challenge ciphertext. For each notion, the corresponding definition for chosen-plaintext attacks can be obtained by simply removing the decryption oracle.

In the context of KEM-DEM framework for constructing PKE schemes, we will find it sufficient to work with an even weaker notion of anonymity for KEMs, that we refer to as *weak* anonymity. Here, the security game above is modified by giving the adversary only C^* in response to its challenge query, instead of (C^*, k^*); see Figure 2. We then refer to wANO-{CPA,CCA} security and define adversarial advantages as above.

We also define weak robustness (WROB) and strong robustness (SROB) security notions for general KEMs. The security games described in Figure 2 define both notions via two different finalisation steps. Note that the security game for WROB has a subtle difference from the corresponding WROB-ATK game defined for general encryption schemes in [2] (in addition to the fact that, in the latter game, an adversary can have access to multiple public-keys). The difference is that in our notion, an adversary outputs a bit b that determines which of the two public-keys $(\mathsf{pk}_0, \mathsf{pk}_1)$ will be used for encapsulation. This is required because the weak robustness notion is inherently *asymmetric* w.r.t. the two challenge public-keys, since one key is used for encapsulation (resp. encryption in case of PKE schemes) and the other for decapsulation (resp. decryption in case of PKE schemes).

Again, analogous WROB-CPA and SROB-CPA definitions can be obtained simply by removing decapsulation queries in the above games. The advantage of an adversary $\mathcal{A}$ in the {WROB,SROB}-{CPA,CCA} game is then defined as:

$$\mathbf{Adv}_{\mathsf{KEM}}^{\{\text{WROB,SROB}\}-\{\text{CPA,CCA}\}}(\mathcal{A}) = \Pr[\mathsf{G}^{\mathcal{A}} = 1]$$

where $\mathsf{G}^{\mathcal{A}}$ refers to $\mathcal{A}$ playing in the appropriate version of the robustness game.

Note that these robustness definitions apply mainly for KEMs that have explicit rejection on decapsulation errors. KEMs that offer only implicit rejection can never satisfy even the WROB-CPA notion.

With these anonymity and robustness notions in hand, it is straightforward to extend the result of [33, Claim 3.3] concerning anonymity preservation from the specific case of KEMs constructed directly from PKEs to fully general KEMs (with a non-zero decapsulation error probability); in fact, we can also show the robustness of hybrid PKE schemes constructed from robust KEMs via the KEM-DEM framework. Namely, we have the following:

Theorem 1. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen}, \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *with a one-time secure authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$*. If* KEM *is* δ*-correct, then:*

1. *For any* ANO-CCA *adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exist* wANO-CCA *adversary* $\mathcal{B}$*,* IND-CCA *adversary* $\mathcal{C}$ *and* WROB-CPA *adversary* $\mathcal{D}$ *against* KEM*, and* INT-CTXT *adversary* $\mathcal{E}$ *against* DEM *such that*

$$\begin{aligned}\mathbf{Adv}_{\mathsf{PKE}^{hy}}^{\text{ANO-CCA}}(\mathcal{A}) \leq & \mathbf{Adv}_{\mathsf{KEM}}^{\text{wANO-CCA}}(\mathcal{B}) + 2\mathbf{Adv}_{\mathsf{KEM}}^{\text{IND-CCA}}(\mathcal{C}) \\ & + \mathbf{Adv}_{\mathsf{KEM}}^{\text{WROB-CPA}}(\mathcal{D}) + \mathbf{Adv}_{\mathsf{DEM}}^{\text{INT-CTXT}}(\mathcal{E}) + \delta\,.\end{aligned}$$

   The running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{E}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{D}$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

2. *For any* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exists* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{B}$ *against* KEM *such that*

$$\begin{aligned}\mathbf{Adv}_{\mathsf{PKE}^{hy}}^{\text{WROB-ATK}}(\mathcal{A}) &\leq \mathbf{Adv}_{\mathsf{KEM}}^{\text{WROB-ATK}}(\mathcal{B})\,, \\ \mathbf{Adv}_{\mathsf{PKE}^{hy}}^{\text{SROB-ATK}}(\mathcal{A}) &\leq \mathbf{Adv}_{\mathsf{KEM}}^{\text{SROB-ATK}}(\mathcal{B})\,,\end{aligned}$$

   where $\text{ATK} \in \{\text{CPA}, \text{CCA}\}$ *and the running time of* $\mathcal{B}$ *is that of* $\mathcal{A}$*.*

Proof (sketch). The proof of Theorem 1.1 closely follows that of [33, Claim 3.3] in terms of the sequence of game-hops. Also for certain game-hops, we rely on security notions that are weaker than the corresponding notions considered in the proof of [33, Claim 3.3] (e.g., WROB-CPA, instead of WROB-CCA, security of the underlying KEM). The full details of the proof can be found in Appendix C.1.

To sketch a proof for Theorem 1.2, note that an adversary $\mathcal{A}$ wins the WROB-ATK game w.r.t. PKE^{hy} if it returns a pair (m, b) such that $\mathsf{Dec}^{hy}(sk_{1-b}, C) \neq \perp$ where $C(= (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})) \leftarrow_{\$} \mathsf{Enc}^{hy}(pk_b, m)$. Let $(C_{\mathsf{KEM}}, k_b) \leftarrow_{\$} \mathsf{Encap}(\mathsf{pk}_b)$ and $\mathsf{Decap}(\mathsf{sk}_{1-b}, C_{\mathsf{KEM}}) = k_{1-b}$. It is easy to see that $k_{1-b} \neq \perp$, since

$\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, C) \neq \perp$. This implies that we can return bit b to win the WROB-ATK game w.r.t. KEM. We can use a similar argument for the SROB-ATK case as well. The full details can again be found in Appendix C.1.

Note that Theorem 1 is only meaningful for KEMs with explicit rejection, since for implicit rejection KEMs, the term $\mathbf{Adv}^{\text{WROB-ATK}}_{\mathsf{KEM}}(\cdot)$ in the above security bounds can be large.

3.1 Generic Composition for Implicit Rejection KEMs

Robustness: We first consider what can be said about robustness for PKE schemes built from KEMs offering implicit rejection. We begin with a relaxed notion of robustness, namely *collision freeness* (as introduced for the specific case of KEMs obtained from PKEs in [33]). Informally, a scheme is said to be collision-free if a ciphertext always decrypts to two *different* messages under two different secret keys. We consider two variants, weak (WCFR) and strong collision freeness (SCFR). The security games defined in Figure 2 define both notions via two different finalisation steps.

As usual, analogous WCFR-CPA and SCFR-CPA definitions can be obtained by removing decapsulation queries in the above games. Adversary A's advantage in the {WCFR,SCFR}-{CPA,CCA} game is defined to be:

$$\mathbf{Adv}^{\{\text{WCFR,SCFR}\}-\{\text{CPA,CCA}\}}_{\mathsf{KEM}}(\mathcal{A}) := \Pr[\mathsf{G}^{\mathcal{A}} = 1]$$

where G^{A} refers to A playing in the appropriate version of the CFR game.

Now suppose we have a KEM that is SCFR-CCA (resp. WCFR-CCA) secure and a DEM that is FROB (resp. XROB) secure. (Recall that FROB and XROB are robustness notions for symmetric encryption schemes introduced in [18] and defined in Figure 1.) Then we can show that the hybrid PKE scheme obtained by composing these KEM and DEM schemes is SROB-CCA (resp. WROB-CCA) secure. More formally,

Theorem 2. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen}, \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *with a DEM* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$*. Then for any* SROB-CCA *(resp.* WROB-CCA*) adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exist* SCFR-CCA *(resp.* WCFR-CCA*) adversary* $\mathcal{B}$ *againt* KEM *and* FROB *(resp.* XROB*) adversary* $\mathcal{C}$ *against* DEM *such that*

$$\mathbf{Adv}^{\text{SROB-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + \mathbf{Adv}^{\text{FROB}}_{\mathsf{DEM}}(\mathcal{C}),$$
$$\mathbf{Adv}^{\text{WROB-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{WCFR-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + \mathbf{Adv}^{\text{XROB}}_{\mathsf{DEM}}(\mathcal{C}),$$

where the running times of $\mathcal{B}$ *and* $\mathcal{C}$ *are the same as that of* $\mathcal{A}$*.*

Proof (sketch). Note that an adversary $\mathcal{A}$ wins the SROB-CCA game w.r.t. PKE^{hy} if it returns a ciphertext C $(= (C_{\mathsf{KEM}}, C_{\mathsf{DEM}}))$ such that $\mathsf{Dec}^{hy}(sk_0, C) \neq \perp$ and $\mathsf{Dec}^{hy}(sk_1, C) \neq \perp$. Let $\mathsf{Decap}(\mathsf{sk}_0, C_{\mathsf{KEM}}) = k_0$ and $\mathsf{Decap}(\mathsf{sk}_1, C_{\mathsf{KEM}}) = k_1$. It is easy to see that $k_0 \neq \perp$ and $k_1 \neq \perp$. Now if $k_0 = k_1$, we can return C_{KEM} to win the SCFR-CCA game w.r.t. KEM. If $k_0 \neq k_1$, we can return $(C_{\mathsf{DEM}}, k_0, k_1)$ to win the FROB game w.r.t. DEM. We can do a similar case-distinction to argue about WROB-CCA security as well. The full details of the proof can be found in Appendix C.2.

Note that Farshim et al. [18] provide efficient constructions of FROB- and XROB-secure AE schemes, meaning that the requirements for the above theorem can be easily met. At the same time, they showed that a symmetric AE scheme that achieves the standard ROR-CCA notion of security is also inherently robust, albeit w.r.t. some weaker notions compared to FROB. Namely, such ROR-CCA secure AE schemes were shown to satisfy the so-called *semi-full robustness* (SFROB) notion in [18]. The SFROB notion of robustness for symmetric AE schemes is a (potentially) weaker variant of FROB where, in the corresponding security game, the adversary does not get to choose any keys. Instead, two keys are honestly generated and the adversary is given oracle access to encryption and decryption algorithms under both keys. The adversary is also given access to one of the keys, and the game is won (similar to that of FROB) if the adversary returns a ciphertext that decrypts correctly under both honestly generated keys.

The following theorem shows that a DEM that is only ROR-CCA secure – and that lacks the stronger robustness properties from [18] – is incapable of *generically* transforming strongly collision-free implicit rejection KEMs to strongly robust hybrid PKEs.

Theorem 3. *Suppose there exists a KEM that is simultaneously* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure. Suppose that there exists a* SUF-CMA*-secure MAC scheme and an* ROR-CPA *secure symmetric encryption scheme (such schemes can be built assuming only the existence of one-way functions). Suppose also that collision-resistant hash functions exist. Then there exists an implicit-rejection KEM that is* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure and a DEM that is* ROR-CCA *secure, such that the hybrid PKE scheme obtained from their composition is not* SROB-CCA *secure.*

Proof (sketch). Let $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vf})$ be an SUF-CMA secure MAC. We construct $\overline{\mathsf{MAC}} = (\overline{\mathsf{Tag}}, \overline{\mathsf{Vf}})$ where the only difference from MAC is that we fix a "faulty" key $\overline{k}$ chosen uniformly at random from the original MAC key-space such that $\overline{\mathsf{Vf}}(\overline{k}, \cdot) = 1$. Note that $\overline{\mathsf{MAC}}$ is also SUF-CMA secure. So by composing $\overline{\mathsf{MAC}}$ with an ROR-CPA secure symmetric encryption scheme SE that *never* rejects invalid ciphertexts via the "Encrypt-then-MAC" construction, we get an AE-secure $\overline{\mathsf{DEM}}$. Now let $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM that is SCFR-CCA, IND-CCA and ANO-CCA secure, and H be a collision-resistant hash function with its range being the key-space of SE. We construct $\overline{\mathsf{KEM}} = (\mathsf{KGen}, \mathsf{Encap}, \overline{\mathsf{Decap}})$ where the only difference from KEM is that the ciphertext space is augmented by a "special" bitstring $\overline{c}$ such that $\overline{\mathsf{Decap}}(\mathsf{sk}, \overline{c}) = H(\mathsf{pk})||\overline{k}$, for any KEM key-pair $(\mathsf{pk}, \mathsf{sk})$. It is not hard to see that $\overline{\mathsf{KEM}}$ is also IND-CCA, ANO-CCA secure, and SCFR-CCA secure (relying on the collision-resistance of H). Now the composition of $\overline{\mathsf{KEM}}$ and $\overline{\mathsf{DEM}}$ will not result in an SROB-CCA secure hybrid PKE. Specifically, an adversary can return the ciphertext $(\overline{c}, c'||\sigma')$, where $c'||\sigma'$ is an arbitrary $\overline{\mathsf{DEM}}$ ciphertext, to win the corresponding SROB-CCA game with probability 1. Full details of the proof can be found in Appendix C.3.

Anonymity: Now we turn to the question of what can be said about anonymity for PKE schemes built from KEMs offering implicit rejection. We prove a negative result that strengthens an analogous result of [33]. That result showed that there exist KEMs that are ANO-CCA (and IND-CCA) secure and XROB-secure authenticated encryption schemes, such that the hybrid PKE scheme resulting from their composition is *not* ANO-CCA secure. Thus anonymity is not preserved in the hybrid construction. However the KEM construction that was used to show this negative result in [33] is not SCFR-CCA secure, which might lead one to think that the strong collision freeness of implicit rejection KEMs might be sufficient to preserve anonymity. Here, we show this not to be true.

Theorem 4. *Suppose there exists a KEM that is simultaneously* SROB-CCA*,* IND-CCA *and* ANO-CCA *secure, a claw-free pair of permutations with domain and range being the encapsulated key-space of the KEM, and a collision-resistant hash function. Suppose also that there exists a DEM that is* ROR-CCA *and* XROB*-secure. Then there exists an implicit-rejection KEM that is* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure and a DEM that is* ROR-CCA *and* XROB*-secure, such that the resulting hybrid PKE is not* ANO-CCA *secure.*

Proof (sketch). Let $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM that is IND-CCA, ANO-CCA and SROB-CCA secure. Let H be a collision-resistant hash function that maps the space of public-keys of KEM to its encapsulated key-space. We now construct $\overline{\mathsf{KEM}} = (\mathsf{KGen}, \overline{\mathsf{Encap}}, \overline{\mathsf{Decap}})$ as follows. For the public parameters of $\overline{\mathsf{KEM}}$, we first generate a pair of claw-free permutations with corresponding fixed public-key PK (see [12, Section 4.2] for a more formal definition) $f_1(\mathsf{PK}, \cdot)$ and $f_2(\mathsf{PK}, \cdot)$ with domain and range being the encapsulated key-space of KEM. Now $\overline{\mathsf{Encap}}(\mathsf{pk})$ returns $(C, \overline{k})$ where $(C, k) \leftarrow\!\!\$\ \mathsf{Encap}(\mathsf{pk})$ and $\overline{k} := f_1(\mathsf{PK}, k)$. $\overline{\mathsf{Decap}}(\mathsf{sk}, C)$ returns $\overline{k}'$ where, for $k' \leftarrow \mathsf{Decap}(\mathsf{sk}, C)$, $\overline{k}' := f_1(\mathsf{PK}, k')$ if $k' \neq \perp$ and $\overline{k}' := f_2(\mathsf{PK}, H(\mathsf{pk}))$ if $k' = \perp$. Using straightforward reductions, it is not hard to show that $\overline{\mathsf{KEM}}$ is also IND-CCA and ANO-CCA secure. In addition, we can show that $\overline{\mathsf{KEM}}$ is SCFR-CCA secure by relying on the SROB-CCA security of KEM, collision-resistance of H and claw-freeness assumption w.r.t. $f_1(\mathsf{PK}, \cdot)$ and $f_2(\mathsf{PK}, \cdot)$.

Now let $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ be an ROR-CCA secure AEAD which is additionally XROB-secure. We now describe an adversary $\mathcal{A}$ against the ANO-CCA security of the hybrid PKE scheme w.r.t. the composition of $\overline{\mathsf{KEM}}$ and DEM. Upon receiving two public-keys pk_0 and pk_1 (along with the public-parameters $f_1(\mathsf{PK}, \cdot)$ and $f_2(\mathsf{PK}, \cdot)$), $\mathcal{A}$ selects an arbitrary message m and forwards m to the ANO-CCA challenger. It then receives the ciphertext $C = (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})$ where $(C_{\mathsf{KEM}}, k) \leftarrow\!\!\$\ \overline{\mathsf{Encap}}(\mathsf{pk}_b)$ and $C_{\mathsf{DEM}} \leftarrow\!\!\$\ \mathsf{Enc}(k, m)$, for bit $b \leftarrow\!\!\$\ \{0, 1\}$. Then, $\mathcal{A}$ asks for the decryption of ciphertext $C' = (C_{\mathsf{KEM}}, C'_{\mathsf{DEM}})$ w.r.t. sk_0 where $C'_{\mathsf{DEM}} = \mathsf{Enc}(\hat{k}, m)$ with $\hat{k} = f_2(\mathsf{PK}, H(\mathsf{pk}_0))$. If the response is $\perp$, then $\mathcal{A}$ outputs 0; else, it outputs 1. We use similar arguments

$\mathsf{Encap}(\mathsf{pk})$		$\mathsf{Decap}(\mathsf{sk}, c)$	
1 :	$m \leftarrow_{\$} \mathcal{M}$	1 :	Parse $c = (c_1, c_2)$
2 :	$c_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, m; G(m))$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c_1)$
3 :	$c_2 \leftarrow H'(m)$	3 :	$c_1' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; G(m'))$
4 :	$\boxed{c_2 \leftarrow H'(m, c_1)}$	4 :	**if** $c_1' = c_1 \wedge H'(m') = c_2$ **then**
5 :	$c \leftarrow (c_1, c_2)$	5 :	$\boxed{\textbf{if } c_1' = c_1 \wedge H'(m', c_1) = c_2 \textbf{ then}}$
6 :	$k = H(m, c)$	6 :	**return** $H(m', c)$
7 :	**return** (c, k)	7 :	**else return** $\perp$

Fig. 3. The KEM $\mathsf{HFO}^{\perp}[\mathsf{PKE}, G, H, H']$. Boxed code shows modifications to $\mathsf{HFO}^{\perp}[\mathsf{PKE}, G, H, H']$ required to obtain scheme $\mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$. Both constructed schemes reuse algorithm KGen from PKE.

as that of [33, Claim 3.1] to show that $\mathcal{A}$ succeeds with a high probability. Full details of the proof can be found in Appendix C.4.

The consequence of the above theorem (and its counterexample) is that, for implicit rejection KEMs, we cannot hope to transfer anonymity properties of the KEM to those of the hybrid PKE scheme resulting from the standard KEM-DEM construction in a fully generic manner. To make further progress in this direction, then, we need to look more closely at specific KEM constructions.

4 Anonymity and Robustness of KEMs Obtained from Fujisaki-Okamoto Transforms in the QROM

Fujisaki and Okamoto [19–21] introduced generic transformations that turn weakly secure PKE schemes (e.g. OW-CPA or IND-CPA secure PKE schemes) into IND-CCA secure KEMs and PKE schemes. Several distinct transforms have emerged, each with slightly different flavours; we broadly follow the naming conventions in [24]. One main distinction is whether the constructed KEM offers implicit rejection ($\mathsf{FO}^{\not\perp}$) or explicit rejection ($\mathsf{QFO}_m^{\perp}$). As we have already seen, this distinction is important in considering robustness, and we divide our analysis of the FO transforms in the same way. Since all NIST PQC candidates in the KEM/PKE category except one alternate candidate offer implicit rejection, we mainly focus on the corresponding $\mathsf{FO}^{\not\perp}$ transform. Also, since we are mainly concerned with the post-quantum setting, our analysis that follows will be in the QROM.

4.1 KEMs With Explicit Rejection

Before we focus on the $\mathsf{FO}^{\not\perp}$ transform, we briefly discuss our results related to explicit-rejection KEMs. The paper [28] presents a variant of the Fujisaki-Okamato transform, namely $\mathsf{HFO}^{\perp}$, that results in IND-CCA secure KEMs in the QROM. Given a PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ (with message space $\mathcal{M}$) and hash functions G, H and H', the resulting $\mathsf{KEM}^{\perp} = \mathsf{HFO}^{\perp}[\mathsf{PKE}, G, H, H'] = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ is described in Figure 3.

We introduce a slight variant of the above transform, namely $\mathsf{HFO}^{\perp'}$, as shown in Figure 3. The only change is that the c_2 component of the ciphertext–used for so-called *plaintext confirmation*–is derived as $c_2 \leftarrow H'(m, c_1)$ instead of as $c_2 \leftarrow H'(m)$. However, this seemingly minor change not only allows the $\mathsf{HFO}^{\perp'}$ transform to result in IND-CCA secure KEMs, but also strongly anonymous (ANO-CCA secure) and robust (SROB-CCA secure) KEMs in the QROM. In Appendix D, we formally state and prove the corresponding theorems.

4.2 KEMs With Implicit Rejection

Given a PKE scheme $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and hash functions G and H, the KEM $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ is shown in Figure 4. As described in [24], the $\mathsf{FO}^{\not\perp}$ transform "implicitly"

KGen'	$\mathsf{Encap}(\mathsf{pk})$	$\mathsf{Decap}(\mathsf{sk}', c)$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	1 : $m \leftarrow_{\$} \mathcal{M}$	1 : Parse $\mathsf{sk}' = (\mathsf{sk}, s)$
2 : $s \leftarrow_{\$} \mathcal{M}$	2 : $r \leftarrow G(m)$	2 : $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 : $\mathsf{sk}' = (\mathsf{sk}, s)$	3 : $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$	3 : $r' \leftarrow G(m')$
4 : **return** $(\mathsf{pk}, \mathsf{sk}')$	4 : $k \leftarrow H(m, c)$	4 : $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
	5 : **return** (c, k)	5 : **if** $c' = c$ **then**
		6 : **return** $H(m', c)$
		7 : **else return** $H(s, c)$

Fig. 4. The KEM $\mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$.

uses a modular transformation T that converts a OW-CPA/IND-CPA secure PKE scheme PKE into a *deterministic* PKE scheme $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G] = (\mathsf{KGen}, \mathsf{Enc}', \mathsf{Dec}')$ that is secure in the presence of so-called *plaintext-checking attacks*. The deterministic encryption $\mathsf{Enc}'(\mathsf{pk}, m)$ returns c where $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; G(m))$. The decryption $\mathsf{Dec}'(\mathsf{sk}, c)$ first computes $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$ and then returns m' if the *re-encryption* check "$\mathsf{Enc}(\mathsf{pk}, m'; G(m')) = c$" succeeds; otherwise, $\perp$ is returned.

It was proved in [27] that the $\mathsf{FO}^{\not\perp}$ transform lifts IND-CPA security of PKE to IND-CCA security of $\mathsf{KEM}^{\not\perp}$ in the QROM. We provide some further enhancement results for $\mathsf{FO}^{\not\perp}$. They demonstrate that, provided the starting PKE scheme PKE and the derived deterministic scheme PKE_1 satisfy some mild security assumptions on anonymity (wANO-CPA[3]) and collision-freeness (SCFR-CPA) respectively, then $\mathsf{FO}^{\not\perp}$ confers strong anonymity (ANO-CCA) and collision-freeness (SCFR-CCA) to the final $\mathsf{KEM}^{\not\perp}$ in the QROM. (In Appendix B, we present lemmas related to the QROM that are used in proving our results.)

Theorem 5. *Suppose* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct and has message space* $\mathcal{M}$*. Then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ *issuing at most* q_G *(resp.* q_H*) queries*[4] *to the quantum random oracle* G *(resp.* H*) and at most* q_D *queries to the (classical) decapsulation oracles, there exist* wANO-CPA *adversary* $\mathcal{B}$ *and* OW-CPA *adversary* $\mathcal{C}$ *against* PKE*, and* SCFR-CPA *adversary* $\mathcal{D}$ *against* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ *issuing at most* q_G *queries to* G*, such that:*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})} + q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D + 2)\sqrt{2\delta}\,.$$

Moreover, the running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{D}$ *are the same as that of* $\mathcal{A}$*.*

Proof (sketch). In a reduction from ANO-CCA security of $\mathsf{KEM}^{\not\perp}$ to wANO-CPA security of PKE, note that we need to simulate two different decapsulation oracles consistently without possessing the corresponding secret keys. Our approach is to generalize the simulation trick of [27, 35] in the QROM from a single-key setting (in the context of IND-CCA security) to a two-key setting (ANO-CCA). Namely, given two public-keys pk_0, pk_1, note that the encapsulation algorithm for both of them uses a common key-derivation function (KDF) "$k = H(m, c)$" (see Fig. 4). So we associate this KDF with two *secret* random functions H_0 and H_1 as follows: given an input (m, c), if $c = \mathsf{Enc}(\mathsf{pk}_i, m; G(m))$ (i.e., c results likely from $\mathsf{Encap}(\mathsf{pk}_i)$), then replace the KDF with "$k = H_i(c)$". Note that in this case, we can simply simulate the decapsulation oracles as $\mathsf{Decap}(\mathsf{sk}_i, c) = H_i(c)$ without requiring the secret keys. Now to argue that this replacement of KDF is indistinguishable w.r.t. an adversary, we require the functions $\mathsf{Enc}(\mathsf{pk}_i, \cdot\ ; G(\cdot))$ to be injective.

[3] The wANO-CPA security notion for PKE is a weaker variant of ANO-CPA where, in the corresponding security game, the challenger encrypts a uniformly random *secret* message under either of the two honestly generated public-keys and *only* provides the resulting ciphertext to the adversary, along with the generated public-keys.

[4] Following [24, 27], we make the convention that the number q_O of queries made by an adversary $\mathcal{A}$ to a random oracle O counts the total number of times O is executed in the corresponding security experiment; i.e., the number of $\mathcal{A}$'s explicit queries to O plus the number of implicit queries to O made by the experiment.

Thus, following [27], we first replace oracle G with G' where G' only returns "good" encryption randomness w.r.t. $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$ – i.e., $\forall m, \mathsf{Dec}(sk_i, \mathsf{Enc}(\mathsf{pk}_i, m; G'(m))) = m$, for $i \in \{0, 1\}$. We again generalize the argument of [27] from a single-key setting to a two-key setting to show that this replacement of G is indistinguishable, relying on the δ-correctness of PKE.

However, note that we additionally have to account for pairs (m, c) which satisfy $\mathsf{Enc}(\mathsf{pk}_0, m; G'(m)) = \mathsf{Enc}(\mathsf{pk}_1, m; G'(m)) = c$; in this case, the reduction does not know which public-key was used to generate c during key-encapsulation. So we rely on SCFR-CPA security to argue that it is computationally hard for an adversary to ask for the (classical) decapsulation of such "peculiar" ciphertexts c. Such a c results in $\mathsf{Dec}(sk_0, c) = \mathsf{Dec}(sk_1, c) = m$, thereby breaking the SCFR-CPA security of $\mathsf{T}[\mathsf{PKE}, G']$, and hence, that of $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ (up to an additive loss). Full details of the proof can be found in Appendix C.5. Note that it is similar in structure to that of [27, Theorem 1] in terms of the sequence of game-hops. But for the sake of completeness, we provide a self-contained proof.

To establish strong collision-freeness of the implicit-rejection KEMs constructed using $\mathsf{FO}^{\not\perp}$, we require the following *claw-freeness* property of quantum random oracles.

Lemma 1 ([41, Lemma 2.3]). *There is a universal constant α (< 648) such that the following holds: Let $\mathcal{X}_0$, $\mathcal{X}_1$ and $\mathcal{Y}$ be finite sets. Let $N_0 = |\mathcal{X}_0|$ and $N_1 = |\mathcal{X}_1|$, with $N_0 \leq N_1$. Let $H_0 : \mathcal{X}_0 \to \mathcal{Y}$ and $H_1 : \mathcal{X}_1 \to \mathcal{Y}$ be two random oracles.*

If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H_0 and H_1 at most q times, then we have

$$\Pr[H_0(x_0) = H_1(x_1) : (x_0, x_1) \leftarrow \mathcal{A}^{H_0, H_1}] \leq \frac{\alpha(q+1)^3}{|\mathcal{Y}|},$$

where all oracle accesses of $\mathcal{A}$ can be quantum.

For the following result, we in-fact need a weaker property than the one described in the above lemma; namely, it's hard for an adversary to return a value $x \in \mathcal{X}_0 \cap \mathcal{X}_1$ such that $H_0(x) = H_1(x)$. We leave the derivation of the corresponding upper-bound as an open problem.

Theorem 6. *Suppose $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is δ-correct. Then for any* SCFR-CCA *adversary $\mathcal{A}$ against $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ issuing at most q_D queries to the (classical) decapsulation oracles, at most q_G (resp. q_H) queries to the quantum random oracle G (resp. H), there exists an* SCFR-CPA *adversary $\mathcal{B}$ against $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ issuing at most q_G queries to G such that*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{\alpha(q_H+1)^3}{|\mathcal{K}|} + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D+2)\sqrt{2\delta}\,.$$

Here $\mathcal{K}$ denotes the encapsulated key-space of $\mathsf{KEM}^{\not\perp}$ and α (< 648) is the constant from Lemma 1. The running time of $\mathcal{B}$ is the same as that of $\mathcal{A}$.

Proof (sketch). Here we reduce the SCFR-CCA security of $\mathsf{KEM}^{\not\perp}$ to the hardness of claw-finding w.r.t. QROs. The proof is similar in structure to that of Theorem 5. Namely, we start with an SCFR-CCA adversary $\mathcal{A}$ and do a similar sequence of game-hops until the point where the decapsulation oracles don't require the corresponding secret keys – namely, $\mathsf{Decap}(\mathsf{sk}_i, c) = H_i(c)$ for (secret) random functions $H_0, H_1 : \overline{\mathcal{C}} \to \mathcal{K}$, where $\overline{\mathcal{C}}$ denotes the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$. Now $\mathcal{A}$ wins this modified SCFR-CCA game if it returns c such that $\mathsf{Decap}(\mathsf{sk}_0, c) = \mathsf{Decap}(\mathsf{sk}_1, c)$, or equivalently, $H_0(c) = H_1(c)$. Note that (c, c) is then a *claw* w.r.t. the pair of QROs (H_0, H_1). Hence, we can bound $\mathcal{A}$'s winning probability using Lemma 1. A complete proof can be found in Appendix C.6.

From Theorems 5 and 6, we see that by applying the $\mathsf{FO}^{\not\perp}$ transformation to weakly secure (i.e., OW-CPA) and weakly anonymous (i.e., wANO-CPA) PKE schemes, with an additional assumption of strong collision-freeness (against chosen plaintext attacks) of the deterministic version of the underlying PKE scheme (PKE_1 = $\mathsf{T}[\mathsf{PKE}, G]$), not only do we obtain strongly secure KEMs (i.e., IND-CCA security) but also KEMs that

are strongly anonymous (i.e., ANO-CCA) and are strongly collision-free against chosen ciphertext attacks (SCFR-CCA) in the QROM.

At the same time, we showed a negative result in Theorem 4. It essentially shows that starting from a KEM that is IND-CCA, ANO-CCA and SCFR-CCA secure does not *generically* result in a strongly anonymous (ANO-CCA) hybrid PKE scheme via the KEM-DEM composition. Nonetheless, we are able to show the following positive result for KEMs obtained via the $\mathsf{FO}^{\not\perp}$ transform. We only need a weak additional property of the underlying PKE scheme, namely that it be γ-spread (as defined in Appendix A).

Theorem 7. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen}', \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ *with a one-time authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$*. Suppose* PKE *is* δ*-correct and* γ*-spread (with message space* $\mathcal{M}$*). Then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* PKE^{hy} *issuing at most* q_G *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*), there exist* ANO-CCA *adversary* $\mathcal{B}$ *and* IND-CCA *adversary* $\mathcal{C}$ *against* $\mathsf{KEM}^{\not\perp}$*,* WCFR-CPA *adversary* $\mathcal{D}$ *against* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$*, and* INT-CTXT *adversary* $\mathcal{E}$ *against* DEM *such that:*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{C}) + \mathbf{Adv}^{\text{WCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2^{-\gamma}\,.$$

Moreover, the running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{E}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{D}$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

Proof (sketch). We use the proof of Theorem 1. Let $(\mathsf{pk}_0, \mathsf{sk}'_0)$ and $(\mathsf{pk}_1, \mathsf{sk}'_1)$ be two key-pairs generated in the ANO-CCA security game w.r.t. PKE^{hy}, and $b \leftarrow_{\$} \{0,1\}$ be the challenge bit. Let $c^* = (c_1^*, c_2^*)$ be the challenge ciphertext given to an adversary $\mathcal{A}$; i.e., $(c_1^*, k^*) \leftarrow \mathsf{KEM}^{\not\perp}.\mathsf{Encap}(\mathsf{pk}_b)$ and $c_2^* \leftarrow \mathsf{Enc}^{sym}(m)$ where m is chosen by $\mathcal{A}$ upon first receiving pk_0, pk_1. In the proof of Theorem 1, we make some initial game-hops to modify the $\mathsf{Dec}^{hy}(sk'_{1-b}, \cdot)$ oracle such that if the query is of the form (c_1^*, c_2), the oracle returns $\perp$. There we rely on the WROB-CPA security of the underlying KEM to justify this modification. However, $\mathsf{KEM}^{\not\perp}$ is trivially not WROB-CPA secure. Nevertheless, we show that by relying on γ-spreadness of PKE, WCFR-CPA security of PKE_1 and INT-CTXT security of DEM, we can still make the above modification of the $\mathsf{Dec}^{hy}(sk'_{1-b}, \cdot)$ oracle. From that point on, we essentially use the same game-hops as in the proof of Theorem 1 in our reduction to ANO-CCA security of $\mathsf{KEM}^{\not\perp}$. Full details can be found in Appendix C.7.

5 Anonymity and Robustness of NIST PQC Candidates

After analyzing the anonymity and robustness enhancing properties of the "standard" FO transforms in Section 4, we extend our analysis to the specific instantiations of these transforms used by Classic McEliece, proto-Saber (the simplified version of Saber in [15]) and FrodoKEM. We conclude this section by discussing some limitations of our techniques w.r.t. analyzing Saber and Kyber.

5.1 Classic McEliece

Classic McEliece (CM) as defined in its third round NIST specification [3] applies a slight variant of the $\mathsf{FO}^{\not\perp}$ transform to its starting deterministic PKE scheme (see Fig. 5). It can easily be shown that our generic transformation results on $\mathsf{FO}^{\not\perp}$, namely Theorems 5 and 6, apply to the $\mathsf{FO}^{\not\perp}$-like transformation used by CM, while accounting for the additional "Dent hash". Hence, the only thing that would remain to be analyzed is whether the base PKE scheme used by CM satisfies the pre-requisite security properties of Theorems 5 and 6, namely wANO-CPA and SCFR-CPA. As we show next, the base PKE scheme used by CM fails to be collision-free in a striking way that rules out the application of these results. This failure also propagates to PKE schemes built from the CM KEM via the standard KEM-DEM construction.

KGen'	$\mathsf{Encap}(\mathsf{pk})$	$\mathsf{Decap}(\mathsf{sk}', (c, h))$
1 : $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	1 : $m \leftarrow\!\!\$\ \mathcal{M}$	1 : Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, s)$
2 : $s \leftarrow\!\!\$\ \mathcal{M}$	2 : $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$	2 : $m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 : $\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}, s)$	3 : $h \leftarrow H_2(m)$	3 : $c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m')$
4 : **return** $(\mathsf{pk}, \mathsf{sk}')$	4 : $k \leftarrow H_1(m, (c, h))$	4 : **if** $c' = c \wedge H_2(m') = h$ **then**
	5 : **return** $((c, h), k)$	5 : **return** $H_1(m', (c, h))$
		6 : **else return** $H_0(s, (c, h))$

Fig. 5. Classic McEliece uses a slight variant of the $\mathsf{FO}^{\not\perp}$ transform that starts with deterministic PKE schemes. Here H_0 and H_1 are two different hash functions. The so-called "Dent hash" H_2 is used as an additional component in the KEM ciphertext [3].

The base CM scheme: The base CM scheme is deterministic. To encrypt a message m, first encode m as a binary column vector e of some fixed length n and fixed Hamming weight t. Then compute ciphertext $c = He \in \mathbb{F}_2^n$ where H is an $(n-k) \times n$ matrix of the form $H = (I_{n-k} \,|\, T)$, where T is some $(n-k) \times k$ matrix whose value is unimportant below. Matrix H is the parity check matrix of an error correcting code whose error correcting capacity is at least t. Decryption is done by using the private key to rewrite matrix H in such a way that efficient decoding can be performed to recover e with perfect correctness. The base CM scheme is closely related to the Niederreiter variant of the McEliece PKE scheme.

Collision-freeness of the base CM scheme: Recall that we would require the base CM scheme to satisfy the SCFR-CPA property in order to make use of our generic results concerning the $\mathrm{FO}^{\not\perp}$ transform. This property is crucial in the CPA $\rightarrow$ CCA security proofs where we have to simulate the decapsulation oracles under two different secret keys without access to the keys. As we will show now, the base CM scheme is not SCFR-CPA secure, nor even WCFR-CPA secure. In fact, we can go further and exhibit a strong robustness failure of the base CM scheme, and explain how it leads to robustness failures in the CM KEM and hybrid PKE schemes built from it.

Consider any weight t error vector e in which the t 1's in e are concentrated in the first $n-k$ bit positions of e (in all the parameter sets used in Classic McEliece, $n - k = mt \geq t$, for a positive integer m, so this is always possible). We call such an e *concentrated*. Note that any concentrated e can be written $e = \binom{e_{n-k}}{0_k}$ with e_{n-k} of length $n-k$ and 0_k being the vector of k zeros. Since encryption is done by computing $c = He$, and H is of the form $(I_{n-k} \,|\, T)$, it is easy to see that c is a fixed vector independent of the T part of H: namely, $He = e_{n-k}$ which depends only on the first $n-k$ bit positions of e.

Note that this property holds independent of the public key of the base CM scheme (which is effectively the matrix H). Thus there is a class of base CM messages (of size $\binom{n-k}{t}$) for which the resulting ciphertext c can be predicted as a function of the message *without even knowing the public key*. By correctness of the base CM scheme, such ciphertexts must decrypt to the selected message *under any base CM scheme private key*.

It is immediate that this property can be used to violate SCFR-CPA and WCFR-CPA security of the base CM scheme. This presents a significant barrier to the application of our general theorems for establishing robustness and anonymity of the full CM KEM.

Robustness of the CM KEM and Hybrid PKEs derived from it: The base CM scheme is used to construct the CM KEM according to procedure described in Figure 5. This means that the CM KEM encapsulations are also of the form $c = (He, H_2(e))$ where $H_2(\cdot)$ is a hash function; meanwhile the encapsulated keys are set as $H_1(e, c)$ where $H_1(\cdot)$ is another hash function. The CM KEM performs implicit rejection, so one cannot hope for robustness. However, one might hope for some form of collision-freeness. Our analysis above shows that the CM KEM does not provide even this, since when e is concentrated, $c = (He, H_2(e))$ decapsulates to $H_1(e, c)$ under any CM private key.

Finally, one might ask about the robustness of PKE scheme built by combining the CM KEM with a DEM in the standard way. Again, such a PKE cannot be strongly collision free (and therefore not strongly

robust either), since it is trivial using our observations to construct a hybrid PKE ciphertext that decrypts correctly under *any* CM private key to *any* fixed choice of message m (without even knowing the public key). To see this, simply consider hybrid ciphertexts of the form $(He, H_2(e), \mathsf{AEAD.Enc}(K, m; r))$ where e is concentrated, $K = H_1(e, c)$ is the symmetric key encapsulated by the KEM part $c = (He, H_2(e))$ of the hybrid ciphertext, and r is some fixed randomness for the AEAD scheme. Such ciphertexts decrypt to the freely chosen message m under any CM private key.

Robustness could plausibly be conferred on this hybrid PKE scheme by including a hash of the public key in the key derivation step. However CM keys are large, so this would have a negative effect on performance. Robustness is *not* conferred in general by replacing the DEM with an AEAD scheme and including the hash of the public key in the associated data to create a "labelled DEM". This is easy to see by adapting the counter-example construction used in the proof of Theorem 3.

Further remarks on CM: The analysis above shows that we cannot hope to establish anonymity or robustness of the CM KEM or PKEs built from it via the standard KEM-DEM construction using the sequence of results in this paper. But this does not rule out more direct approaches to proving anonymity. For example, Persichetti [34] has analysed the anonymity of a scheme called HN (for "hybrid Niederreiter") that is rather close to the natural hybrid scheme one would obtain from CM. However, the analysis is in the ROM rather than the QROM. We are not aware of any further analysis of the anonymity properties of schemes that are close to CM and that might be easily adapted to CM.

In the context of the NIST PQC process, it remains an important open problem to establish anonymity of the CM scheme.

5.2 proto-Saber

KGen'		$\mathsf{Encap}(\mathsf{pk})$		$\mathsf{Decap}(\mathsf{sk}', c)$	
1 :	$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	1 :	$m \leftarrow\!\!\$\ \mathcal{M}$	1 :	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, F(\mathsf{pk}), s)$
2 :	$s \leftarrow\!\!\$\ \mathcal{M}$	2 :	$(\hat{k}, r) \leftarrow G(F(pk), m)$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
3 :	$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}, F(\mathsf{pk}), s)$	3 :	$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$	3 :	$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}), m')$
4 :	**return** $(\mathsf{pk}, \mathsf{sk}')$	4 :	$k \leftarrow H(\hat{k}, c)$	4 :	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
		5 :	**return** (c, k)	5 :	**if** $c' = c$ **then**
				6 :	**return** $H(\hat{k}', c)$
				7 :	**else return** $H(s, c)$

Fig. 6. pSaber uses a variant of the $\mathsf{FO}^{\not\perp}$ transform. Here G, F and H are hash functions.

The scheme "proto-Saber" (pSaber for short) is a KEM that was introduced in [15] and which is included in the NIST third round specification document for Saber [8]. Saber and pSaber use the same base PKE scheme but apply *different* FO-type transforms to obtain their respective KEMs. The QROM IND-CCA security claims for Saber [8, Theorem 6.5] seem to have been taken directly from those for pSaber [15, Theorem 6] without any modification. However, as we will explain below, there are issues with pSaber's IND-CCA security claims, and yet further issues for Saber's.

Now pSaber uses a transform that differs significantly from the standard $\mathsf{FO}^{\not\perp}$ one (see Fig. 6). These significant deviations act as an obstacle to applying our generic results on anonymity and SCFR enhancement of $\mathsf{FO}^{\not\perp}$ to pSaber. The nature of these deviations also led us to ask whether they also act as a barrier in applying the results of [27] to establish the IND-CCA security of pSaber, as claimed in [15]. We believe this to be the case, as we explain next.

IND-CCA security of pSaber in the QROM: We claim that the specific proof techniques used by [27], to obtain relatively tight IND-CCA security bounds for the standard $\mathsf{FO}^{\not\perp}$ transform in the QROM, do not

directly apply to pSaber's variant of the FO transform. An important trick used by [27] in their security proofs of $\mathsf{FO}^{\not\perp}$ is to replace the computation of the key "$k \leftarrow H(m, c)$" with "$k \leftarrow H'(g(m))(= H'(c))$" for function $g(\cdot) = \mathsf{Enc}(\mathsf{pk}, \cdot\,; G(\cdot))$ and a secret random function $H'(\cdot)$; note that in this case, we simply have $\mathsf{Decap}(\mathsf{sk}, c) = H'(c)$ leading to an "efficient" simulation of the decapsulation oracle without using the secret key sk. To justify this replacement, the authors of [27] then argue about the injectivity of $g(\cdot)$, relying on the correctness of the underlying PKE scheme to establish this.

But in pSaber, the keys are computed as "$k \leftarrow H(\hat{k}, c)$" where the "pre-key" $\hat{k}$ is derived as a hash of the message m (to be specific, $(\hat{k}, r) \leftarrow G(F(\mathsf{pk}), m)$). So there is an extra *layer* of hashing between m and the computation of k. Hence, to use a similar trick as [27], we would require some additional injectivity arguments. Thus, strictly speaking, the proof techniques of [27] do not directly apply to pSaber.

Nevertheless, we are able to overcome the above barrier by adapting the analysis of $\mathsf{FO}^{\not\perp}$ in [27] to obtain an explicit IND-CCA security proof for pSaber in the QROM, with the *same* tightness as claimed in [15]. The formal proof can be found in Appendix E. We give a high-level overview of our approach below.

First, note that we can replace the step "$(\hat{k}, r) \leftarrow G(F(\mathsf{pk}), m)$" in pSaber's encapsulation by "$\hat{k} \leftarrow G_{\hat{k}}(m)$" and "$r \leftarrow G_r(m)$" for two fresh random oracles $G_{\hat{k}}$, $G_r : \{0,1\}^{256} \rightarrow \{0,1\}^{256}$. Now our key observation is that the extra layer of hashing "$G_{\hat{k}}(\cdot)$" between m and k is actually *length-preserving*, i.e., the hash function has the same domain and range. So following [24, 38], we can replace the random oracle $G_{\hat{k}}(\cdot)$ with a random *polynomial* of degree $2q_G - 1$ over a finite field representation of $\{0,1\}^{256}$ (i.e., a $2q_G$-wise independent function). Here q_G is the number of queries made to oracle G in the IND-CCA security reduction for pSaber. Because of Lemma 2 in Appendix B, this change is perfectly indistinguishable to an adversary making at most q_G queries to $G_{\hat{k}}$. This will allow us to recover m from a corresponding pre-key value $\hat{k}$ by computing roots of the polynomial $G_{\hat{k}}(x) - \hat{k}$. Hence we can *invert* this "nested" hashing of m in order to apply the trick of [27]. Namely, we can now replace the key derivation "$k \leftarrow H(\hat{k}, c)$" with "$k \leftarrow H'(g(m))(= H'(c))$" for function $g(\cdot) = \mathsf{Enc}(\mathsf{pk}, \cdot\,; G_r(\cdot))$, where in addition, m is a root of the polynomial $G_{\hat{k}}(x) - \hat{k}$.

Anonymity and Robustness of pSaber in the QROM: Our approach to repairing pSaber's IND-CCA proof also allows us to derive proofs of anonymity and SCFR enhancement for pSaber with similar tightness.

Now pSaber, and Saber, is a KEM whose claimed security relies on the hardness of the module learning-with-rounding problem, or mod-LWR for short (see [8, 15] for a precise description of the assumption). In the following, we prove the ANO-CPA security of the base PKE scheme Saber.PKE that is used by pSaber, and also currently used by Saber (as per [8]). The result relies on the hardness of mod-LWR. The proof can be found in Appendix C.8. The proof adapts the proof of [15, Theorem 3] showing IND-CPA security of Saber.PKE.

Theorem 8. *For any* ANO-CPA *adversary* $\mathcal{A}$ *against* Saber.PKE*, there exists a distinguisher* $\mathcal{B}_1$ *(resp.,* $\mathcal{B}_2$*) between* l *(resp.* $l+1$*) samples from a mod-LWR distribution from that of a uniform distribution, with corresponding parameters* l, μ, q *and* p*, such that*

$$\mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \mathbf{Adv}^{\text{mod-lwr}}_{l+1,l,\mu,q,p}(\mathcal{B}_2).$$

Moreover, the running times of $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *are the same as that of* $\mathcal{A}$*.*

Now we establish anonymity and strong collision-freeness of pSaber KEM, which we will denote as "pSaber.KEM" in the following to contrast the scheme with Saber.PKE. We use similar proof strategies that were used to show the same properties for $\mathsf{FO}^{\not\perp}$ in Section 4 (Theorems 5 and 6). A major difference is that instead of relying on the SCFR-CPA security property of Saber.PKE (specifically, its deterministic version), we again rely on hardness of the *claw-finding* problem in a quantum setting (see Lemma 1).

In our next results, we show that the stronger properties of ANO-CCA and SCFR-CCA hold for pSaber.KEM. Below we define $\mathbf{Coll}^{F}_{\mathsf{Saber.PKE}}$ as the probability of the event "$F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$" where pk_0 and pk_1 are two honestly-generated Saber.PKE public-keys. Given the space of Saber's public-keys is sufficiently large (of size greater than 2^{256}), if the hash function F is sufficiently collision-resistant, then $\mathbf{Coll}^{F}_{\mathsf{Saber.PKE}}$ can be considered to be negligible. The proofs of Theorems 9 and 10 can be found in Appendices C.9 and C.10 respectively.

Theorem 9. *Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_D *classical queries to the decapsulation oracles, at most* q_G *(resp.* q_H*) quantum queries to the random oracle* G *(resp.* H*), there exist* ANO-CPA *adversary* $\mathcal{B}$*,* OW-CPA *adversary* $\mathcal{C}$ *against* $\mathsf{Saber.PKE}$ *and a distinguisher* $\mathcal{B}_1$ *between* l *samples from a mod-LWR distribution and a uniform distribution with corresponding parameters* l, μ, q *and* p*, such that*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{B}) + 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{C})} + \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \mathbf{Adv}^{mod\text{-}lwr}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{2}{2^{256}} + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta}$$

Here α (< 648) *is the constant from Lemma 1. The running times of* $\mathcal{B}$ *and* $\mathcal{C}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{B}_1$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

Theorem 10. *Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, for any* SCFR-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_D *queries to the (classical) decapsulation oracles, at most* q_G *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*), we have*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{\alpha(q_H+1)^3}{2^{256}} + \frac{4q_H}{2^{128}}$$

Here α (< 648) *is the constant from Lemma 1.*

Regarding hybrid PKE schemes obtained from $\mathsf{pSaber.KEM}$ via the KEM-DEM composition, we additionally show that such PKE schemes satisfy the stronger ANO-CCA notion of anonymity, in a similar vein to Theorem 7 w.r.t. $\mathsf{FO}^{\not\perp}$-based KEMs. The proof can be found in Appendix C.11.

Theorem 11. *Let* $\mathsf{pSaber.PKE}^{hy} = (\mathsf{KGen}', \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *with a one-time authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$*. Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{pSaber.PKE}^{hy}$ *issuing at most* q_G *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*), there exist* ANO-CCA *adversary* $\mathcal{B}$*,* IND-CCA *adversary* $\mathcal{C}$ *against* $\mathsf{pSaber.KEM}$*,* INT-CTXT *adversary* $\mathcal{E}$ *against* DEM *and distinguisher* $\mathcal{B}_1$ *between* l *samples from a mod-LWR distribution and a uniform distribution, with corresponding parameters* l, μ, q *and* p*, such that*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{C}) + \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \mathbf{Adv}^{mod\text{-}lwr}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta} + \frac{1}{2^{256}}$$

and the running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{E}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{B}_1$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

At the same time, from Theorems 2 and 10, we note that if the DEM component is also FROB secure, then the corresponding hybrid PKE scheme will be strongly robust (i.e., SROB-CCA secure). Hence, our above results give a complete picture of anonymity and robustness properties of pSaber as well as the hybrid PKE schemes derived from it.

5.3 FrodoKEM

FrodoKEM uses an *identical* FO-type transform, described as "$\mathsf{FO}^{\not\perp\prime}$" in the specification document [4], as pSaber does (see Fig. 6) on its base PKE scheme "FrodoPKE". Hence, our positive results on tight IND-CCA security, anonymity and robustness of pSaber should also apply to FrodoKEM in a similar fashion; instead of relying on hardness of mod-LWR problem, we have to rely on hardness of the learning-with-errors (LWE) problem.

For example, when it comes to establishing anonymity of FrodoKEM, we only need to prove the ANO-CPA security of FrodoPKE and then rely on the "ANO-CPA $\rightarrow$ ANO-CCA" enhancement property of $\mathsf{FO}^{\not\perp\prime}$ (LWE variant of Theorem 9). The ANO-CPA security of FrodoPKE can be shown in a similar manner as

that of Saber.PKE (Theorem 8): namely, by adapting the IND-CPA security proof of FrodoPKE. To be more precise, it is shown in [4, 31] w.r.t. FrodoPKE = (KGen, Enc, Dec) that given $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KGen}$ and *any* valid message m, the distribution $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m))$ is computationally indistinguishable from (pk, c^*) where c^* is a uniformly random ciphertext, relying on the LWE hardness assumption. Hence, in the ANO-CPA security game w.r.t. FrodoPKE, given two honestly-generated public-keys $\mathsf{pk}_0, \mathsf{pk}_1$ and a message m chosen by an adversary, it cannot distinguish the encryption of m under pk_0 from a uniformly random ciphertext that is independent of pk_0. Similarly, the adversary also cannot distinguish the uniformly random ciphertext from the encryption of m under pk_1. It follows that the adversary cannot distinguish between the encryptions of m under pk_0 and pk_1, thereby establishing the ANO-CPA security of FrodoPKE.

5.4 Saber and Kyber

It turns out that Saber and Kyber implement a transform that deviates *even further* from the $\mathsf{FO}^{\not\perp}$ transform than pSaber does (see Fig. 7). Specifically, the keys in Saber are computed as "$k \leftarrow F(\hat{k}, F(c))$" where the "pre-key" $\hat{k}$ is derived as a hash of the message m (to be specific, $(\hat{k}, r) \leftarrow G(F(\mathsf{pk}), m)$). Again there is an extra hashing step between m and the computation of k, as we have seen for pSaber. But at the same time, there is also a "nested" hashing of ciphertext in the key-derivation (i.e., Saber uses "$F(c)$" in place of just "c") as opposed to the standard "single" hashing in $\mathsf{FO}^{\not\perp}$ and pSaber.

KGen'	$\mathsf{Encap}(\mathsf{pk})$	$\mathsf{Decap}(\mathsf{sk}', c)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}$	$m \leftarrow_{\$} \mathcal{M}$	Parse $\mathsf{sk}' = (\mathsf{sk}, \mathsf{pk}, F(\mathsf{pk}), s)$
$s \leftarrow_{\$} \mathcal{M}$	$m \leftarrow F(m)$	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$
$\mathsf{sk}' \leftarrow (\mathsf{sk}, \mathsf{pk}, F(\mathsf{pk}), s)$	$(\hat{k}, r) \leftarrow G(F(pk), m)$	$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}), m')$
return $(\mathsf{pk}, \mathsf{sk}')$	$c \leftarrow \mathsf{Enc}(\mathsf{pk}, m; r)$	$c' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; r')$
	$k \leftarrow F(\hat{k}, F(c))$	**if** $c' = c$ **then**
	return (c, k)	**return** $F(\hat{k}', F(c))$
		else return $F(s, F(c))$

Fig. 7. Saber uses a variant of the $\mathsf{FO}^{\not\perp}$ transform. Here G and F are hash functions [8].

This "extra" hash of the ciphertext is a significant barrier to applying the techniques we used to prove anonymity of pSaber. It also acts as a barrier when trying to apply the generic proof techniques of [27] towards establishing the IND-CCA security of Saber in the QROM, with the *same* bounds as was claimed in its NIST third round specification [8]. At least for pSaber, as discussed above, we were able to account for the "nested" hashing of message because it was *length-preserving*. However, this is not the case for "$F(c)$" in Saber. We believe that an IND-CCA security reduction for Saber, along the lines of [27], in the QROM would need to rely on the collision-resistance of $F(\cdot)$ when modelled as a quantum random oracle. But a corresponding additive term is missing in the IND-CCA security bounds claimed in the Saber specification. We have shared these observations with the Saber team. A representative of the team [40] accepted our findings on the IND-CCA security of pSaber. Regarding Saber, they maintain that the nested hash of ciphertext $F(c)$ should not pose a security problem for Saber as c is "deterministically derived from limited entropy". However, they do not know if this allows a security proof to go through in the QROM [40].

When it comes to robustness however, the news is better. Namely, we can apply similar proof strategies used to establish strong collision-freeness of $\mathsf{FO}^{\not\perp}$-based KEMs (Theorem 6) and pSaber (Theorem 10) to show SCFR-CCA security of Saber in the QROM. The corresponding proof, presented in detail in Appendix C.12, on a high-level uses the fact that the hash of public-keys are included in Saber's key-derivation step (in contrast to Classic McEliece). This allows us to establish the SCFR-CCA security of Saber KEM by mainly relying on properties of quantum random oracles G and F, namely collision-resistance and claw-freeness.

Theorem 12. *For any* SCFR-CCA *adversary* $\mathcal{A}$ *against the scheme* $\mathsf{Saber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_G *(resp.* q_F*) queries to the quantum random oracle* G *(resp.* F*), we have*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{Saber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{4\alpha(q_F+1)^3}{2^{256}} + \frac{4q_F}{2^{128}}$$

Here α (< 648) *is the constant from Lemma 1.*

Kyber uses an FO-type transform which is essentially the same as that of Saber (see Fig. 7). Hence, the issues we identified with Saber above w.r.t. IND-CCA security claims in the QROM as described in the specification document, as well as establishing anonymity of the scheme, apply to Kyber too. We have shared these observations with the Kyber team. At the 3rd NIST PQC Standardization Conference, a representative of the Kyber team [37] acknowledged that the nested hash of ciphertext $F(c)$ could make it "tricky" to prove the security of Kyber in the QROM, while removing this nested hash would overcome this issue.

But on the positive side, our result on strong collision-freeness (SCFR-CCA security) of Saber–namely, Theorem 12 above–also applies to Kyber in the same fashion, because of the similarity in their respective FO-type transforms. In other words, the current versions of Kyber and Saber also lead to strongly robust hybrid PKE schemes in the QROM.

In conclusion, we consider the *concrete* IND-CCA security–as claimed in [8, 6]–and anonymity (ANO-CCA security) of Saber and Kyber to remain open. We also suggest a modification to Saber and Kyber: namely, to apply the *same* FO-type transform as pSaber uses (as in Figure 6) to the relevant base PKE scheme, thus replacing the "nested" hashing of ciphertext in key-derivation with single hashing. In doing so, not only would the two NIST finalists then enjoy the same provable IND-CCA security guarantees of $\mathsf{FO}^{\not\perp}$-based KEMs in the QROM as established in the literature [27, 35], but this would also allow our techniques establishing anonymity of pSaber to be extended to Saber and Kyber.[5]

6 Conclusions and Future Work

In this work, we initiated the study of anonymous and robust KEMs and PKE schemes in the post-quantum setting. We resolved several core technical questions, and showed that proto-Saber, a simplified version of Saber, and FrodoKEM can be used to build anonymous, robust hybrid PKE schemes. We also pointed out gaps in the current IND-CCA security analyses of Saber and Kyber. Both NIST finalists do lead to robust hybrid PKE from our analysis. Finally, we highlighted a surprising property of Classic McEliece (CM) showing that it does not lead to robust PKE schemes via the standard KEM-DEM construction.

Important questions remain about the anonymity and robustness of the NIST finalists and alternate candidates. For example, it is plausible that the anonymity of CM could be proven by a direct approach; the same applies for Saber and Kyber. Notable among the alternate schemes is SIKE, which uses radically different algebraic problems to build a KEM; extending our work to SIKE would be interesting. One broader question about post-quantum PKE which has not been widely studied is multi-receiver hybrid PKE (with or without anonymity/robustness). Such schemes would have applications in group-oriented end-to-end secure messaging.

Acknowledgements. It is our pleasure to thank the Classic McEliece, Kyber, Saber and FrodoKEM teams, along with Kathrin Hövelmanns and Keita Xagawa, for helpful discussions. We also thank the anonymous reviewers of Eurocrypt 2022 for their constructive comments and suggestions. Paterson's research was supported in part by a gift from VMware.

References

1. M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *CRYPTO 2005*, pages 205–222, 2005.

[5] For Kyber's anonymity, we would rely on the hardness of module learning-with-errors (mod-LWE) problem instead of mod-LWR, akin to our discussion on FrodoKEM; see Subsection 5.3.

2. M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *TCC 2010*, pages 480–497, 2010.
3. M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich, R. Misoczki, R. Niederhagen, K. G. Paterson, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier, J. Szefer, C. J. Tjhai, M. Tomlinson, and W. Wang. Classic McEliece: NIST round 3 submission, 2021.
4. E. Alkim, J. W. Bos, L. Ducas, P. Longa, I. Mironov, M. Naehrig, V. Nikolaenko, C. Peikert, A. Raghunathan, and D. Stebila. FrodoKEM: NIST round 3 submission, 2021.
5. A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483, 2014.
6. R. Avanzi, J. Bos, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Kyber: NIST round 3 submission, 2021.
7. A. Barth, D. Boneh, and B. Waters. Privacy in encrypted content distribution using private broadcast encryption. In *FC 2006*, pages 52–64, 2006.
8. A. Basso, J. M. B. Mera, J. D'Anvers, A. Karmakar, S. S. Roy, M. V. Beirendonck, and F. Vercauteren. Saber: NIST round 3 submission, 2021.
9. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT 2001*, pages 566–582, 2001.
10. M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT 2006*, pages 409–426, 2006.
11. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474, 2014.
12. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, pages 41–69, 2011.
13. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT 2001*, pages 93–118, 2001.
14. C. Chen, O. Danba, J. Hoffstein, A. Hülsing, J. Rijneveld, T. Saito, J. M. Schanck, P. Schwabe, W. Whyte, K. Xagawa, T. Yamakawa, and Z. Zhang. NTRU: NIST round 3 submission, 2021.
15. J.-P. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In *AFRICACRYPT 18*, pages 282–305, 2018.
16. Y. Dodis, P. Grubbs, T. Ristenpart, and J. Woodage. Fast message franking: From invisible salamanders to encryptment. In *CRYPTO 2018, Part I*, pages 155–186, 2018.
17. P. Farshim, B. Libert, K. G. Paterson, and E. A. Quaglia. Robust encryption, revisited. In *PKC 2013*, pages 352–368, 2013.
18. P. Farshim, C. Orlandi, and R. Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symm. Cryptol.*, 2017(1):449–473, 2017.
19. E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In *PKC'99*, pages 53–68, 1999.
20. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, pages 537–554, 1999.
21. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, 2013.
22. P. Grubbs, J. Lu, and T. Ristenpart. Message franking via committing authenticated encryption. In *CRYPTO 2017, Part III*, pages 66–97, 2017.
23. R. Hayashi and K. Tanaka. PA in the two-key setting and a generic conversion for encryption with anonymity. In *ACISP 06*, pages 271–282, 2006.
24. D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017, Part I*, pages 341–371, 2017.
25. K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. In *PKC 2020, Part II*, pages 389–422, 2020.
26. A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016, Part I*, pages 387–416, 2016.
27. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018, Part III*, pages 96–125, 2018.
28. H. Jiang, Z. Zhang, and Z. Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *PKC 2019, Part II*, pages 618–645, 2019.
29. J. Len, P. Grubbs, and T. Ristenpart. Partitioning oracle attacks. In *USENIX Security*, 2021.
30. B. Libert, K. G. Paterson, and E. A. Quaglia. Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model. In *PKC 2012*, pages 206–224, 2012.
31. R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA 2011*, pages 319–339, 2011.

32. C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J. Robert, P. Véron, and G. Zémor. HQC: NIST round 3 submission, 2021.
33. P. Mohassel. A closer look at anonymity and robustness in encryption schemes. In *ASIACRYPT 2010*, pages 501–518, 2010.
34. E. Persichetti. Secure and anonymous hybrid encryption from coding theory. In *Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013*, pages 174–187, 2013.
35. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018, Part III*, pages 520–551, 2018.
36. K. Sako. An auction protocol which hides bids of losers. In *PKC 2000*, pages 422–432, 2000.
37. P. Schwabe. Crystals-kyber round 3 presentation. 3rd NIST PQC Standardization Conference, 2021.
38. E. E. Targhi and D. Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *TCC 2016-B, Part II*, pages 192–216, 2016.
39. D. Unruh. Revocable quantum timed-release encryption. In *EUROCRYPT 2014*, pages 129–146, 2014.
40. F. Vercauteren. Private communication, 2021.
41. K. Xagawa. Ntru leads to anonymous, robust public-key encryption. Cryptology ePrint Archive, Report 2021/741, 2021. https://ia.cr/2021/741.
42. M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In *CRYPTO 2012*, pages 758–775, 2012.
43. M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7–8), 2015.

A Extra Preliminaries

Authenticated encryption scheme (with associated data). An authenticated encryption with associated data (AEAD or AE) scheme $\mathsf{AEAD} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is a triple of algorithms. Key generation KGen takes no input and outputs a single symmetric key k. The randomized symmetric encryption algorithm Enc takes as input a symmetric key k, a message m, and optionally some associated data AD, and outputs a ciphertext C. (When there is no associated data, it is standard to omit that argument to Enc and Dec.) The deterministic decryption algorithm Dec takes as input a key k, ciphertext C, and optionally some associated data AD, and outputs a message m or an error symbol $\perp$. Symmetric encryption (SE) schemes are very similar to AEADs, except they do not accept associated data in Enc or Dec.

Message authentication code. A message authentication code (MAC) $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ is a triple of algorithms. Key generation KGen works as AEAD's key generation above. The deterministic tag algorithm Tag takes as input a key k and a message m, and outputs a tag T. The deterministic verification algorithm Vf takes as input a key k, message m, and tag T, and outputs a bit. (Deterministic MACs have a canonical verification algorithm, which simply re-runs Tag and returns the result of comparing this internally re-computed tag to T.)

Correctness properties. We will briefly re-define three correctness notions from [24]: a correctness property of a KEM and two correctness properties of PKE. We say that KEM is δ-correct if

$$\Pr[\mathsf{KEM.Decap}(\mathsf{sk}, C) \neq \mathsf{k} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KEM.KGen}\,;\ (\mathsf{k}, C) \leftarrow_{\$} \mathsf{KEM.Encap}(\mathsf{pk})\,] \leq \delta\ .$$

We say that a public-key encryption scheme PKE is δ-correct if

$$\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{PKE.Dec}(\mathsf{sk}, C) \neq m \mid C \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, m)\,]\right] \leq \delta$$

where the expectation is taken over the output distribution of $\mathsf{PKE.KGen}$.

γ-spreadness. We now define γ-spreadness of a PKE: we say that PKE is γ-spread if for every key pair $(\mathsf{pk}, \mathsf{sk})$, message $m \in \mathcal{M}$, and ciphertext $C \in \mathcal{C}$,

$$\Pr_{r \leftarrow_{\$} \mathcal{R}}[C = \mathsf{PKE.Enc}(\mathsf{pk}, m; r)] \leq 2^{-\gamma}$$

where $\mathcal{R}$ is the set of all possible random strings that can be sampled in $\mathsf{PKE.Enc}$.

B QROM Lemmas

The following lemma allows a *perfect* simulation of a quantum random oracle against an adversary.

Lemma 2 (Simulating a QRO [42]). *Let $H(\cdot)$ be an oracle drawn from the set of 2q-wise independent functions uniformly at random. Then the advantage any quantum algorithm making at most q quantum queries to $H(\cdot)$ has in distinguishing $H(\cdot)$ from a truly random oracle is identically zero.*

The second lemma intuitively states that a quantum random oracle can be used as a *quantum-accessible* pseudo-random function, even if the distinguisher is given full access to the quantum random oracle in addition to the PRF oracle.

Lemma 3 (PRF based on a QRO). *Let Ω_H be the set of all functions $H : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ and Ω_R be the set of all functions $R : \mathcal{X} \to \mathcal{Y}$. Let $H \leftarrow_\$ \Omega_H$, $k \leftarrow_\$ \mathcal{K}$ and $R \leftarrow_\$ \Omega_R$. Define the oracles $F_0 = H(k, \cdot)$ and $F_1 = R(\cdot)$. Consider an oracle algorithm/distinguisher A^{H,F_i} that makes at most q queries to H and F_i ($i \in \{0, 1\}$). If ("the PRF key") k is chosen independently from A^{H,F_i}'s view, then we have*

$$|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]| \leq \frac{2q}{\sqrt{|\mathcal{K}|}}$$

The third lemma provides a generic reduction from a hiding-style property (indistinguishability) to a one-wayness-style property (unpredictability) in the QROM.

Lemma 4 (One-Way to Hiding (OW2H) [39]). *Let Ω_H be the set of all functions $H : \mathcal{X} \to \mathcal{Y}$ and let $H \leftarrow_\$ \Omega_H$ be a quantum random oracle. Consider an oracle algorithm A^H that makes at most q queries to H. Let B^H be an oracle algorithm that on input x does the following: picks $i \leftarrow_\$ \{1, \ldots, q\}$ and $y \leftarrow_\$ \mathcal{Y}$, runs $A^H(x, y)$ until (just before) the i-th query, measures the argument of the query in the computational basis and outputs the measurement outcome (if A makes less than i queries, B outputs $\perp \notin \mathcal{X}$). Let,*

$$\begin{aligned}
P_A^1 &= \Pr[b' = 1 : H \leftarrow_\$ \Omega_H,\ x \leftarrow_\$ \mathcal{X},\ b' \leftarrow A^H(x, H(x))] \\
P_A^2 &= \Pr[b' = 1 : H \leftarrow_\$ \Omega_H,\ x \leftarrow_\$ \mathcal{X},\ y \leftarrow_\$ \mathcal{Y},\ b' \leftarrow A^H(x, y)] \\
P_B &= \Pr[x' = x : H \leftarrow_\$ \Omega_H,\ x \leftarrow_\$ \mathcal{X},\ x' \leftarrow B^H(x, i)]
\end{aligned}$$

Then, we have $|P_A^1 - P_A^2| \leq 2q\sqrt{P_B}$.

The following lemma gives a lower bound for a decisional variant of so-called *generic quantum search problem*.

Lemma 5 (Generic Search Problem [5, 26]). *Let $\gamma \in [0, 1]$ and Z be a finite set. Define $N_1 : Z \to \{0, 1\}$ to be the following function: for each $z \in Z$, $N_1(z) = 1$ with probability p_z ($p_z \leq \gamma$), and $N_1(z) = 0$ else. Let N_2 be the function with $N_2(z) = 0$ $\forall z \in Z$. If an oracle algorithm A (possibly unbounded) makes at most q quantum queries to N_1 (or N_2), then*

$$|\Pr[b = 1 : b \leftarrow A^{N_1}] - \Pr[b = 1 : b \leftarrow A^{N_2}]| \leq 2q\sqrt{\gamma}.$$

The following lemma describes the collision-resistance of quantum random oracles.

Lemma 6 ([43, Theorem 3.1]). *There is a universal constant α (< 648) such that the following holds: Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $H : \mathcal{X} \to \mathcal{Y}$ be a random oracle. If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H at most q times, then we have*

$$\Pr[H(x_0) = H(x_1) \wedge x_0 \neq x_1 : (x_0, x_1) \leftarrow \mathcal{A}^H] \leq \frac{\alpha(q+1)^3}{|\mathcal{Y}|},$$

where all oracle accesses of $\mathcal{A}$ can be quantum.

C Omitted Proofs

C.1 Proof of Theorem 1

Theorem 1. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen}, \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *with a one-time secure authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$*. If* KEM *is* δ*-correct, then:*

1. *For any* ANO-CCA *adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exist* wANO-CCA *adversary* $\mathcal{B}$*,* IND-CCA *adversary* $\mathcal{C}$ *and* WROB-CPA *adversary* $\mathcal{D}$ *against* KEM*, and* INT-CTXT *adversary* $\mathcal{E}$ *against* DEM *such that*

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq & \mathbf{Adv}^{\text{wANO-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}}(\mathcal{C}) \\ & + \mathbf{Adv}^{\text{WROB-CPA}}_{\mathsf{KEM}}(\mathcal{D}) + \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \delta\,.\end{aligned}$$

   The running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{E}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{D}$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

2. *For any* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exists* WROB-ATK *(resp.* SROB-ATK*) adversary* $\mathcal{B}$ *against* KEM *such that*

$$\begin{aligned}\mathbf{Adv}^{\text{WROB-ATK}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) &\leq \mathbf{Adv}^{\text{WROB-ATK}}_{\mathsf{KEM}}(\mathcal{B})\,, \\ \mathbf{Adv}^{\text{SROB-ATK}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) &\leq \mathbf{Adv}^{\text{SROB-ATK}}_{\mathsf{KEM}}(\mathcal{B})\,,\end{aligned}$$

   where $\text{ATK} \in \{\text{CPA}, \text{CCA}\}$ *and the running time of* $\mathcal{B}$ *is that of* $\mathcal{A}$*.*

Proof (of Theorem 1.1*).*

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for PKE^{hy}. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_4$ described in Figure 8.

Game G_0**:** The game G_0 is exactly the ANO-CCA game for PKE^{hy}. Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A})$$

Game G_1**:** In game G_1, we first make some "cosmetic" changes. Namely, the pair (c_1^*, k^*) is generated by running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b *before* the adversary $\mathcal{A}$ gets to choose a message m. This change does not affect $\mathcal{A}$'s view in any way.

Next, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_b, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$ (and $c_2 \neq c_2^*$), then the oracle uses k^* to decrypt c_2, instead of first decapsulating c_1^* to recover a session key k'. It is not hard to see that the games G_0 and G_1 are equivalent unless there is a decapsulation error w.r.t. KEM. Therefore, we have

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \delta$$

Game G_2**:** In game G_2, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle returns $\perp$. Again it is not hard to see that the games G_1 and G_2 are equivalent unless the following event occurs: $\mathsf{Decap}(\mathsf{sk}_{1-b}, c_1^*) = k' \neq \perp$ (and $\mathsf{Dec}(k', c_2) \neq \perp$) where $\mathsf{Encap}(\mathsf{pk}_b) = (c_1^*, k^*)$. And we can bound the probability of this event occurring by the advantage of an adversary $\mathcal{D}$ in the WROB-CPA game of KEM. The adversary $\mathcal{D}$, upon receiving public-keys pk_0 and pk_1, simply samples a bit b uniformly at random, i.e., $b \leftarrow\!\!\$\, \{0, 1\}$, and returns the bit to the WROB-CPA challenger. Hence,

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathbf{Adv}^{\text{WROB-CPA}}_{\mathsf{KEM}}(\mathcal{D})$$

Game G_3**:** In game G_3, we compute c_2^* in the setup as "$c_2^* = \mathsf{Enc}(\hat{k}, m)$", instead of "$c_2^* = \mathsf{Enc}(k^*, m)$" as in G_2, for a uniformly random key $\hat{k}$ (i.e., $\hat{k} \leftarrow\!\!\$\, \mathcal{K}$, where $\mathcal{K}$ is the encapsulated key-space of KEM) that is independent of k^*. We make the appropriate modification in the $\mathsf{Dec}^{hy}(\mathsf{sk}_b, \cdot)$ oracle as well, i.e., if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle uses $\hat{k}$ (instead of k^*) to decrypt c_2.

We now show that the difference in $\mathcal{A}$'s success probabilities in games G_2 and G_3 can be bounded by the advantage of an adversary $\mathcal{C}$ in the IND-CCA game of KEM. Upon receiving the input (pk, c^*, k) from

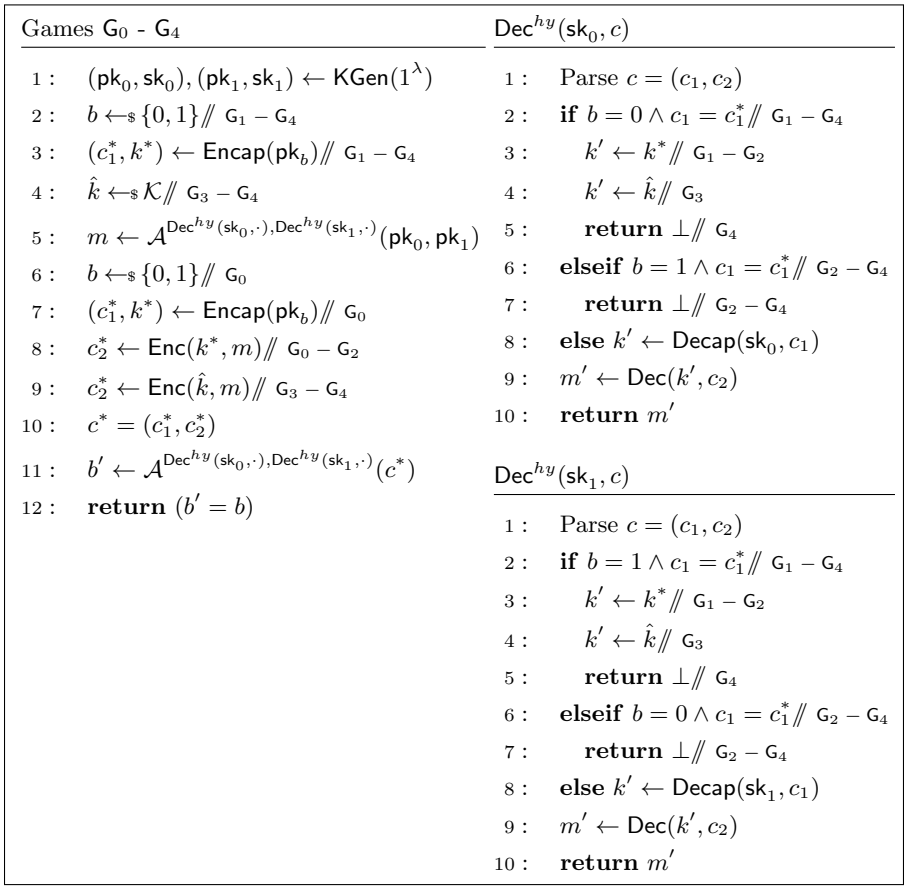


Fig. 8. Games G_0 – G_4 for the proof of Theorem 1.

its IND-CCA challenger, where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk})$ and $k \leftarrow_\$ \{k^*, \hat{k}\}$ for a uniformly random key $\hat{k}$ that is independent of k^*, $\mathcal{C}$ proceeds as described in Figure 9. Note that if k is a "real" (respectively, "random") key, i.e., $k = k^*$ (resp., $k = \hat{k}$), then $\mathcal{C}$ perfectly simulates game G_2 (resp., G_3) towards $\mathcal{A}$ (also note that, to answer $\mathcal{A}$'s decryption queries, $\mathcal{C}$ never has to make the *forbidden* query $c^*(= c_1^*)$ to its decapsulation oracle $\mathsf{Decap}(\mathsf{sk}, \cdot)(= \mathsf{Decap}(\mathsf{sk}_b, \cdot))$. Therefore, we have

$$\begin{aligned}|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| = |\Pr[1 \leftarrow \mathcal{C}^{\mathsf{Decap}(\mathsf{sk},\cdot)}(\mathsf{pk}, c^*, k) \mid k = \hat{k}] \\ - \Pr[1 \leftarrow \mathcal{C}^{\mathsf{Decap}(\mathsf{sk},\cdot)}(\mathsf{pk}, c^*, k) \mid k = k^*]| \leq 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}}(\mathcal{C})\end{aligned}$$

$\mathcal{C}^{\mathsf{Decap}(\mathsf{sk},\cdot)}(\mathsf{pk}, c^*, k)$

1 : $b \leftarrow_\$ \{0,1\}$
2 : $\mathsf{pk}_b = \mathsf{pk}$
3 : $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}(1^\lambda)$
4 : $c_1^* = c^*$
5 : $m \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
6 : $c_2^* \leftarrow \mathsf{Enc}(k, m)$
7 : $c^* = (c_1^*, c_2^*)$
8 : $b' \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(c^*)$
9 : **return** $(b' = b)$

$\mathsf{Dec}^{hy}(\mathsf{sk}_0, c)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $b = 0 \wedge c_1 = c_1^*$
3 : $\quad k' \leftarrow k$
4 : **elseif** $b = 1 \wedge c_1 = c_1^*$
5 : $\quad$ **return** $\perp$
6 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0, c_1)$
7 : $m' \leftarrow \mathsf{Dec}(k', c_2)$
8 : **return** m'

$\mathsf{Dec}^{hy}(\mathsf{sk}_1, c)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $b = 1 \wedge c_1 = c_1^*$
3 : $\quad k' \leftarrow k$
4 : **elseif** $b = 0 \wedge c_1 = c_1^*$
5 : $\quad$ **return** $\perp$
6 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1, c_1)$
7 : $m' \leftarrow \mathsf{Dec}(k', c_2)$
8 : **return** m'

Fig. 9. IND-CCA adversary $\mathcal{C}^{\mathsf{Decap}(\mathsf{sk},\cdot)}$ for the proof of Theorem 1.

Game G_4: In game G_4, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_b, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle returns $\perp$. It is not hard to see that the games G_3 and G_4 are equivalent unless the following event occurs: $\mathcal{A}$ makes a decryption query (c_1^*, c_2) to the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_b, \cdot)$ such that $\mathsf{Dec}(\hat{k}, c_2) \neq \perp$, for a uniformly random key $\hat{k}$. And we can bound the probability of this event occurring by the advantage of an adversary $\mathcal{E}$ in the INT-CTXT game of DEM. In the INT-CTXT game, we are implicitly defining $\hat{k}$ to be the random secret key chosen by the challenger. The adversary $\mathcal{E}$ proceeds as described in Figure 20. Note that if the aforementioned event occurs, then $\mathcal{E}$ wins its corresponding game (also note that, $\mathcal{E}$ only makes a single encryption query to the one-time AE-secure DEM, namely "$c_2^* = \mathsf{Enc}(\hat{k}, m)$", and it never makes the forbidden query c_2^* to its decryption oracle $\mathsf{Dec}(\hat{k}, \cdot)$). Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E})$$

Finally, we show that $\mathcal{A}$'s success probability in game G_4 can be bounded by the advantage of an adversary $\mathcal{B}$ in the wANO-CCA game of KEM. Upon receiving public-keys pk_0 and pk_1 along with the ciphertext c^*, where $(c^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b chosen by the challenger, the adversary $\mathcal{B}$ proceeds as described in Figure 11. Observe that $\mathcal{B}$ perfectly simulates the game G_4 towards $\mathcal{A}$ (also note that, to

$\mathcal{E}^{\mathsf{Enc}(\hat{k},\cdot),\mathsf{Dec}(\hat{k},\cdot)}(1^\lambda)$

1 : $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}(1^\lambda)$
2 : $b \leftarrow\!\!\$\ \{0,1\}$
3 : $(c_1^*, k^*) \leftarrow \mathsf{Encap}(\mathsf{pk}_b)$
4 : $m \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
5 : $c_2^* \leftarrow \mathsf{Enc}(\hat{k}, m)$
6 : $c^* = (c_1^*, c_2^*)$
7 : $b' \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(c^*)$
8 : **return** $\perp$

$\mathsf{Dec}^{hy}(\mathsf{sk}_0, c)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $b = 0 \wedge c_1 = c_1^*$
3 : query $\mathsf{Dec}(\hat{k}, c_2)$
4 : **return** $\perp$
5 : **elseif** $b = 1 \wedge c_1 = c_1^*$
6 : **return** $\perp$
7 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0, c_1)$
8 : $m' \leftarrow \mathsf{Dec}(k', c_2)$
9 : **return** m'

$\mathsf{Dec}^{hy}(\mathsf{sk}_1, c)$

1 : Parse $c = (c_1, c_2)$
2 : **if** $b = 1 \wedge c_1 = c_1^*$
3 : query $\mathsf{Dec}(\hat{k}, c_2)$
4 : **return** $\perp$
5 : **elseif** $b = 0 \wedge c_1 = c_1^*$
6 : **return** $\perp$
7 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1, c_1)$
8 : $m' \leftarrow \mathsf{Dec}(k', c_2)$
9 : **return** m'

Fig. 10. INT-CTXT adversary $\mathcal{E}^{\mathsf{Enc}(\hat{k},\cdot),\mathsf{Dec}(\hat{k},\cdot)}$ for the proof of Theorem 1.

answer $\mathcal{A}$'s decryption queries, $\mathcal{B}$ never has to make the *forbidden* query $c^*(=c_1^*)$ to its decapsulation oracles $\mathsf{Decap}(\mathsf{sk}_0,\cdot)$ and $\mathsf{Decap}(\mathsf{sk}_1,\cdot)$). Therefore, we have $|\Pr[\mathsf{G}_4=1]-1/2| = \mathbf{Adv}^{\text{wANO-CCA}}_{\mathsf{KEM}}(\mathcal{B})$.

Collecting all of the above bounds, we finally arrive at

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{wANO-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}}(\mathcal{C}) + \mathbf{Adv}^{\text{WROB-CPA}}_{\mathsf{KEM}}(\mathcal{D}) + \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \delta$$

$\mathcal{B}^{\mathsf{Decap}(\mathsf{sk}_0,\cdot),\mathsf{Decap}(\mathsf{sk}_1,\cdot)}(\mathsf{pk}_0,\mathsf{pk}_1,c^*)$

1 : $\hat{k} \leftarrow_{\$} \mathcal{K}$
2 : $c_1^* = c^*$
3 : $m \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(\mathsf{pk}_0,\mathsf{pk}_1)$
4 : $c_2^* \leftarrow \mathsf{Enc}(\hat{k},m)$
5 : $c^* = (c_1^*, c_2^*)$
6 : $b' \leftarrow \mathcal{A}^{\mathsf{Dec}^{hy}(\mathsf{sk}_0,\cdot),\mathsf{Dec}^{hy}(\mathsf{sk}_1,\cdot)}(c^*)$
7 : **return** b'

$\mathsf{Dec}^{hy}(\mathsf{sk}_0,c)$

1 : Parse $c=(c_1,c_2)$
2 : **if** $c_1 = c_1^*$ **return** $\perp$
3 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_0,c_1)$
4 : $m' \leftarrow \mathsf{Dec}(k',c_2)$
5 : **return** m'

$\mathsf{Dec}^{hy}(\mathsf{sk}_1,c)$

1 : Parse $c=(c_1,c_2)$
2 : **if** $c_1 = c_1^*$ **return** $\perp$
3 : **else** $k' \leftarrow \mathsf{Decap}(\mathsf{sk}_1,c_1)$
4 : $m' \leftarrow \mathsf{Dec}(k',c_2)$
5 : **return** m'

Fig. 11. wANO-CCA adversary $\mathcal{B}^{\mathsf{Decap}(\mathsf{sk}_0,\cdot),\mathsf{Decap}(\mathsf{sk}_1,\cdot)}$ for the proof of Theorem 1.

Proof (of Theorem 1.2*).*

Let $\mathcal{A}$ be an adversary in the WROB-ATK game for PKE^{hy}. Upon receiving two (honestly-generated) public-keys pk_0 and pk_1, $\mathcal{A}$ wins the game if it returns a message and a bit, namely (m,b), such that $\mathsf{Dec}^{hy}(sk_{1-b},C) \neq \perp$ where $C(=(C_{\mathsf{KEM}},C_{\mathsf{DEM}})) \leftarrow_{\$} \mathsf{Enc}^{hy}(pk_b,m)$. Let $(C_{\mathsf{KEM}},k_b) \leftarrow_{\$} \mathsf{Encap}(\mathsf{pk}_b)$ and $\mathsf{Decap}(\mathsf{sk}_{1-b},C_{\mathsf{KEM}}) = k_{1-b}$. It is easy to see that $k_{1-b} \neq \perp$, since $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b},C) \neq \perp$ implies $\mathsf{Dec}(k_{1-b},C_{\mathsf{DEM}}) \neq \perp$. The probability of $\mathcal{A}$ winning the game can then be bounded by the advantage of an adversary $\mathcal{B}$ in the WROB-ATK game for KEM. Upon receiving two public-keys pk_0 and pk_1 from its WROB-ATK challenger, $\mathcal{B}$ forwards the keys to $\mathcal{A}$ and simulates the WROB-ATK game w.r.t. PKE^{hy} (note that if ATK = CCA, then $\mathcal{B}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i,\cdot)$ oracles since it has access to the $\mathsf{Decap}(\mathsf{sk}_i,\cdot)$ oracles in its WROB-CCA game). Once $\mathcal{A}$ finally submits the pair (m,b), $\mathcal{B}$ forwards the bit b to the WROB-ATK challenger. Note that a win for $\mathcal{A}$ implies a win for $\mathcal{B}$.

Similarly, let $\mathcal{A}$ be an adversary in the SROB-ATK game for PKE^{hy}. Upon receiving two (honestly-generated) public-keys pk_0 and pk_1, $\mathcal{A}$ wins the game if it returns a ciphertext C $(=(C_{\mathsf{KEM}},C_{\mathsf{DEM}}))$ such that $\mathsf{Dec}^{hy}(sk_0,C) \neq \perp$ and $\mathsf{Dec}^{hy}(sk_1,C) \neq \perp$. Let $\mathsf{Decap}(\mathsf{sk}_0,C_{\mathsf{KEM}}) = k_0$ and $\mathsf{Decap}(\mathsf{sk}_1,C_{\mathsf{KEM}}) = k_1$. It is again easy to see that $k_0 \neq \perp$ and $k_1 \neq \perp$ since we have $\mathsf{Dec}(k_0,C_{\mathsf{DEM}}) \neq \perp$ and $\mathsf{Dec}(k_1,C_{\mathsf{DEM}}) \neq \perp$. Hence we can bound the winning probability of $\mathcal{A}$ by the advantage of an adversary $\mathcal{B}$ in the SROB-ATK game for KEM. Upon receiving two public-keys pk_0 and pk_1 from its SROB-ATK challenger, $\mathcal{B}$ forwards the keys to $\mathcal{A}$ and simulates the SROB-ATK game w.r.t. PKE^{hy} (note that if ATK = CCA, then $\mathcal{B}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i,\cdot)$ oracles since it has access to the $\mathsf{Decap}(\mathsf{sk}_i,\cdot)$ oracles in its SROB-CCA game). Once $\mathcal{A}$ submits the final ciphertext $C=(C_{\mathsf{KEM}},C_{\mathsf{DEM}})$, $\mathcal{B}$ forwards C_{KEM} to the SROB-ATK challenger. Again, a win for $\mathcal{A}$ implies a win for $\mathcal{B}$.

C.2 Proof of Theorem 2

Theorem 2. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen},\mathsf{Enc}^{hy},\mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing a KEM* $\mathsf{KEM} = (\mathsf{KGen},\mathsf{Encap},\mathsf{Decap})$ *with a DEM* $\mathsf{DEM} = (\mathsf{Enc},\mathsf{Dec})$. *Then for any* SROB-CCA *(resp.*

WROB-CCA*) adversary* $\mathcal{A}$ *against* PKE^{hy}*, there exist* SCFR-CCA *(resp.* WCFR-CCA*) adversary* $\mathcal{B}$ *againt* KEM *and* FROB *(resp.* XROB*) adversary* $\mathcal{C}$ *against* DEM *such that*

$$\mathbf{Adv}^{\text{SROB-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + \mathbf{Adv}^{\text{FROB}}_{\mathsf{DEM}}(\mathcal{C}),$$
$$\mathbf{Adv}^{\text{WROB-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{WCFR-CCA}}_{\mathsf{KEM}}(\mathcal{B}) + \mathbf{Adv}^{\text{XROB}}_{\mathsf{DEM}}(\mathcal{C}),$$

where the running times of $\mathcal{B}$ *and* $\mathcal{C}$ *are the same as that of* $\mathcal{A}$*.*

Proof. Let $\mathcal{A}$ be an adversary in the SROB-CCA game for PKE^{hy}. Upon receiving two (honestly-generated) public-keys pk_0 and pk_1, $\mathcal{A}$ wins the game if it returns a ciphertext C $(= (C_{\mathsf{KEM}}, C_{\mathsf{DEM}}))$ such that $\mathsf{Dec}^{hy}(sk_0, C) \neq \perp$ and $\mathsf{Dec}^{hy}(sk_1, C) \neq \perp$. Let $\mathsf{Decap}(\mathsf{sk}_0, C_{\mathsf{KEM}}) = k_0$ and $\mathsf{Decap}(\mathsf{sk}_1, C_{\mathsf{KEM}}) = k_1$. It is easy to see that $k_0 \neq \perp$ and $k_1 \neq \perp$. Now we consider two (disjoint) sub-events w.r.t. this winning event:

- $\underline{k_0 = k_1}$. It is easy to see that the probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{B}$ in the SCFR-CCA game for KEM. Upon receiving two public-keys pk_0 and pk_1 from its SCFR-CCA challenger, $\mathcal{B}$ forwards the keys to $\mathcal{A}$ and simulates the SROB-CCA game w.r.t. PKE^{hy} (note that $\mathcal{B}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i, \cdot)$ oracles since it has access to the $\mathsf{Decap}(\mathsf{sk}_i, \cdot)$ oracles in its SCFR-CCA game). Once $\mathcal{A}$ submits the final ciphertext $C = (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})$, $\mathcal{B}$ forwards C_{KEM} to the SCFR-CCA challenger. Note that $k_0 = k_1$ implies a win for $\mathcal{B}$.
- $\underline{k_0 \neq k_1}$. The probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{C}$ in the FROB game for DEM. $\mathcal{C}$ generates two key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$ honestly using KGen and forwards $(\mathsf{pk}_0, \mathsf{pk}_1)$ to $\mathcal{A}$. $\mathcal{C}$ then simulates the SROB-CCA game w.r.t. PKE^{hy} towards $\mathcal{A}$ (again note that $\mathcal{C}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i, \cdot)$ oracles since it has access to the corresponding secret keys sk_0, sk_1). Once $\mathcal{A}$ submits the final ciphertext $C = (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})$, $\mathcal{C}$ first computes k_0, k_1 as above and forwards $(C_{\mathsf{DEM}}, k_0, k_1)$ to the FROB challenger. Note that $\mathcal{A}$ winning implies $\mathsf{Dec}^{hy}(\mathsf{sk}_i, C) \neq \perp$ which in turn implies $\mathsf{Dec}(k_i, C_{\mathsf{DEM}}) \neq \perp$. Therefore, the (sub-)event that $k_0 \neq k_1$ implies a win for $\mathcal{C}$.

Similarly, let $\mathcal{A}$ be an adversary in the WROB-CCA game for PKE^{hy}. Upon receiving two (honestly-generated) public-keys pk_0 and pk_1, $\mathcal{A}$ wins the game if it returns a message and a bit, namely (m, b), such that $\mathsf{Dec}^{hy}(sk_{1-b}, C) \neq \perp$ where $C(= (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})) \leftarrow\!\!\$\, \mathsf{Enc}^{hy}(pk_b, m)$. Let $(C_{\mathsf{KEM}}, k_b) \leftarrow\!\!\$\, \mathsf{Encap}(\mathsf{pk}_b)$ and $\mathsf{Decap}(\mathsf{sk}_{1-b}, C_{\mathsf{KEM}}) = k_{1-b}$. It is easy to see that $k_{1-b} \neq \perp$, since $\mathsf{Dec}^{hy}(sk_{1-b}, C) \neq \perp$. Now we consider two (disjoint) sub-events w.r.t. this winning event:

- $\underline{k_b = k_{1-b}}$. It is easy to see that the probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{B}$ in the WCFR-CCA game for KEM. Upon receiving two public-keys pk_0 and pk_1 from its WCFR-CCA challenger, $\mathcal{B}$ forwards the keys to $\mathcal{A}$ and simulates the WROB-CCA game w.r.t. PKE^{hy} (note that $\mathcal{B}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i, \cdot)$ oracles since it has access to the $\mathsf{Decap}(\mathsf{sk}_i, \cdot)$ oracles in its WCFR-CCA game). Once $\mathcal{A}$ finally submits the pair (m, b), $\mathcal{B}$ forwards the bit b to the WCFR-CCA challenger. Note that $k_b = k_{1-b}$ implies a win for $\mathcal{B}$.
- $\underline{k_b \neq k_{1-b}}$. The probability of this winning sub-event can be bounded by the advantage of an adversary $\mathcal{C}$ in the XROB game for DEM. $\mathcal{C}$ generates two key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$ honestly using KGen and forwards $(\mathsf{pk}_0, \mathsf{pk}_1)$ to $\mathcal{A}$. $\mathcal{C}$ then simulates the WROB-CCA game w.r.t. PKE^{hy} towards $\mathcal{A}$ (again note that $\mathcal{C}$ can simulate the $\mathsf{Dec}^{hy}(\mathsf{sk}_i, \cdot)$ oracles since it has access to the corresponding secret keys sk_0, sk_1). Once $\mathcal{A}$ submits the pair (m, b), $\mathcal{C}$ first computes k_b and k_{1-b} as $(C_{\mathsf{KEM}}, k_b) \leftarrow\!\!\$\, \mathsf{Encap}(\mathsf{pk}_b)$ and $k_{1-b} \leftarrow \mathsf{Decap}(\mathsf{sk}_{1-b}, C_{\mathsf{KEM}})$ respectively. Then it samples uniform random coins r to be used in the DEM encryption of m and forwards $(m, k_b, r, C_{\mathsf{DEM}}, k_{1-b})$ to the XROB challenger. It is not hard to see that $\mathcal{A}$ winning its WROB-CCA game coupled with the sub-event $k_0 \neq k_1$ implies a win for $\mathcal{C}$.

C.3 Proof of Theorem 3

Theorem 3. *Suppose there exists a KEM that is simultaneously* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure. Suppose that there exists a* SUF-CMA*-secure MAC scheme and an* ROR-CPA *secure symmetric encryption scheme (such schemes can be built assuming only the existence of one-way functions). Suppose also that collision-resistant hash functions exist. Then there exists an implicit-rejection KEM that is* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure and a DEM that is* ROR-CCA *secure, such that the hybrid PKE scheme obtained from their composition is not* SROB-CCA *secure.*

Proof. We focus on the "Encrypt-then-MAC" (EtM) construction of a DEM. Namely, let $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vf})$ be an SUF-CMA secure message authentication code. We construct $\overline{\mathsf{MAC}} = (\overline{\mathsf{Tag}}, \overline{\mathsf{Vf}})$ where the only difference from MAC is that a fixed special key $\overline{k}$ is chosen uniformly at random from the original MAC key-space such that the verification of *any* tag under $\overline{k}$ verifies successfully, i.e., $\overline{\mathsf{Vf}}(\overline{k}, \cdot) = 1$. Note that $\overline{\mathsf{MAC}}$ is also SUF-CMA secure because the probability of sampling $\overline{k}$ uniformly at random from the key-space can be considered to be negligible. So by composing $\overline{\mathsf{MAC}}$ with an ROR-CPA secure symmetric encryption scheme that *never* rejects invalid ciphertexts via the EtM construction, we get an AE-secure $\overline{\mathsf{DEM}}$.

Now let $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM that is SCFR-CCA, IND-CCA and ANO-CCA secure. Also let H be a collision-resistant hash function with its range being the key-space of the ROR-CPA symmetric encryption scheme used to obtain $\overline{\mathsf{DEM}}$. We construct $\overline{\mathsf{KEM}} = (\mathsf{KGen}, \mathsf{Encap}, \overline{\mathsf{Decap}})$ where the only difference from KEM is that the ciphertext space is augmented by a special bitstring $\overline{c}$. With respect to $\overline{c}$, the decapsulation operation works as follows: $\overline{\mathsf{Decap}}(\mathsf{sk}, \overline{c}) = H(\mathsf{pk})||\overline{k}$, for any key-pair $(\mathsf{pk}, \mathsf{sk})$ generated by KGen and the fixed $\overline{\mathsf{MAC}}$ key $\overline{k}$ described above. It is not hard to see that $\overline{\mathsf{KEM}}$ is also IND-CCA and ANO-CCA secure. To argue for the SCFR-CCA security of $\overline{\mathsf{KEM}}$, the only additional case to consider is when the adversary returns the final ciphertext $\overline{c}$. Note that $\overline{\mathsf{Decap}}(\mathsf{sk}_0, \overline{c}) = \overline{\mathsf{Decap}}(\mathsf{sk}_1, \overline{c})$, or equivalently, $H(\mathsf{pk}_0)||\overline{k} = H(\mathsf{pk}_1)||\overline{k}$, happens with a negligible probability because of the collision-resistance of H.

Now the resulting hybrid PKE scheme obtained by composing $\overline{\mathsf{KEM}}$ and $\overline{\mathsf{DEM}}$ is not SROB-CCA secure. This is because an SROB-CCA adversary, upon receiving two public-keys $\mathsf{pk}_0, \mathsf{pk}_1$, could simply output the ciphertext $(\overline{c}, c'||\sigma')$ where $c'||\sigma'$ is an arbitrary $\overline{\mathsf{DEM}}$ ciphertext. The adversary wins the SROB-CCA game because when decrypting $(\overline{c}, c'||\sigma')$ under sk_i $(i \in \{0, 1\})$ we have $\overline{\mathsf{Decap}}(\mathsf{sk}_i, \overline{c}) = H(\mathsf{pk}_i)||\overline{k}$. Since the use of key $\overline{k}$ always leads to successful verification of the $\overline{\mathsf{DEM}}$ ciphertext and the underlying ROR-CPA symmetric encryption never rejects, we thus have that the final decryption of $(\overline{c}, c'||\sigma')$ does *not* return $\perp$ under either of the secret keys sk_0, sk_1.

C.4 Proof of Theorem 4

Theorem 4. *Suppose there exists a KEM that is simultaneously* SROB-CCA*,* IND-CCA *and* ANO-CCA *secure, a claw-free pair of permutations with domain and range being the encapsulated key-space of the KEM, and a collision-resistant hash function. Suppose also that there exists a DEM that is* ROR-CCA *and* XROB*-secure. Then there exists an implicit-rejection KEM that is* SCFR-CCA*,* IND-CCA *and* ANO-CCA *secure and a DEM that is* ROR-CCA *and* XROB*-secure, such that the resulting hybrid PKE is not* ANO-CCA *secure.*

Proof. Let $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ be a key encapsulation mechanism that is IND-CCA, ANO-CCA and SROB-CCA secure. Let $(\mathcal{F}_1, \mathcal{F}_2)$ be a claw-free pair of permutations, with the domain and range being the encapsulated key-space of KEM, and let H be a collision-resistant hash function that maps the space of public-keys of KEM to the encapsulated key-space. We now construct $\overline{\mathsf{KEM}} = (\mathsf{KGen}, \overline{\mathsf{Encap}}, \overline{\mathsf{Decap}})$ that is IND-CCA, ANO-CCA and SCFR-CCA secure, but when composed with an XROB-secure DEM, does not result in an ANO-CCA secure hybrid PKE scheme.

We first generate public parameters for $\overline{\mathsf{KEM}}$ which are related to the instantiation of $(\mathcal{F}_1, \mathcal{F}_2)$. Recall that $\mathcal{F}_i = (G_i, f_i, f_i^{-1})$ where $G = G_1 = G_2$ is the generator for the pair of claw-free permutations. Hence, we generate the public parameters $f_1(\mathsf{PK}, .)$ and $f_2(\mathsf{PK}, .)$, where PK is the public-key of the pair of claw-free permutations. The subsequent key generation algorithm of $\overline{\mathsf{KEM}}$ (which is independent of the generation of public parameters) is the same as that of KEM. The $\overline{\mathsf{Encap}}$ and $\overline{\mathsf{Decap}}$ algorithms of $\overline{\mathsf{KEM}}$ are described in the Figure 12.

It is not hard to see that $\overline{\mathsf{KEM}}$ is also ANO-CCA secure. To argue about the IND-CCA security of $\overline{\mathsf{KEM}}$ based on the IND-CCA security of KEM, we need to observe in the reduction that when the IND-CCA challenger of KEM returns a uniformly random key k (in the real-or-random experiment), $f_1(\mathsf{PK}, k)$ is a uniformly random key as well, since $f_1(\mathsf{PK}, .)$ is a permutation. To show the SCFR-CCA security of $\overline{\mathsf{KEM}}$, consider an SCFR-CCA adversary that, after receiving two $\overline{\mathsf{KEM}}$ public-keys $\mathsf{pk}_0, \mathsf{pk}_1$, wins the corresponding security game by returning the ciphertext C such that $\overline{\mathsf{Decap}}(\mathsf{sk}_0, C) = \overline{\mathsf{Decap}}(\mathsf{sk}_1, C)$. There are 3 cases to consider:

- **Case 1:** If $\mathsf{Decap}(\mathsf{sk}_0, C) \neq \perp$ and $\mathsf{Decap}(\mathsf{sk}_1, C) \neq \perp$, then we can break the SROB-CCA security of KEM via a straightforward reduction.

$\overline{\mathsf{Encap}}(\mathsf{pk})$	$\overline{\mathsf{Decap}}(\mathsf{sk}, C)$
$(C, k) \leftarrow\!\!\$\ \mathsf{Encap}(\mathsf{pk})$	$k' \leftarrow \mathsf{Decap}(\mathsf{sk}, C)$
$\overline{k} \leftarrow f_1(\mathsf{PK}, k)$	**if** $k' = \perp$ **then**
return $(C, \overline{k})$	$\quad \overline{k}' \leftarrow f_2(\mathsf{PK}, H(\mathsf{pk}))$
	else $\overline{k}' \leftarrow f_1(\mathsf{PK}, k')$
	return $\overline{k}'$

Fig. 12. $\overline{\mathsf{Encap}}$ and $\overline{\mathsf{Decap}}$ algorithms of $\overline{\mathsf{KEM}}$ for the proof of Theorem 4.

- **Case 2:** If $\mathsf{Decap}(\mathsf{sk}_0, C) = \perp$ and $\mathsf{Decap}(\mathsf{sk}_1, C) = \perp$, then this would mean that $f_2(\mathsf{PK}, H(\mathsf{pk}_0)) = f_2(\mathsf{PK}, H(\mathsf{pk}_1))$. This would break the collision-resistance of H as $f_2(\mathsf{PK}, .)$ is a permutation, and with high probability, $\mathsf{pk}_0 \neq \mathsf{pk}_1$.
- **Case 3:** Without loss of generality, let $\mathsf{Decap}(\mathsf{sk}_0, C) = k \neq \perp$ and let $\mathsf{Decap}(\mathsf{sk}_1, C) = \perp$. This would mean that $f_1(\mathsf{PK}, k) = f_2(\mathsf{PK}, H(\mathsf{pk}_1))$. But then the pair $(k, H(\mathsf{pk}_1))$ is a claw w.r.t. $f_1(\mathsf{PK}, .)$ and $f_2(\mathsf{PK}, .)$ which breaks the underlying claw-freeness assumption of $(\mathcal{F}_1, \mathcal{F}_2)$.

Now let $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ be an ROR-CCA secure AEAD which is additionally XROB-secure. We describe and then analyse an adversary $\mathcal{A}$ for the ANO-CCA security game against the hybrid PKE scheme resulting from the composition of $\overline{\mathsf{KEM}}$ and DEM.

Upon receiving two public-keys pk_0 and pk_1 (along with the public parameters $f_1(\mathsf{PK}, \cdot)$ and $f_2(\mathsf{PK}, \cdot)$), $\mathcal{A}$ selects an arbitrary message m and forwards the challenge message m in the ANO-CCA game. It then receives the ciphertext $C = (C_{\mathsf{KEM}}, C_{\mathsf{DEM}})$ where $(C_{\mathsf{KEM}}, k) \leftarrow\!\!\$\ \overline{\mathsf{Encap}}(\mathsf{pk}_b)$ and $C_{\mathsf{DEM}} \leftarrow\!\!\$\ \mathsf{Enc}(k, m)$, for a uniformly random bit $b \leftarrow\!\!\$\ \{0, 1\}$. Then, $\mathcal{A}$ asks for the decryption of ciphertext $C' = (C_{\mathsf{KEM}}, C'_{\mathsf{DEM}})$ w.r.t. sk_0 where $C'_{\mathsf{DEM}} = \mathsf{Enc}(\hat{k}, m)$ with $\hat{k} = f_2(\mathsf{PK}, H(\mathsf{pk}_0))$. If the response is $\perp$, then the adversary $\mathcal{A}$ outputs 0; else, it outputs 1.

To see why $\mathcal{A}$ breaks the ANO-CCA security of the hybrid PKE scheme, consider the following 2 cases:

- $\mathbf{b = 0}$: In the decryption of $C' = (C_{\mathsf{KEM}}, C'_{\mathsf{DEM}})$ w.r.t. sk_0, we have that $\mathsf{Decap}(\mathsf{sk}_0, C_{\mathsf{KEM}}) = k'$ where $f_1(\mathsf{PK}, k') = k$. Therefore, we have $f_1(\mathsf{PK}, k') = k \neq f_2(\mathsf{PK}, H(\mathsf{pk}_0))$ (i.e., $k \neq \hat{k}$) with a high probability owing to the claw-freeness of $(\mathcal{F}_1, \mathcal{F}_2)$. Since DEM is XROB-secure, we also have $\mathsf{Dec}(k, \mathsf{Enc}(\hat{k}, m)) = \perp$ with a high probability. Hence, the adversary guesses correctly by outputting 0.
- $\mathbf{b = 1}$: In the decryption of $C' = (C_{\mathsf{KEM}}, C'_{\mathsf{DEM}})$ w.r.t. sk_0, we have that $\mathsf{Decap}(\mathsf{sk}_0, C_{\mathsf{KEM}}) = \perp$ with a high probability because the underlying KEM is SROB-CCA secure (note that $\mathsf{Encap}(\mathsf{pk}_1) = (C_{\mathsf{KEM}}, k')$ where $f_1(\mathsf{PK}, k') = k$). Because of the way $\overline{\mathsf{KEM}}$ was constructed, we thus have $\overline{\mathsf{Decap}}(\mathsf{sk}_0, C_{\mathsf{KEM}}) = f_2(\mathsf{PK}, H(\mathsf{pk}_0))(= \hat{k})$. Therefore, we have $\mathsf{Dec}(\hat{k}, \mathsf{Enc}(\hat{k}, m)) = m \neq \perp$. Again, the adversary guesses correctly by outputting 1.

C.5 Proof of Theorem 5

Theorem 5. *Suppose* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct and has message space* $\mathcal{M}$*. Then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ *issuing at most* q_G[6] *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*) and at most* q_D *queries to the (classical) decapsulation oracles, there exist* wANO-CPA *adversary* $\mathcal{B}$ *and* OW-CPA *adversary* $\mathcal{C}$ *against* PKE*, and* SCFR-CPA *adversary* $\mathcal{D}$ *against* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ *issuing at most* q_G *queries to* G*, such that:*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})} + q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D + 2)\sqrt{2\delta}\,.$$

[6] Following [24, 27], we make the convention that the number q_O of queries made by an adversary $\mathcal{A}$ to a random oracle O counts the total number of times O is executed in the corresponding security experiment; i.e., the number of $\mathcal{A}$'s explicit queries to O plus the number of implicit queries to O made by the experiment.

Moreover, the running times of $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ are the same as that of $\mathcal{A}$.

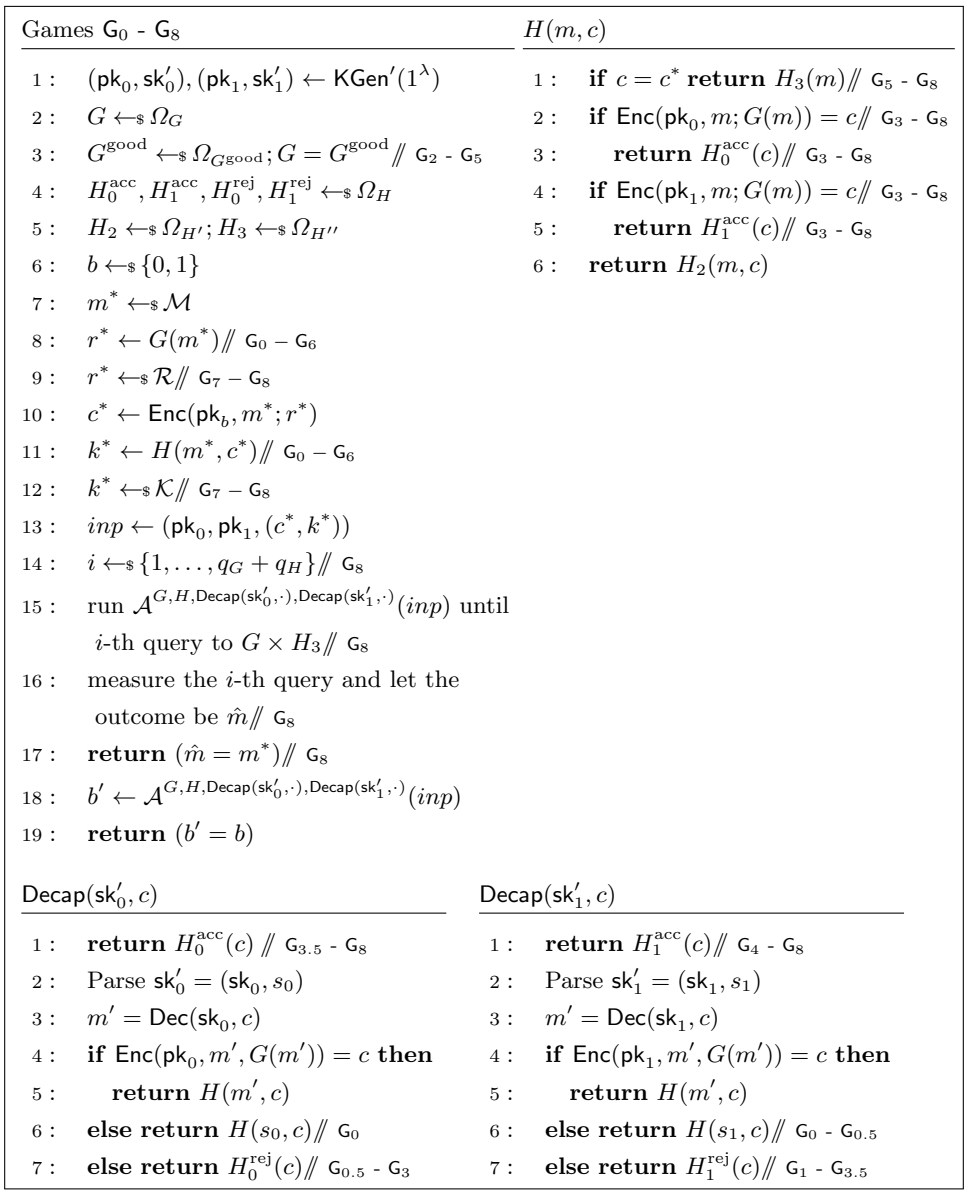


Fig. 13. Games G_0 – G_8 for the proof of Theorem 5.

Proof. Denote Ω_G, Ω_H, $\Omega_{H'}$ and $\Omega_{H''}$ to be the set of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \overline{\mathcal{C}} \to \mathcal{K}$, $H' : \mathcal{M} \times \overline{\mathcal{C}} \to \mathcal{K}$ and $H'' : \mathcal{M} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{K}$ is the encapsulated key-space of $\mathsf{KEM}^{\not\perp}$ and $\overline{\mathcal{C}}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$.

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for $\mathsf{KEM}^{\not\perp}$ issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G (resp., q_H) quantum queries to the random oracles G (resp. H). Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_8$ described in Figure 13.

Game G_0 The game G_0 is exactly the ANO-CCA game for $\mathsf{KEM}^{\not\perp}$ ($= \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$). Hence,

$$\left| \Pr[\mathsf{G}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A})$$

Game $\mathsf{G}_{0.5}$ In game $\mathsf{G}_{0.5}$, we modify the decapsulation oracle $\mathsf{Decap}(\mathsf{sk}_0', \cdot)$ such that $H_0^{\mathrm{rej}}(c)$ is returned instead of $H(s_0, c)$ for an invalid ciphertext c. That is, pseudo-random decapsulation of invalid ciphertexts (w.r.t. sk_0) are replaced by truly random outputs. Define an oracle algorithm A^{H_2,F_i} $(i \in \{0,1\})$ as described in Figure 14. Let $F_0(\cdot) = H_2(s_0, \cdot)$ for secret $s_0 \leftarrow_{\$} \mathcal{M}$ and $F_1(\cdot) = H_0^{\mathrm{rej}}(\cdot)$, where H_2 and H_0^{rej} are chosen in the same way as in G_0 and $\mathsf{G}_{0.5}$. Then note that, $\Pr[\mathsf{G}_0 = 1] = \Pr[1 \leftarrow A^{H_2,F_0}]$ and $\Pr[\mathsf{G}_{0.5} = 1] = \Pr[1 \leftarrow A^{H_2,F_1}]$. Since the uniform secret s_0 is chosen independently from A^{H_2,F_i}'s view, we use Lemma 3 to obtain

$$|\Pr[\mathsf{G}_{0.5} = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{2q_H}{\sqrt{|\mathcal{M}|}}$$

$A^{H_2,F_i}(1^\lambda)$

1 : $(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KGen}(1^\lambda)$ // s_0 implicitly defined
2 : $(\mathsf{pk}_1, \mathsf{sk}_1') \leftarrow \mathsf{KGen}'(1^\lambda)$ // $\mathsf{sk}_1' = (\mathsf{sk}_1, s_1)$
3 : $G \leftarrow_{\$} \Omega_G$
4 : $b \leftarrow_{\$} \{0,1\}$
5 : $m^* \leftarrow_{\$} \mathcal{M}$
6 : $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; G(m^*))$
7 : $k^* \leftarrow H(m^*, c^*)$
8 : $b' \leftarrow \mathcal{A}^{G,H,\mathsf{Decap}(\mathsf{sk}_0',\cdot),\mathsf{Decap}(\mathsf{sk}_1',\cdot)}(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
9 : **return** $(b' = b)$

$H(m, c)$

1 : **return** $H_2(m, c)$

$\mathsf{Decap}(\mathsf{sk}_0', c)$

1 : $m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
2 : **if** $\mathsf{Enc}(\mathsf{pk}_0, m', G(m')) = c$
3 : **return** $H(m', c)$
4 : **else return** $F_i(c)$

$\mathsf{Decap}(\mathsf{sk}_1', c)$

1 : Parse $\mathsf{sk}_1' = (\mathsf{sk}_1, s_1)$
2 : $m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
3 : **if** $\mathsf{Enc}(\mathsf{pk}_1, m', G(m')) = c$
4 : **return** $H(m', c)$
5 : **else return** $H(s_1, c)$

Fig. 14. Algorithm A^{H_2,F_i} for the proof of Theorem 5.

Game G_1 In game G_1, we modify the decapsulation oracle $\mathsf{Decap}(\mathsf{sk}_1', \cdot)$ such that $H_1^{\mathrm{rej}}(c)$ is returned instead of $H(s_1, c)$ for an invalid ciphertext c. Using Lemma 3 in a similar manner as the previous "game-hop", it is not hard to obtain

$$|\Pr[\mathrm{G}_1^{\mathcal{A}} \to 1] - \Pr[\mathrm{G}_{0.5}^{\mathcal{A}} \to 1]| \leq \frac{2q_H}{\sqrt{|\mathcal{M}|}}$$

Game G_2 In game G_2, we change the random oracle G such that it uniformly samples "good" random coins w.r.t. the key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$. To be specific, given a PKE key-pair $(\mathsf{pk}, \mathsf{sk})$ and a message $m \in \mathcal{M}$, define

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

and $\mathcal{R}_{\mathrm{bad}}((\mathsf{pk}, \mathsf{sk}), m) = \mathcal{R} \setminus \mathcal{R}_{\mathrm{good}}((\mathsf{pk}, \mathsf{sk}), m)$. Now w.r.t. the key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$, denote $\Omega_{G^{\mathrm{good}}}$ to be the set of all random functions G^{good} such that $G^{\mathrm{good}}(m)$ is sampled according to a uniform distribution in $(\mathcal{R}_{\mathrm{good}}(\mathsf{pk}_0, \mathsf{sk}_0, m) \cap \mathcal{R}_{\mathrm{good}}(\mathsf{pk}_1, \mathsf{sk}_1, m))$. Hence in G_2, we replace the oracle G with G^{good}. Note that the task of distinguishing between G_1 and G_2 is equivalent to that of distinguishing between the oracles G and G^{good}. To be specific, we can construct a distinguisher $B^{\hat{G}}((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1))$ between G and G^{good} that simulates the adversary $\mathcal{A}$'s view in games G_1 or G_2 by using the oracle $\hat{G}$. That is, for any two fixed key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$, $(\mathsf{pk}_1, \mathsf{sk}_1)$ generated by KGen, if $\hat{G} = G$, $B^{\hat{G}}((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1))$ simulates G_1, and if $\hat{G} = G^{\mathrm{good}}$, $B^{\hat{G}}((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1))$ perfectly simulates G_2. Therefore,

$$\begin{aligned}
&|\Pr[\mathsf{G}_2 = 1 \mid (\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}] - \Pr[\mathsf{G}_1 = 1 \mid (\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}]| \\
&= |\Pr[1 \leftarrow B^{G^{\mathrm{good}}}((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1))] - \Pr[1 \leftarrow B^{G}((\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1))]|
\end{aligned}$$

Now any distinguisher between G and G^{good} can be converted to a distinguisher between N_1 and N_2 where N_1 is a function such that $N_1(m)$ is sampled according to the Bernoulli distribution $B_{\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m)}$, i.e., $\Pr[N_1(m) = 1] = \delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m)$ (resp. $\Pr[N_1(m) = 0] = 1 - \delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m)$), where

$$\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m) = \frac{|\mathcal{R}_{\text{bad}}(\mathsf{pk}_0,\mathsf{sk}_0,m) \cup \mathcal{R}_{\text{bad}}(\mathsf{pk}_1,\mathsf{sk}_1,m)|}{|\mathcal{R}|}$$

and N_2 is a constant function that always outputs 0 for any input m. Specifically, for any distinguisher $B^{\hat{G}}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))$, we can construct a distinguisher $C^N((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))$ that is described in Figure 15. $\mathsf{Sample}(\mathcal{Y})$ is a probabilistic algorithm that returns a uniformly distributed $y \leftarrow_{\$} \mathcal{Y}$ and $\mathsf{Sample}(\mathcal{Y}; f(m))$ denotes the deterministic execution of $\mathsf{Sample}(\mathcal{Y})$ using explicit randomness $f(m)$.

$C^N((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))$

1 : Pick a $2q_G$-wise function f
2 : $b'' \leftarrow B^{\hat{G}}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))$
3 : **return** b''

$\hat{G}(m)$

1 : **if** $N(m) = 0$
2 : $\quad \hat{G}(m) = \mathsf{Sample}(\mathcal{R}_{\text{good}}(\mathsf{pk}_0,\mathsf{sk}_0,m) \cap \mathcal{R}_{\text{good}}(\mathsf{pk}_1,\mathsf{sk}_1,m); f(m))$
3 : **else**
4 : $\quad \hat{G}(m) = \mathsf{Sample}(\mathcal{R}_{\text{bad}}(\mathsf{pk}_0,\mathsf{sk}_0,m) \cup \mathcal{R}_{\text{bad}}(\mathsf{pk}_1,\mathsf{sk}_1,m); f(m))$
5 : **return** $\hat{G}(m)$

Fig. 15. Algorithm C^N for the proof of Theorem 5.

Note that if $N = N_1$, then $\hat{G} = G$, and if $N = N_2$, then $\hat{G} = G^{\text{good}}$. Therefore, for any two fixed key-pairs $(\mathsf{pk}_0,\mathsf{sk}_0)$, $(\mathsf{pk}_1,\mathsf{sk}_1)$ generated by KGen, we have $\Pr[1 \leftarrow C^{N_1}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))] = \Pr[1 \leftarrow B^{G}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))]$ and $\Pr[1 \leftarrow C^{N_2}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))] = \Pr[1 \leftarrow B^{G^{\text{good}}}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))]$. Hence, from Lemma 5, we have

$$\begin{aligned}
&|\Pr[1 \leftarrow B^{G^{\text{good}}}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))] - \Pr[1 \leftarrow B^{G}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))]| \\
&= |\Pr[1 \leftarrow C^{N_2}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))] - \Pr[1 \leftarrow C^{N_1}((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))]| \\
&\qquad \leq 2q_G\sqrt{\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))}
\end{aligned}$$

where $\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)) = \max_{m\in\mathcal{M}} \delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m)$. Hence, conditioned on two fixed key-pairs $(\mathsf{pk}_0,\mathsf{sk}_0)$, $(\mathsf{pk}_1,\mathsf{sk}_1)$ generated by KGen, we obtain

$$\begin{aligned}
&|\Pr[\mathsf{G}_2 = 1 \,|\, (\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}] - \Pr[\mathsf{G}_1 = 1 \,|\, (\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}]| \\
&\qquad \leq 2q_G\sqrt{\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))}
\end{aligned}$$

Averaging over $(\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KGen}$, $(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}$, and applying Jensen's inequality w.r.t. the square root function, we get

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq 2q_G\sqrt{\mathbf{E}[\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))]}$$

where the expectation is taken over $(\mathsf{pk}_0,\mathsf{sk}_0) \leftarrow \mathsf{KGen}$, $(\mathsf{pk}_1,\mathsf{sk}_1) \leftarrow \mathsf{KGen}$. From the notion of δ-correctness, note that for a *single* key-pair $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KGen}$, $\mathbf{E}[\delta((\mathsf{pk},\mathsf{sk}))] = \delta$, where $\delta((\mathsf{pk},\mathsf{sk})) = \max_{m\in\mathcal{M}} \delta((\mathsf{pk},\mathsf{sk}),m)$ and $\delta((\mathsf{pk},\mathsf{sk}),m) = \frac{\mathcal{R}_{\text{bad}}(\mathsf{pk},\mathsf{sk},m)}{\mathcal{R}}$. We now show that for two key-pairs, $\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)) \leq 2\delta$. First

note that, for a particular message m, $\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1),m) \leq \delta((\mathsf{pk}_0,\mathsf{sk}_0),m) + \delta((\mathsf{pk}_1,\mathsf{sk}_1),m)$, and hence, $\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)) \leq \delta((\mathsf{pk}_0,\mathsf{sk}_0)) + \delta((\mathsf{pk}_1,\mathsf{sk}_1))$. We now have the following

$$\begin{aligned}
\mathbf{E}[\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1))] &= \sum_{\substack{(\mathsf{pk}_0,\mathsf{sk}_0)\\(\mathsf{pk}_1,\mathsf{sk}_1)}} \Pr[(\mathsf{pk}_0,\mathsf{sk}_0)]\Pr[(\mathsf{pk}_1,\mathsf{sk}_1)]\delta((\mathsf{pk}_0,\mathsf{sk}_0),(\mathsf{pk}_1,\mathsf{sk}_1)) \\
&\leq \sum_{\substack{(\mathsf{pk}_0,\mathsf{sk}_0)\\(\mathsf{pk}_1,\mathsf{sk}_1)}} \Pr[(\mathsf{pk}_0,\mathsf{sk}_0)]\Pr[(\mathsf{pk}_1,\mathsf{sk}_1)](\delta((\mathsf{pk}_0,\mathsf{sk}_0)) + \delta((\mathsf{pk}_1,\mathsf{sk}_1))) \\
&= \sum_{(\mathsf{pk}_1,\mathsf{sk}_1)} \Big(\sum_{(\mathsf{pk}_0,\mathsf{sk}_0)} \Pr[(\mathsf{pk}_0,\mathsf{sk}_0)]\delta((\mathsf{pk}_0,\mathsf{sk}_0)) \Big) \Pr[(\mathsf{pk}_1,\mathsf{sk}_1)] \\
&+ \sum_{(\mathsf{pk}_0,\mathsf{sk}_0)} \Big(\sum_{(\mathsf{pk}_1,\mathsf{sk}_1)} \Pr[(\mathsf{pk}_1,\mathsf{sk}_1)]\delta((\mathsf{pk}_1,\mathsf{sk}_1)) \Big) \Pr[(\mathsf{pk}_0,\mathsf{sk}_0)] \\
&= \sum_{(\mathsf{pk}_1,\mathsf{sk}_1)} \delta \cdot \Pr[(\mathsf{pk}_1,\mathsf{sk}_1)] + \sum_{(\mathsf{pk}_0,\mathsf{sk}_0)} \delta \cdot \Pr[(\mathsf{pk}_0,\mathsf{sk}_0)] = 2\delta
\end{aligned}$$

where $\Pr[(\mathsf{pk}_i,\mathsf{sk}_i)]$ denotes the probability of the fixed key-pair $(\mathsf{pk}_i,\mathsf{sk}_i)$ being generated by KGen. We also used the fact that the key-pairs $(\mathsf{pk}_0,\mathsf{sk}_0)$, $(\mathsf{pk}_1,\mathsf{sk}_1)$ are generated independently. Thus, we finally obtain

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq 2q_G\sqrt{2\delta}$$

Game G_3 In game G_3, we implicitly divide the H-queries (m,c) into three disjoint categories: (1) $\mathsf{Enc}(\mathsf{pk}_0,m;G(m)) = c$, (2) $\mathsf{Enc}(\mathsf{pk}_0,m;G(m)) \neq c = \mathsf{Enc}(\mathsf{pk}_1,m;G(m))$, and (3) $\mathsf{Enc}(\mathsf{pk}_0,m;G(m)) \neq c \wedge \mathsf{Enc}(\mathsf{pk}_1,m;G(m)) \neq c$. We then respond to the queries from the respective categories with $H_0^{\mathrm{acc}}(c)$, $H_1^{\mathrm{acc}}(c)$ and $H_2(m,c)$ respectively, where H_0^{acc} and H_1^{acc} are internal random functions not directly accessible to the adversary $\mathcal{A}$. Because G samples "good" random coins, it is not hard to see that the encryption functions $\mathsf{Enc}(\mathsf{pk}_0,.;G(\cdot))$ and $\mathsf{Enc}(\mathsf{pk}_1,.;G(\cdot))$ are injective, and hence, the output distributions of the H-oracle in the games G_2 and G_3 are equivalent. Therefore,

$$\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_2 = 1]$$

Game $\mathsf{G}_{3.5}$ In game $\mathsf{G}_{3.5}$, we change the $\mathsf{Decap}(\mathsf{sk}_0',\cdot)$ oracle such that there is no need for the secret key sk_0'. Namely, $H_0^{\mathrm{acc}}(c)$ is returned for the decapsulation of ciphertext c w.r.t. sk_0'. Let $m' = \mathsf{Dec}(\mathsf{sk}_0,c)$. Consider the following two cases:

- $\mathsf{Enc}(\mathsf{pk}_0,m';G(m')) = c$. In this case, the $\mathsf{Decap}(\mathsf{sk}_0',\cdot)$ oracles in games G_3 and $\mathsf{G}_{3.5}$ return the same value $H_0^{\mathrm{acc}}(c)$.
- $\mathsf{Enc}(\mathsf{pk}_0,m';G(m')) \neq c$. In game G_3, as the random function H_0^{rej} is independent of all other oracles, the output $H_0^{\mathrm{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game $\mathsf{G}_{3.5}$, the only way $\mathcal{A}$ gets prior access to the function H_0^{acc} is if it made a H-query (m'',c) such that $\mathsf{Enc}(\mathsf{pk}_0,m'';G(m'') = c$. But because G samples good random coins, we have $\mathsf{Dec}(\mathsf{sk}_0,c) = m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}_0,m';G(m')) \neq c$". Therefore, such a prior access is not possible and $H_0^{\mathrm{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of the $\mathsf{Decap}(\mathsf{sk}_0',\cdot)$ oracle in G_3 and $\mathsf{G}_{3.5}$ are the same in both cases, we have

$$\Pr[\mathsf{G}_{3.5} = 1] = \Pr[\mathsf{G}_3 = 1]$$

Game G_4 In game G_4, we change the $\mathsf{Decap}(\mathsf{sk}_1',\cdot)$ oracle such that $H_1^{\mathrm{acc}}(c)$ is returned for the decapsulation of *any* ciphertext c w.r.t. sk_1. The analysis here follows quite similarly to that of the previous game-hop except that this simulation of the $\mathsf{Decap}(\mathsf{sk}_1',\cdot)$ oracle – without the secret key sk_1' – will fail if $\mathcal{A}$ asks for the decapsulation of a ciphertext $\hat{c}$ such that $m' = \mathsf{Dec}(\mathsf{sk}_1,\hat{c})$ and $\mathsf{Enc}(\mathsf{pk}_0,m';G(m')) = \mathsf{Enc}(\mathsf{pk}_1,m';G(m')) = \hat{c}$. In this peculiar case, $H_0^{\mathrm{acc}}(\hat{c})$ is returned in G_3 and $H_1^{\mathrm{acc}}(\hat{c})$ is returned in G_4.

We bound the probability of this peculiar event (i.e., $\mathcal{A}$ asking for the decapsulation of such an above ciphertext $\hat{c}$ w.r.t. sk_1') by the advantage of an SCFR-CPA adversary $\mathcal{D}$ against the deterministic scheme

$\mathsf{PKE}_1^{\text{good}} = \mathsf{T}[\mathsf{PKE}, G^{\text{good}}]$. First note that, because G^{good} samples good random coins, for such ciphertexts $\hat{c}$ we have $\mathsf{Dec}(sk_0, \hat{c}) = \mathsf{Dec}(sk_1, \hat{c}) = m'$ and $\mathsf{Enc}(\mathsf{pk}_0, m'; G^{\text{good}}(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G^{\text{good}}(m')) = \hat{c}$. Note that such a $\hat{c}$ corresponds to winning the SCFR-CPA game of $\mathsf{PKE}_1^{\text{good}}$. So we can construct a corresponding SCFR-CPA adversary $\mathcal{D}$ that has access to the (non-ideal) "good" random oracle G^{good}. Upon receiving two public-keys pk_0 and pk_1, $\mathcal{D}$ simulates G_4 for the adversary $\mathcal{A}$ and maintains a list of $\mathcal{A}$'s *classical* queries to the oracle $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ (note that $\mathcal{D}$ can simulate the two decapsulation oracles as in G_4 even with no access to the corresponding secret keys sk_0 and sk_1). Then $\mathcal{D}$ chooses a ciphertext uniformly at random from the list and forwards it as the final message to the SCFR-CPA challenger of $\text{PKE}_1^{\text{good}}$.

Let $\Pr[\mathcal{P}]$ be the probability of this peculiar event, denoted as $\mathcal{P}$, occurring. We have the games $\mathsf{G}_{3.5}$ and G_4 to be equivalent unless the event $\mathcal{P}$ occurs. From the construction of the SCFR-CPA adversary $\mathcal{D}$ above, it is not hard to see that $\mathbf{Adv}^{\text{SCFR-CPA}}_{\text{PKE}_1^{\text{good}}}(\mathcal{D}) \geq \frac{1}{q_D} \cdot \Pr[\mathcal{P}]$. Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_{3.5} = 1]| \leq \Pr[\mathcal{P}] \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\text{PKE}_1^{\text{good}}}(\mathcal{D})$$

Using a similar analysis as the game-hop $\mathsf{G}_1 \rightarrow \mathsf{G}_2$, by replacing G^{good} with an ideal random oracle G w.r.t. the SCFR-CPA adversary $\mathcal{D}$, we obtain

$$|\Pr[\mathsf{G}_4^{\mathcal{A}} \rightarrow 1] - \Pr[\mathsf{G}_{3.5}^{\mathcal{A}} \rightarrow 1]| \leq q_D \cdot (\mathbf{Adv}^{\text{SCFR-CPA}}_{\text{PKE}_1}(\mathcal{D}) + 2q_G\sqrt{2\delta})$$

Game G_5 In game G_5, we answer H-queries of the form (m, c^*) with $H_3(m)$, where H_3 is an independent random function. Since G samples good randomness, there are at most two H-queries worth considering, namely (m_0, c^*) and (m_1, c^*), where $\mathsf{Enc}(\mathsf{pk}_0, m_0; G(m_0)) = c^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G(m_1)) = c^*$ (for the other H-queries (m', c^*), where $m' \notin \{m_0, m_1\}$. we are replacing the oracle outputs $H_2(m', c^*)$ in G_4 with $H_3(m')$ in G_5). W.r.t. these two queries, the H oracle would return $H_0^{\text{acc}}(c^*)$, $H_1^{\text{acc}}(c^*)$ respectively in G_4, and $H_3(m_0)$, $H_3(m_1)$ respectively in G_5. The adversary $\mathcal{A}$'s view would be identical even after this change because the random values $H_0^{\text{acc}}(c^*)$, $H_1^{\text{acc}}(c^*)$ are only accessible to $\mathcal{A}$ via the H-oracle in G_4, and in particular, not through the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles since c^* is a forbidden decapsulation query. Hence in G_5, we are effectively replacing (at most) two uniformly random values that can only be accessed via the H-oracle by $\mathcal{A}$ with two other uniformly random values (the simpler case of $m_0 = m_1$ would follow similarly). Since the output distributions of the H-oracle in the games G_4 and G_5 are equivalent, we have

$$\Pr[\mathsf{G}_5 = 1] = \Pr[\mathsf{G}_4 = 1]$$

Game G_6 In game G_6, we reset G to be an ideal random oracle, i.e., $G(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_1 \rightarrow \mathsf{G}_2$, by using a similar analysis, it is not hard to obtain

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq 2q_G\sqrt{2\delta}$$

Game G_7 In the setup of game G_7, we replace the hash evaluations "$r^* \leftarrow G(m^*)$" and "$k^* \leftarrow H(m^*, c^*)(= H_3(m^*))$" with "$r^* \leftarrow_{\$} \mathcal{R}$" and "$k^* \leftarrow_{\$} \mathcal{K}$" respectively. That is, r^* and k^* are now uniformly random values that are generated independently of the random oracles G and H_3. We use Lemma 4 to bound the difference in the success probabilities of $\mathcal{A}$ in G_6 and G_7. Let A be an oracle algorithm that has quantum access to the random oracle $G \times H_3$, where $(G \times H_3)(m) = (G(m), H_3(m))$. Figure 16 describes $A^{G \times H_3}$'s operation on input $(m^*, (r^*, k^*))$. Note that the algorithm $A^{G \times H_3}$ makes at most $q_G + q_H$ number of queries to the random oracle $G \times H_3$ to respond to $\mathcal{A}$'s oracle queries[7].

Let B be an oracle algorithm that on input m^* does the following: picks $i \leftarrow_{\$} \{1, \ldots, q_G + q_H\}$, generates $r^* \leftarrow_{\$} \mathcal{R}$ and $k^* \leftarrow_{\$} \mathcal{K}$, runs $A^{G \times H_3}(m^*, (r^*, k^*))$ until the i-th query, measures the argument of the $(G \times H_3)$-query in the computational basis and outputs the measurement outcome (if $A^{G \times H_3}$ makes less than i queries, B outputs $\perp$). With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_6 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_7 = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 4 w.r.t. the algorithm $A^{G \times H_3}$. Therefore, we now define game G_8 (see Fig. 13) such that $P_B = \Pr[\mathsf{G}_8 = 1]$, where P_B is as defined in Lemma 4 w.r.t. the algorithm $B^{G \times H_3}$. From Lemma 4, we thus have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq 2(q_G + q_H)\sqrt{\Pr[\mathsf{G}_8 = 1]}$$

[7] For example, if $A^{G \times H_3}$ wants to respond to $\mathcal{A}$'s H-query, then $A^{G \times H_3}$ prepares a uniform superposition of all states in the output register corresponding to G (see [38] for particulars of this "trick").

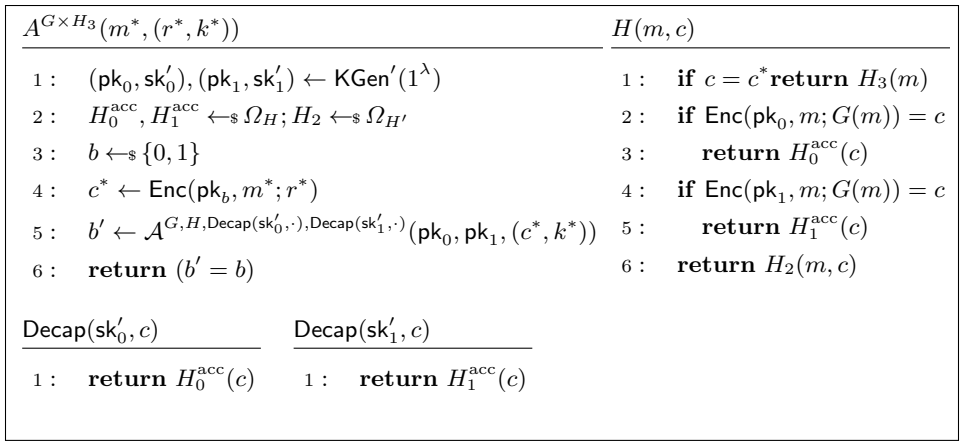


Fig. 16. Algorithm $A^{G\times H_3}$ for the proof of Theorem 5.

We now bound the success probability of $\mathcal{A}$ in G_7 by the advantage of an adversary $\mathcal{B}$ in the wANO-CPA game of PKE. Upon receiving public-keys pk_0 and pk_1 along with the ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ for uniformly random bit $b(\leftarrow_{\$}\{0,1\})$, (secret) message $m^*(\leftarrow_{\$}\mathcal{M})$ and randomness $r^*(\leftarrow_{\$}\mathcal{R})$ chosen by the challenger, $\mathcal{B}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_7.
- Uses a $2q_G$-wise independent function and four different $2q_H$-wise independent functions to perfectly simulate the random oracles $G, H_0^{\mathrm{acc}}, H_1^{\mathrm{acc}}, H_2$ and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. The random oracle H is simulated in the same way as in G_7.
- Answers decapsulation queries using the oracles H_i^{acc} $(i\in\{0,1\})$ as in G_7.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^*\leftarrow_{\$}\mathcal{K}$ and responds with $(\mathsf{pk}_0,\mathsf{pk}_1,(c^*,k^*))$.
- After obtaining a bit b' from $\mathcal{A}$, forwards b' to its wANO-CPA challenger as the final message.

It is easy to see that $|\Pr[\mathsf{G}_7=1]-\frac{1}{2}| = \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B})$. Now we bound the success probability of $\mathcal{A}$ in G_8 by the advantage of an adversary $\mathcal{C}$ in the OW-CPA game of PKE. Upon receiving a public-key pk along with a ciphertext c^*, where $c^*\leftarrow\mathsf{Enc}(\mathsf{pk},m^*;r^*)$ for uniformly random (secret) message $m^*(\leftarrow_{\$}\mathcal{M})$ and randomness $r^*(\leftarrow_{\$}\mathcal{R})$ chosen by the challenger, $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_8.
- Uses a $2q_G$-wise independent function and four different $2q_H$-wise independent functions to perfectly simulate the random oracles $G, H_0^{\mathrm{acc}}, H_1^{\mathrm{acc}}, H_2$ and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also evaluates $\mathcal{A}$'s G- and H-queries using the oracle $G\times H_3$; the random oracle H is simulated in the same way as in G_8,
- Answers decapsulation queries using the oracles H_i^{acc} $(i\in\{0,1\})$ as in G_8.
- For $\mathcal{A}$'s challenge query, first samples a uniformly random bit $b\leftarrow_{\$}\{0,1\}$ and sets $\mathsf{pk}_b=\mathsf{pk}$. Then generates a key-pair $(\mathsf{pk}_{1-b},\mathsf{sk}_{1-b})\leftarrow\mathsf{KGen}(1^\lambda)$, samples a uniformly random key $k^*\leftarrow_{\$}\mathcal{K}$ and responds with $(\mathsf{pk}_0,\mathsf{pk}_1,(c^*,k^*))$. (By doing this, note that we have $c^*\leftarrow\mathsf{Enc}(\mathsf{pk}_b,m^*;r^*)$ in $\mathcal{A}$'s view.)
- Selects $i\leftarrow_{\$}\{1,\ldots,q_G+q_H\}$, measures the i-th query to oracle $G\times H_3$ and returns the outcome $\hat{m}$.

Again, it is not hard to see that $\Pr[\mathsf{G}_8=1]\le\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})$. Hence by collecting all of the above bounds, we arrive at

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \le{}& \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_G+q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})}\\ &+ q_D\cdot\mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D+2)\sqrt{2\delta}\end{aligned}$$

C.6 Proof of Theorem 6

Theorem 6. *Suppose* $\mathsf{PKE}=(\mathsf{KGen},\mathsf{Enc},\mathsf{Dec})$ *is* δ*-correct. Then for any* SCFR-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\not\perp}=\mathsf{FO}^{\not\perp}[\mathsf{PKE},G,H]$ *issuing at most* q_D *queries to the (classical) decapsulation oracles, at most* q_G

(resp. q_H) queries to the quantum random oracle G (resp. H), there exists an SCFR-CPA *adversary* $\mathcal{B}$ *against* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ *issuing at most* q_G *queries to* G *such that*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{\alpha(q_H+1)^3}{|\mathcal{K}|} + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D+2)\sqrt{2\delta}\,.$$

Here $\mathcal{K}$ *denotes the encapsulated key-space of* $\mathsf{KEM}^{\not\perp}$ *and* α (< 648) *is the constant from Lemma 1. The running time of* $\mathcal{B}$ *is the same as that of* $\mathcal{A}$*.*

Games G_0 - G_5

$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'(1^\lambda)$
$G \leftarrow\!\!\$\ \Omega_G$
$G^{\text{good}} \leftarrow\!\!\$\ \Omega_{G^{\text{good}}}; G = G^{\text{good}} /\!\!/ \ \mathsf{G}_2$ - G_4
$H_0^{\text{acc}}, H_1^{\text{acc}}, H_0^{\text{rej}}, H_1^{\text{rej}} \leftarrow\!\!\$\ \Omega_H$
$H_2 \leftarrow\!\!\$\ \Omega_{H'}$
$inp \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
$c \leftarrow \mathcal{A}^{G, H, \mathsf{Decap}(\mathsf{sk}'_0, \cdot), \mathsf{Decap}(\mathsf{sk}'_1, \cdot)}(inp)$
return $(\mathsf{Decap}(\mathsf{sk}'_0, c) = \mathsf{Decap}(\mathsf{sk}'_1, c))$

$H(m, c)$

if $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c /\!\!/ \ \mathsf{G}_3$ - G_5
  return $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_3$ - G_5
if $\mathsf{Enc}(\mathsf{pk}_1, m; G(m)) = c /\!\!/ \ \mathsf{G}_3$ - G_5
  return $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_3$ - G_5
return $H_2(m, c)$

$\mathsf{Decap}(\mathsf{sk}'_0, c)$

return $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_{3.5}$ - G_5
Parse $\mathsf{sk}'_0 = (\mathsf{sk}_0, s_0)$
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
if $\mathsf{Enc}(\mathsf{pk}_0, m', G(m')) = c$ **then**
  return $H(m', c)$
else return $H(s_0, c) /\!\!/ \ \mathsf{G}_0$
else return $H_0^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_{0.5}$ - G_3

$\mathsf{Decap}(\mathsf{sk}'_1, c)$

return $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
Parse $\mathsf{sk}'_1 = (\mathsf{sk}_1, s_1)$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
if $\mathsf{Enc}(\mathsf{pk}_1, m', G(m')) = c$ **then**
  return $H(m', c)$
else return $H(s_1, c) /\!\!/ \ \mathsf{G}_0$ - $\mathsf{G}_{0.5}$
else return $H_1^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1$ - $\mathsf{G}_{3.5}$

Fig. 17. Games G_0 – G_5 for the proof of Theorem 6.

Proof. Denote Ω_G, Ω_H, $\Omega_{H'}$ to be the set of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \overline{\mathcal{C}} \to \mathcal{K}$, $H' : \mathcal{M} \times \overline{\mathcal{C}} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{K}$ is the encapsulated key-space of $\mathsf{KEM}^{\not\perp}$ and $\overline{\mathcal{C}}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$.

Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for $\mathsf{KEM}^{\not\perp}$ issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G (resp., q_H) quantum queries to the random oracles G (resp. H).

The structure of the proof is very similar to that of Theorem 5. Basically we do the same sequence of game-hops as in the proof of Theorem 5 until the point where we can simulate the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ $(i \in \{0, 1\})$ without requiring the corresponding secret keys sk'_i. In the final game-hop, we reset G to be an ideal random oracle.

To be specific, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_5$ as described in Figure 17. By a similar analysis as that of the proof of Theorem 5 w.r.t. these game-hops, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D+2)\sqrt{2\delta}$$

Note that the game G_0 is exactly the SCFR-CCA game for $\mathsf{KEM}^{\not\perp}$. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A})$$

Coming to the game G_5, note that the adversary $\mathcal{A}$ wins the game if it finally outputs a ciphertext c such that $\mathsf{Decap}(\mathsf{sk}'_0, c) = \mathsf{Decap}(\mathsf{sk}'_1, c)$. Because of the modification of the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles, this winning condition translates to $H_0^{\mathrm{acc}}(c) = H_1^{\mathrm{acc}}(c)$, where H_0^{acc} and H_1^{acc} are independent quantum-accessible random functions. Note that in this case, (c, c) is a *claw* w.r.t. the pair of QROs $H_0^{\mathrm{acc}} : \overline{\mathcal{C}} \to \mathcal{K}$ and $H_1^{\mathrm{acc}} : \overline{\mathcal{C}} \to \mathcal{K}$. Hence we can bound the success probability of $\mathcal{A}$ in G_5 by the advantage of an adversary $\mathcal{C}$ against the *claw-finding* problem w.r.t. the instance $(H_0^{\mathrm{acc}}, H_1^{\mathrm{acc}})$. $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_5.
- Uses a $2q_G$-wise independent function and a $2q_H$-wise independent function to perfectly simulate the random oracles G and H_2 in $\mathcal{A}$'s view, as noted in Lemma 2. Also uses the pair of functions $f_0 : \overline{\mathcal{C}} \to \mathcal{K}$ and $f_1 : \overline{\mathcal{C}} \to \mathcal{K}$ – which is the instance of the claw-finding problem – to simulate the oracles H_0^{acc} and H_1^{acc} respectively.
- Answers decapsulation queries using the oracles $f_i(\cdot)$ $(i \in \{0, 1\})$ as in G_4.
- After obtaining a final ciphertext c from $\mathcal{A}$, forwards (c, c) as the claw w.r.t. (f_0, f_1).

Note that $\mathcal{C}$ makes at most q_H queries to the pair (f_0, f_1). It is easy to see that $\Pr[\mathsf{G}_5 = 1] \leq \frac{\alpha(q_H+1)^3}{|\mathcal{K}|}$ from Lemma 1. Hence, we finally get

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{\alpha(q_H+1)^3}{|\mathcal{K}|} + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 2q_G(q_D+2)\sqrt{2\delta}$$

C.7 Proof of Theorem 7

Theorem 7. *Let* $\mathsf{PKE}^{hy} = (\mathsf{KGen}', \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ *with a one-time authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$*. Suppose* PKE *is* δ*-correct and* γ*-spread (with message space* $\mathcal{M}$*). Then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* PKE^{hy} *issuing at most* q_G *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*), there exist* ANO-CCA *adversary* $\mathcal{B}$ *and* IND-CCA *adversary* $\mathcal{C}$ *against* $\mathsf{KEM}^{\not\perp}$*,* WCFR-CPA *adversary* $\mathcal{D}$ *against* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$*, and* INT-CTXT *adversary* $\mathcal{E}$ *against* DEM *such that:*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{C}) + \mathbf{Adv}^{\text{WCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2^{-\gamma}\,.$$

Moreover, the running times of $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{E}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{D}$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

Proof. The structure of the proof is quite similar to that of Theorem 1, except for some initial game-hops. Here we will be focusing on these hops.

Denote Ω_G, Ω_H and $\Omega_{H'}$ to be the set of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \mathcal{M} \times \overline{\mathcal{C}} \to \mathcal{K}$ and $H' : \overline{\mathcal{C}} \to \mathcal{K}$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{K}$ is the encapsulated key-space of $\mathsf{KEM}^{\not\perp}$ and $\overline{\mathcal{C}}$ is the ciphertext space of $\mathsf{PKE}/\mathsf{KEM}^{\not\perp}$. Let $\mathcal{A}$ be an adversary in the ANO-CCA game for PKE^{hy} issuing at most q_G (resp. q_H) quantum queries to the random oracles G (resp. H). Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_6$ described in Figure 18.

Game G_0**:** The game G_0 is equivalent to the ANO-CCA game for PKE^{hy} (the only "cosmetic" change is that the uniform random bit b is sampled before the adversary $\mathcal{A}$ gets to choose a message $\mathbf{m}$). Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A})$$

Game $\mathsf{G}_{0.3}$**:** In game $\mathsf{G}_{0.3}$, we first make some "cosmetic" changes. Namely, the pair (c_1^*, k^*) resulting from running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b is generated *before* the adversary $\mathcal{A}$ gets to choose a message $\mathbf{m}$. This change does not affect $\mathcal{A}$'s view in any way.

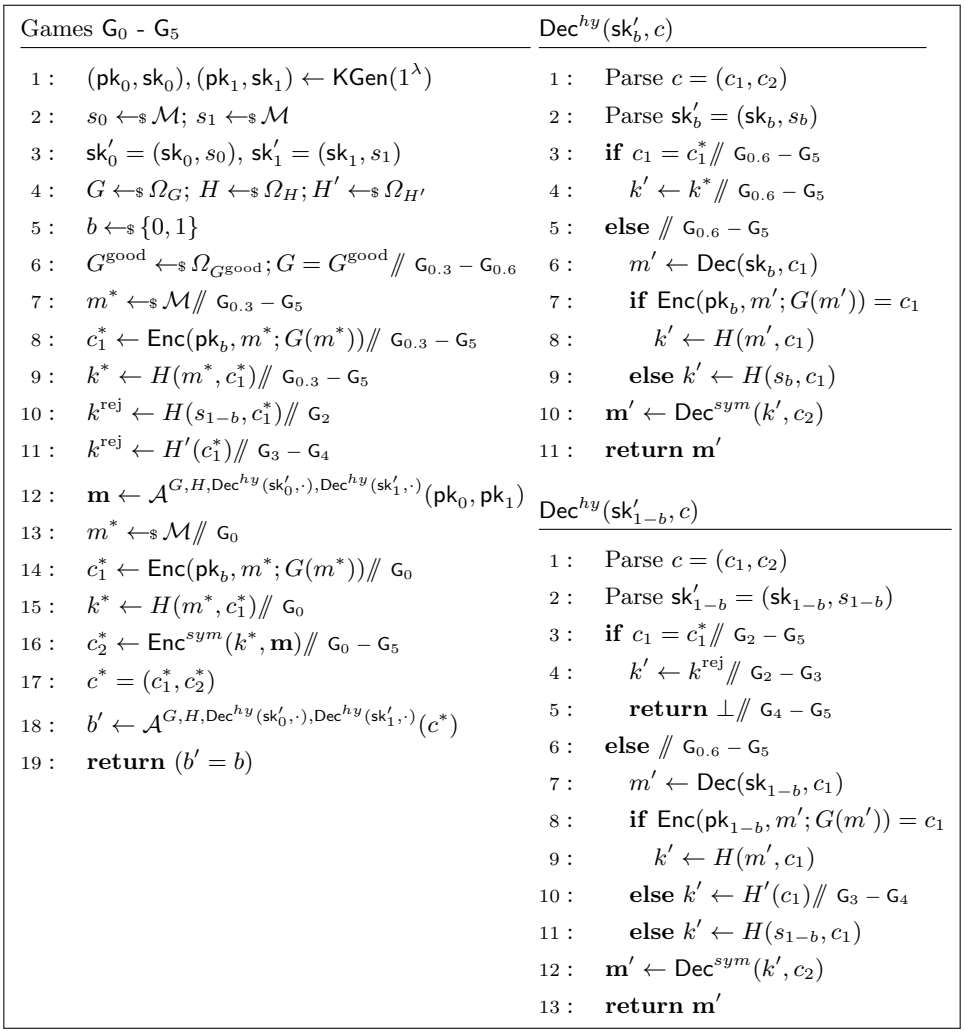


Fig. 18. Games G_0 – G_5 for the proof of Theorem 7.

Next, we change the random oracle G such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_b, \mathsf{sk}_b)$, as seen in the proof of Theorem 5. Specifically, denote $\Omega_{G^{\text{good}}}$ to be the set of all random functions G^{good} such that $G^{\text{good}}(m)$ is sampled according to a uniform distribution in $(\mathcal{R}_{\text{good}}(\mathsf{pk}_b, \mathsf{sk}_b, m)$. Hence in $\mathsf{G}_{0.3}$, we replace the oracle G with G^{good}. By using a similar analysis as the game-hop $(\mathsf{G}_1 \to \mathsf{G}_2)$ in the proof of Theorem 5 (in fact, the analysis would be simpler in this case since we have to consider a single key-pair $(\mathsf{pk}_b, \mathsf{sk}_b)$ instead of two), it is not hard to obtain

$$|\Pr[\mathsf{G}_{0.3} = 1] - \Pr[\mathsf{G}_0 = 1]| \leq 2q_G\sqrt{\delta}$$

Game $\mathsf{G}_{0.6}$: In game $\mathsf{G}_{0.6}$, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}'_b, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$ (and $c_2 \neq c_2^*$), then the oracle uses k^* to decrypt c_2, instead of first decapsulating c_1^* to recover a session key k'. It is not hard to see that the games G_0 and G_1 are equivalent since G samples good random coins, and hence, there is no decapsulation error w.r.t. KEM. Therefore, we have

$$\Pr[\mathsf{G}_{0.6} = 1] = \Pr[\mathsf{G}_{0.3} = 1]$$

Game G_1: In game G_1, we reset G to be an ideal random oracle, i.e., $G(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_0 \to \mathsf{G}_{0.3}$, by using a similar analysis, it is not hard to obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_{0.6} = 1]| \leq 2q_G\sqrt{\delta}$$

Game G_2: In game G_2, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}'_{1-b}, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle uses $k^{\text{rej}} (= H(s_{1-b}, c_1^*))$ to decrypt c_2. Here k^{rej} is the key returned if $\mathsf{Decap}(\mathsf{sk}'_{1-b}, c_1^*)$ would have resulted in an "implicit rejection". Thus, it is not hard to see that the games G_1 and G_2 are equivalent unless c_1^* is not (implicitly) rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation, or in other words, if the following event occurs: $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G(m')) = c_1^*$ where $\mathsf{Enc}(\mathsf{pk}_b, m^*; G(m^*) = c_1^*)$ and $\mathsf{Dec}(\mathsf{sk}_{1-b}, c_1^*) = m'$ (for $m^* \leftarrow_\$ \mathcal{M}$).

There are two sub-events to consider w.r.t. the above event:

1. $\underline{m' \neq m^*}$: In this case, the random oracle G on a new query m' will return uniformly random coins $r \leftarrow_\$ \mathcal{R}$. Since PKE is γ-spread, for the key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b})$ and message m', we have the re-encryption check, namely "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r) = c_1^*$", to hold with probability $\leq 2^{-\gamma}$, for uniformly random r.
2. $\underline{m' = m^*}$: In this case, we can bound the probability of the sub-event occurring by the advantage of an adversary $\mathcal{D}$ in the WCFR-CPA game of PKE_1 $(= \mathsf{T}[\mathsf{PKE}, G])$. The adversary $\mathcal{D}$, upon receiving public-keys pk_0 and pk_1, simply samples a bit b and message m^* uniformly at random, i.e., $b \leftarrow_\$ \{0,1\}$ and $m^* \leftarrow_\$ \mathcal{M}$, and returns (m, b) to the WCFR-CPA challenger (note that only a single query is made to G on m^* in the security experiment).

Hence,

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathbf{Adv}^{\text{WCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + 2^{-\gamma}$$

Note that for the ANO-CCA security of $\mathsf{KEM}^{\not\perp}$, we anyway rely on the SCFR-CPA security of the deterministic PKE_1.

Game G_3: In game G_3, we modify the decryption oracle $\mathsf{Dec}^{hy}(\mathsf{sk}'_{1-b}, \cdot)$ such that the key $H'(c_1)$ is used to decrypt the DEM ciphertext c_2 instead of $H(s_{1-b}, c_1)$ where the KEM ciphertext c_1 was implicitly rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation (H' is an internal random oracle not directly accessible by the adversary $\mathcal{A}$). We also generate the key k^{rej} as "$k^{\text{rej}} \leftarrow H'(c_1^*)$" (instead of "$k^{\text{rej}} \leftarrow H(s_{1-b}, c_1^*)$"). Define an oracle algorithm A^{H,F_i} $(i \in \{0,1\})$ as described in Figure 19. Let $F_0(\cdot) = H(s, \cdot)$ for secret $s \leftarrow_\$ \mathcal{M}$ and $F_1(\cdot) = H'(\cdot)$, where H and H' are chosen in the same way as in G_2 and G_3. Then note that, $\Pr[\mathsf{G}_2 = 1] = \Pr[1 \leftarrow A^{H,F_0}]$ and $\Pr[\mathsf{G}_3 = 1] = \Pr[1 \leftarrow A^{H,F_1}]$. Since the uniform secret s is chosen independently from A^{H,F_i}'s view, we use Lemma 3 to obtain

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \frac{2q_H}{\sqrt{|\mathcal{M}|}}$$

Game G_4: In game G_4, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle returns $\perp$. It is not hard to see that the games G_3 and G_4 are equivalent

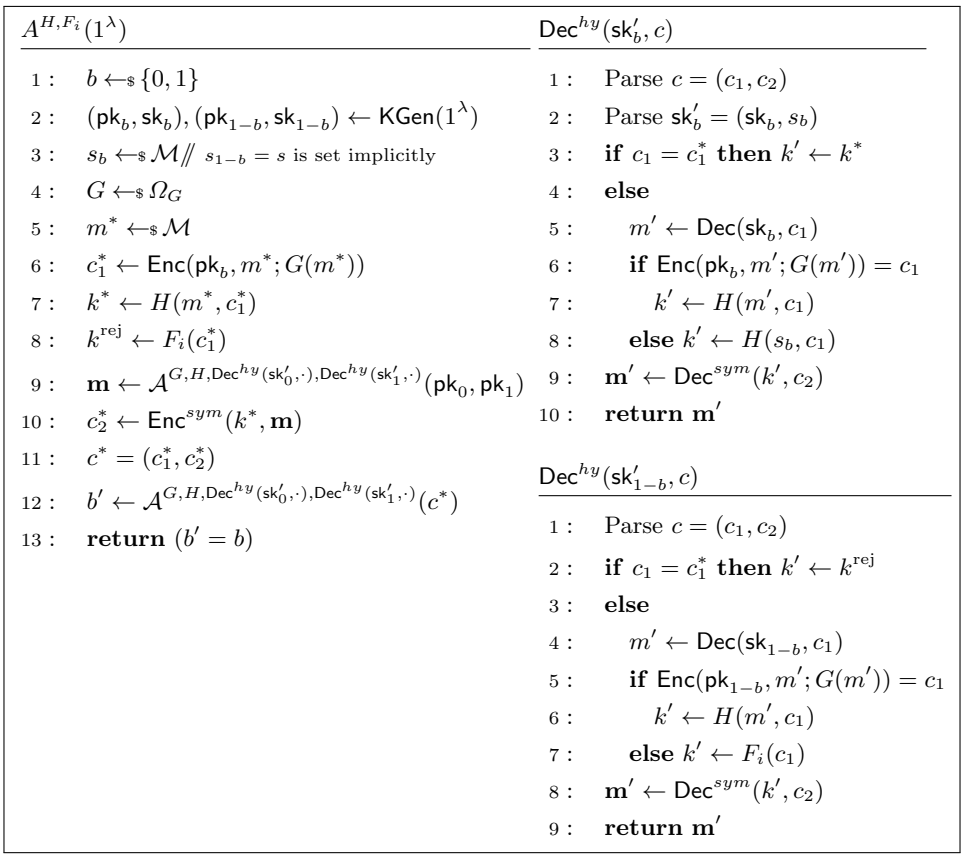


Fig. 19. Algorithm A^{H,F_i} for the proof of Theorem 7.

unless the following event occurs: $\mathcal{A}$ makes a decryption query (c_1^*, c_2) to the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, \cdot)$ such that $\mathsf{Dec}^{sym}(k^{\mathrm{rej}}, c_2) \neq \perp$. And we can bound the probability of this event occurring by the advantage of an adversary $\mathcal{E}$ in the INT-CTXT game of DEM (see Figure 20).

First note that, the internal random oracle H' is only used to process classical queries c_1 (because we consider only *classical* decryption queries in the QROM). Hence $\mathcal{E}$ simulates H' *classically* towards $\mathcal{A}$, e.g., via "lazy sampling" (and uses a $2q_G$-wise and a $2q_H$-wise independent function to simulate the quantum random oracles G and H respectively). Also note that in games G_3 and G_4, H' is never queried on c_1^* (particularly, in the $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, \cdot)$ oracle) except for defining $k^{\mathrm{rej}} (\leftarrow H'(c_1^*))$ in the setup. This is equivalent to having k^{rej} to be a uniformly random key independent of the oracle H', i.e., $k^{\mathrm{rej}} \leftarrow_{\$} \mathcal{K}$.

Now in the INT-CTXT game, we are implicitly defining k^{rej} to be the random secret key chosen by the challenger. The adversary $\mathcal{E}$ proceeds as described in Figure 20. Note that if the aforementioned event occurs, then $\mathcal{E}$ wins its corresponding game (also note that, $\mathcal{E}$ makes no encryption queries to the one-time AE-secure DEM w.r.t. the secret key k^{rej}). Hence, we have

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E})$$

$\mathcal{E}^{\mathsf{Enc}^{sym}(k^{\mathrm{rej}}, \cdot), \mathsf{Dec}^{sym}(k^{\mathrm{rej}}, \cdot)}(1^\lambda)$

$b \leftarrow_{\$} \{0,1\}$
$(\mathsf{pk}_b, \mathsf{sk}_b), (\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}(1^\lambda)$
$s_b \leftarrow_{\$} \mathcal{M}$
$G \leftarrow_{\$} \Omega_G; H \leftarrow_{\$} \Omega_H; H' \leftarrow_{\$} \Omega_{H'}$
$m^* \leftarrow_{\$} \mathcal{M}$
$c_1^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; G(m^*))$
$k^* \leftarrow H(m^*, c_1^*)$
$\mathbf{m} \leftarrow \mathcal{A}^{G, H, \mathsf{Dec}^{hy}(\mathsf{sk}'_0, \cdot), \mathsf{Dec}^{hy}(\mathsf{sk}'_1, \cdot)}(\mathsf{pk}_0, \mathsf{pk}_1)$
$c_2^* \leftarrow \mathsf{Enc}^{sym}(k^*, \mathbf{m})$
$c^* = (c_1^*, c_2^*)$
$b' \leftarrow \mathcal{A}^{G, H, \mathsf{Dec}^{hy}(\mathsf{sk}'_0, \cdot), \mathsf{Dec}^{hy}(\mathsf{sk}'_1, \cdot)}(c^*)$
return $\perp$

$\mathsf{Dec}^{hy}(\mathsf{sk}'_b, c)$

Parse $c = (c_1, c_2)$
Parse $\mathsf{sk}'_b = (\mathsf{sk}_b, s_b)$
if $c_1 = c_1^*$ **then** $k' \leftarrow k^*$
else
  $m' \leftarrow \mathsf{Dec}(\mathsf{sk}_b, c_1)$
  if $\mathsf{Enc}(\mathsf{pk}_b, m'; G(m')) = c_1$
    $k' \leftarrow H(m', c_1)$
  else $k' \leftarrow H(s_b, c_1)$
$\mathbf{m}' \leftarrow \mathsf{Dec}^{sym}(k', c_2)$
return $\mathbf{m}'$

$\mathsf{Dec}^{hy}(\mathsf{sk}'_{1-b}, c)$

Parse $c = (c_1, c_2)$
if $c_1 = c_1^*$
  query $\mathsf{Dec}^{sym}(k^{\mathrm{rej}}, c_2)$
  return $\perp$
else
  $m' \leftarrow \mathsf{Dec}(\mathsf{sk}_{1-b}, c_1)$
  if $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G(m')) = c_1$
    $k' \leftarrow H(m', c_1)$
  else $k' \leftarrow H'(c_1)$
$\mathbf{m}' \leftarrow \mathsf{Dec}^{sym}(k', c_2)$
return $\mathbf{m}'$

Fig. 20. INT-CTXT adversary $\mathcal{E}^{\mathsf{Enc}(\hat{k}, \cdot), \mathsf{Dec}(\hat{k}, \cdot)}$ for the proof of Theorem 1.

Game G_5 In game G_5, we (re-)modify the decryption oracle $\mathsf{Dec}^{hy}(\mathsf{sk}'_{1-b}, \cdot)$ such that the key $H(s_{1-b}, c_1)$ is used to decrypt the DEM ciphertext c_2 instead of $H'(c_1)$ where the KEM ciphertext c_1 was implicitly rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation. In a sense, we are reverting the changes introduced in the $\mathsf{G}_2 \rightarrow \mathsf{G}_3$ hop. Hence, by using a similar analysis as that hop (and note that now, the key k^{rej} is not used anymore),

it is not hard to obtain

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq \frac{2q_H}{\sqrt{|\mathcal{M}|}}$$

Compared to the proof of Theorem 1, we have effectively used the sequence of games $\mathsf{G}_0 - \mathsf{G}_5$ to arrive at a point where we modified the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}'_{1-b}, \cdot)$ such that if the decryption query is (c_1^*, c_2), the oracle returns $\perp$; this particular point is the *hybrid* game "G_2" in the proof of Theorem 1. Now doing a similar sequence of game-hops from that point on, namely "$\mathsf{G}_2 \rightarrow \mathsf{G}_4$", in the current setting starting from G_5, we arrive at

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{PKE}^{hy}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}^{\not\perp}}(\mathcal{C}) + \mathbf{Adv}^{\text{WCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) \\ + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \frac{4q_H}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2^{-\gamma}\end{aligned}$$

C.8 Proof of Theorem 8

Theorem 8. *For any* ANO-CPA *adversary* $\mathcal{A}$ *against* Saber.PKE*, there exists a distinguisher* $\mathcal{B}_1$ *(resp.,* $\mathcal{B}_2$*) between* l *(resp.* $l+1$*) samples from a mod-LWR distribution from that of a uniform distribution, with corresponding parameters* l, μ, q *and* p*, such that*

$$\mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \mathbf{Adv}^{\text{mod-lwr}}_{l+1,l,\mu,q,p}(\mathcal{B}_2).$$

Moreover, the running times of $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *are the same as that of* $\mathcal{A}$*.*

Games $\mathsf{G}_0 - \mathsf{G}_4$

$(\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow\!\!\$\ \mathsf{KGen} \mathbin{/\!\!/} \mathsf{pk}_0 = (A_0, b_0)$
$(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow\!\!\$\ \mathsf{KGen} \mathbin{/\!\!/} \mathsf{pk}_1 = (A_1, b_1)$
$m \leftarrow \mathcal{A}(\mathsf{pk}_0, \mathsf{pk}_1)$
$s' \leftarrow\!\!\$\ \beta_\mu(R_q^{l\times 1})$
$b' = ((A_0 s' + h) \bmod q) >> (\epsilon_q - \epsilon_p) \in R_p^{l\times 1} \mathbin{/\!\!/} \mathsf{G}_0 - \mathsf{G}_1$
$b' \leftarrow\!\!\$\ R_p^{l\times 1} \mathbin{/\!\!/} \mathsf{G}_2$
$b' = ((A_1 s' + h) \bmod q) >> (\epsilon_q - \epsilon_p) \in R_p^{l\times 1} \mathbin{/\!\!/} \mathsf{G}_3 - \mathsf{G}_4$
$v' = b_0^T (s' \bmod p) \in R_p \mathbin{/\!\!/} \mathsf{G}_0$
$v' \leftarrow\!\!\$\ R_p \mathbin{/\!\!/} \mathsf{G}_1 - \mathsf{G}_3$
$v' = b_1^T (s' \bmod p) \in R_p \mathbin{/\!\!/} \mathsf{G}_4$
$c_m = (v' + h_1 - 2^{\epsilon_p - 1} m \bmod p) >> (\epsilon_p - \epsilon_T) \in R_T$
$b \leftarrow \mathcal{A}(c_m, b')$

Fig. 21. Games $\mathsf{G}_0 - \mathsf{G}_4$ for the proof of Theorem 8.

Proof. **Game** G_0 In game G_0, the adversary $\mathcal{A}$ always receives the encryption of its chosen message m under the public-key pk_0.

Game G_1 In game g_1, we replace v' with a uniformly random value in R_p. As explained in [15], on a higher level, Saber.PKE can be seen as a variant of ElGamal public-key encryption. Then in [15], the "Diffie-Hellman key-exchange" counterpart of Saber.PKE – namely $\mathsf{pSaber.KE}'$ – was shown to satisfy the so-called *key-indistinguishability* property based on the hardness of the module learning-with-rounding problem (mod-LWR). This effectively means that the distribution (pk_0, b', v') as generated in G_0 is computationally indistinguishable from (pk_0, b', v'') where $v'' \leftarrow\!\!\$\ R_p$. More concretely, it was shown in [15, Theorem 3] that,

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \mathbf{Adv}^{\text{mod-lwr}}_{l+1,l,\mu,q,p}(\mathcal{B}_2)$$

where $\mathbf{Adv}^{\text{mod-lwr}}_{m,l,\mu,q,p}(\mathcal{B})$ denotes the advantage of an adversary $\mathcal{B}$ in distinguishing between m samples from a mod-LWR distribution from that of a uniform distribution, with corresponding parameters l, μ, q and p.

Game G_2 In game G_2, we replace b' with a uniformly random vector in $R_p^{l\times 1}$. Since (A_0, b') forms a mod-LWR sample (recall that $A_0 \leftarrow_\$ R_q^{l\times l}$) in G_2, it is computationally indistinguishable from (A_0, b'') for $b'' \leftarrow_\$ R_p^{l\times 1}$ based on the hardness of the mod-LWR problem. More concretely, we have

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1)\,.$$

Game G_3 In game G_3, we (re-)compute b' as $b' = ((A_1 s' + h) \bmod q) >> (\epsilon_q - \epsilon_p) \in R_p^{l\times 1}$. Since this game-hop is, in a sense, the inverse of the $(\mathsf{G}_1 \to \mathsf{G}_2)$ hop but w.r.t. a different uniformly random matrix $A_1 (\leftarrow_\$ R_q^{l\times l})$, it is easy to see that

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1)\,.$$

Game G_4 In game G_4, we recompute v' as $v' = b_1^T(s' \bmod p) \in R_p$. Since this game-hop is, in a sense, the inverse of the $(\mathsf{G}_0 \to \mathsf{G}_1)$ hop but w.r.t. a different public-key $\mathsf{pk}_1 (= (A_1, b_1))$, it is again easy to see that

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \mathbf{Adv}^{\text{mod-lwr}}_{l+1,l,\mu,q,p}(\mathcal{B}_2).$$

Now note that in G_4, the adversary $\mathcal{A}$ receives the encryption of its chosen message m under the public-key pk_1. Therefore, we have

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{A}) &= \frac{1}{2}\cdot|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_0 = 1]| \\ &\leq 2\cdot\mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \mathbf{Adv}^{\text{mod-lwr}}_{l+1,l,\mu,q,p}(\mathcal{B}_2)\,.\end{aligned}$$

C.9 Proof of Theorem 9

Theorem 9. *Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_D *classical queries to the decapsulation oracles, at most* q_G *(resp.* q_H*) quantum queries to the random oracle* G *(resp.* H*), there exist* ANO-CPA *adversary* $\mathcal{B}$*,* OW-CPA *adversary* $\mathcal{C}$ *against* $\mathsf{Saber.PKE}$ *and a distinguisher* $\mathcal{B}_1$ *between* l *samples from a mod-LWR distribution and a uniform distribution with corresponding parameters* l, μ, q *and* p*, such that*

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) &\leq \mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{B}) + 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{C})} \\ &+ \mathbf{Coll}^F_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{2}{2^{256}} + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta}\end{aligned}$$

Here α (< 648) *is the constant from Lemma 1. The running times of* $\mathcal{B}$ *and* $\mathcal{C}$ *are the same as that of* $\mathcal{A}$*. The running time of* $\mathcal{B}_1$ *is independent (and less than that) of the running time of* $\mathcal{A}$*.*

Proof. The structure of the proof is quite similar to that of Theorem 5.

Denote Ω_{G_2}, Ω_G, Ω_H and $\Omega_{H'}$ to be the set of all functions $G_2 : \{0,1\}^* \to \{0,1\}^{512}$, $G : \{0,1\}^{256} \to \{0,1\}^{256}$, $H : \{0,1\}^* \to \{0,1\}^{256}$ and $H' : \overline{\mathcal{C}} \to \{0,1\}^{256}$ respectively, where $\overline{\mathcal{C}}$ is the ciphertext space of $\mathsf{Saber.PKE}/\mathsf{pSaber.KEM}$.

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for $\mathsf{pSaber.KEM}$ issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G (resp., q_H) quantum queries to the random oracles G (resp. H). Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_{12}$ described in Figures 22 and 23.

Game G_0 The game G_0 is exactly the ANO-CCA game for $\mathsf{pSaber.KEM}$. Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}).$$

Game G_1 In game G_1, we modify the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ (resp. $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$) such that $H_0^{\text{rej}}(c)$ (resp. $H_1^{\text{rej}}(c)$) is returned instead of $H(s_0, c)$ (resp. $H(s_1, c)$) for an invalid ciphertext c. Since this

Games $\mathsf{G}_0 - \mathsf{G}_{8.5}$

$(\mathsf{pk}_0, \mathsf{sk}_0'), (\mathsf{pk}_1, \mathsf{sk}_1') \leftarrow \mathsf{KGen}'$
$G_2 \leftarrow_\$ \Omega_{G_2}; G_{0r}, G_{1r} \leftarrow_\$ \Omega_G$
$G_{0r}^{\text{good}} \leftarrow_\$ \Omega_{G_0^{\text{good}}}; G_{0r} = G_{0r}^{\text{good}} /\!\!/ \ \mathsf{G}_{6.5}$ - $\mathsf{G}_{8.5}$
$G_{1r}^{\text{good}} \leftarrow_\$ \Omega_{G_1^{\text{good}}}; G_{1r} = G_{1r}^{\text{good}} /\!\!/ \ \mathsf{G}_7$ - $\mathsf{G}_{8.5}$
$G_{0\hat{k}}, G_{1\hat{k}} \leftarrow_\$ \Omega_G /\!\!/ \ \mathsf{G}_0$ - G_8
$G_{0\hat{k}}, G_{1\hat{k}} \leftarrow_\$ \Omega_{\text{poly}} /\!\!/ \ \mathsf{G}_{8.5}$
$H_2 \leftarrow_\$ \Omega_H; H_0^{\text{rej}}, H_1^{\text{rej}} \leftarrow_\$ \Omega_{H'}$
$H_3 \leftarrow_\$ \Omega_G; H_0^{\text{acc}}, H_1^{\text{acc}} \leftarrow_\$ \Omega_{H'}$
$b \leftarrow_\$ \{0,1\}$
$m^* \leftarrow_\$ \{0,1\}^{256}$
$(\hat{k}^*, r^*) \leftarrow G(F(pk_b), m^*) /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
$r^* \leftarrow G_{br}(m^*) /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_{8.5}$
$\hat{k}^* \leftarrow G_{b\hat{k}}(m^*) /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_7$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$k^* \leftarrow H(\hat{k}^*, c^*) /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_7$
$k^* \leftarrow H_3(m^*) /\!\!/ \ \mathsf{G}_8 - \mathsf{G}_{8.5}$
$inp \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
$b' \leftarrow \mathcal{A}^{G, H, \mathsf{Decap}(\mathsf{sk}_0', \cdot), \mathsf{Decap}(\mathsf{sk}_1', \cdot)}(inp)$
return $(b' = b)$

$G(f, m)$ $/\!\!/$ $|f| + |m| = 512$

if $f = F(\mathsf{pk}_0)$ **then** $/\!\!/$ G_2-$\mathsf{G}_{8.5}$
  $r \leftarrow G_{0r}(m) /\!\!/ \ \mathsf{G}_2$ - $\mathsf{G}_{8.5}$
  $\hat{k} \leftarrow G_{0\hat{k}}(m) /\!\!/ \ \mathsf{G}_2$ - $\mathsf{G}_{8.5}$
elseif $f = F(\mathsf{pk}_1)$ **then** $/\!\!/$ G_2-$\mathsf{G}_{8.5}$
  $r \leftarrow G_{1r}(m) /\!\!/ \ \mathsf{G}_2$ - $\mathsf{G}_{8.5}$
  $\hat{k} \leftarrow G_{1\hat{k}}(m) /\!\!/ \ \mathsf{G}_2$ - $\mathsf{G}_{8.5}$
else $(\hat{k}, r) \leftarrow G_2(f, m)$
return $(\hat{k}, r)$

$G(f, m)$ $/\!\!/$ $|f| + |m| \neq 512$

return $G_2(f, m)$

$H(\hat{k}, c)$ $/\!\!/$ $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$

$m' = \mathsf{Dec}(\mathsf{sk}_0, c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c \wedge$
  $G_{0\hat{k}}(m') = \hat{k} /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
  if $c = c^* /\!\!/ \ \mathsf{G}_6 - \mathsf{G}_{8.5}$
    return $H_3(m') /\!\!/ \ \mathsf{G}_6 - \mathsf{G}_{8.5}$
  return $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c \wedge$
  $G_{1\hat{k}}(m') = \hat{k} /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
  if $c = c^* /\!\!/ \ \mathsf{G}_6 - \mathsf{G}_{8.5}$
    return $H_3(m') /\!\!/ \ \mathsf{G}_6 - \mathsf{G}_{8.5}$
  return $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_{8.5}$
return $H_2(\hat{k}, c)$

$H(\hat{k}, c)$ $/\!\!/$ $\hat{k} \notin \{0,1\}^{256}$ or $c \notin \overline{\mathcal{C}}$

return $H_2(\hat{k}, c)$

$\mathsf{Decap}(\mathsf{sk}_0', c)$

return $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_{4.5}$ - $\mathsf{G}_{8.5}$
Parse $\mathsf{sk}_0' = (\mathsf{sk}_0, s_0, F(\mathsf{pk}_0))$
$m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_0), m') /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_{0r}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_4$
$\hat{k}' \leftarrow G_{0\hat{k}}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_4$
if $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$ **then**
  return $H(\hat{k}', c)$
else return $H(s_0, c) /\!\!/ \ \mathsf{G}_0$
else return $H_0^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1$ - G_4

$\mathsf{Decap}(\mathsf{sk}_1', c)$

return $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_5$ - $\mathsf{G}_{8.5}$
Parse $\mathsf{sk}_1' = (\mathsf{sk}_1, s_1, F(\mathsf{pk}_1))$
$m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_1), m') /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_{1r}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_{4.5}$
$\hat{k}' \leftarrow G_{1\hat{k}}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_{4.5}$
if $\mathsf{Enc}(\mathsf{pk}_1, m'; r') = c$ **then**
  return $H(\hat{k}', c)$
else return $H(s_1, c) /\!\!/ \ \mathsf{G}_0$
else return $H_1^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1$ - $\mathsf{G}_{4.5}$

Fig. 22. Games G_0 – $\mathsf{G}_{8.5}$ for the proof of Theorem 9.

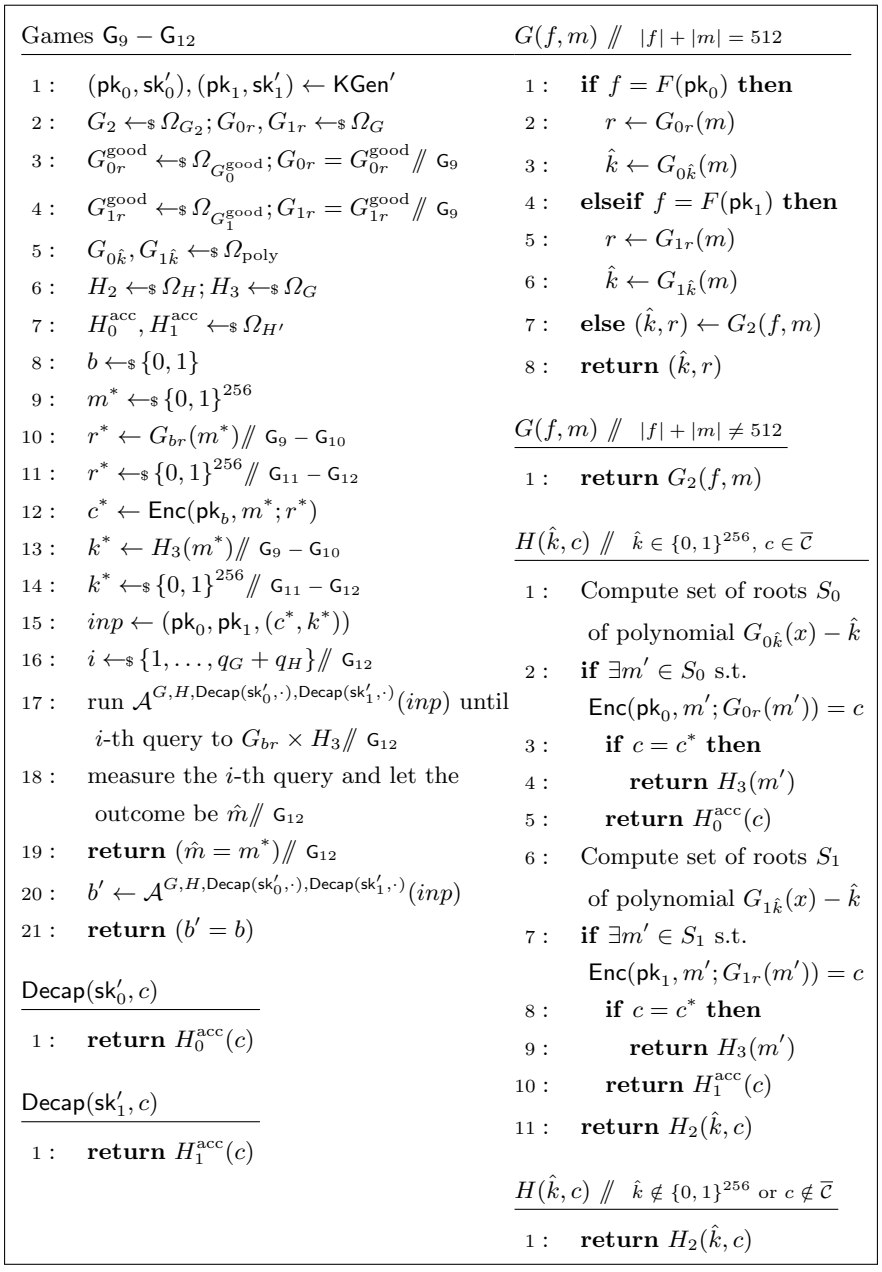


Fig. 23. Games G_9 – G_{12} for the proof of Theorem 9.

change is similar to the sequence of game-hops "$\mathsf{G}_0 \rightarrow \mathsf{G}_{0.5} \rightarrow \mathsf{G}_1$" in the proof of Theorem 5, using Lemma 3, it is not hard to obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{4q_H}{\sqrt{2^{256}}}$$

(note that the message space of Saber.PKE is $\{0,1\}^{256}$).

Game G_2 In game G_2, we implicitly divide the G-queries into at most three categories: (1) query is of the form (f, m) with $|f|+|m| = 512$ and $f = F(\mathsf{pk}_0)$, (2) query is of the form (f, m) with $|f|+|m| = 512$ and $f = F(\mathsf{pk}_1)$, and (3) the remaining queries. We then respond to the queries from the respective categories with $(G_{0\hat{k}}(m), G_{0r}(m))$, $(G_{1\hat{k}}(m), G_{1r}(m))$ and $G_2(m, c)$ respectively, where $G_{i\hat{k}}$, G_{ir} (for $i \in \{0,1\}$) are internal random functions; note that we say "at most" three categories because of the (unlikely) possibility that $F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$. It is not hard to verify that the output distributions of the G-oracle in games G_1 and G_2 are equivalent. Therefore,

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_1 = 1].$$

Game G_3 In game G_3, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\hat{k}^*, r^*$ in setup of the game as "$\hat{k}^* \leftarrow G_{b\hat{k}}(m^*)$" and "$r^* \leftarrow G_{br}(m^*)$" (effectively, replacing the step "$(\hat{k}^*, r^*) \leftarrow G(F(\mathsf{pk}_b), m^*)$" in G_2). We then similarly generate the values $\hat{k}', r'$ w.r.t. the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ ($i \in \{0,1\}$) as "$\hat{k}' \leftarrow G_{i\hat{k}}(m')$" and "$r' \leftarrow G_{ir}(m')$" (replacing the step "$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_i), m')$" in G_2).

Let bad denote the event where the public-keys pk_0 and pk_1 generated honestly in the setup satisfy the following: $F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$. It is not hard to see that the games G_2 and G_3 are equivalent unless the event "bad" happens. Hence, if $\mathbf{Coll}^F_{\mathsf{Saber.PKE}}$ is defined as the probability of the event "$F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$" where pk_0 and pk_1 are two honestly-generated Saber.PKE public-keys, then we have

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \Pr[\mathsf{bad}] \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}}.$$

Game G_4 In game G_4, we implicitly divide the H-queries into three disjoint categories: (1) query is of the form $(\hat{k}, c)$ with $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$, and which satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ and $G_{0\hat{k}}(m) = \hat{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$, (2) query is of the form $(\hat{k}, c)$ with $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$ and which does not fall under "category (1)", while at the same time, satisfies $\mathsf{Enc}(\mathsf{pk}_1, m; G_{1r}(m)) = c$ and $G_{1\hat{k}}(m) = \hat{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_1, c)$, and (3) the remaining queries. We then respond to the queries from the respective categories with $H_0^{\mathrm{acc}}(c)$, $H_1^{\mathrm{acc}}(c)$ and $H_2(\hat{k}, c)$, where H_0^{acc} and H_1^{acc} are internal random functions not directly accessible to the adversary $\mathcal{A}$.

Focusing on H-queries in "category (1)", note that it is not possible for two distinct queries $(\hat{k}', c)$ and $(\hat{k}'', c)$ to result in the same output $H_0^{\mathrm{acc}}(c)$. Note that $\mathsf{Dec}(\mathsf{sk}_0, \cdot)$ and $G_{0\hat{k}}(\cdot)$ are deterministic functions. Hence w.r.t. the queries $(\hat{k}', c)$ and $(\hat{k}'', c)$, there can only exist a unique value m such that $m = \mathsf{Dec}(\mathsf{sk}_0, c)$. At the same time, $G_{0\hat{k}}(m)$ can take at most one value. The same reasoning applies to "category (2)" as well, and hence, the output distributions of the H-oracle in the games G_3 and G_4 are equivalent. Therefore,

$$\Pr[\mathsf{G}_4 = 1] = \Pr[\mathsf{G}_3 = 1].$$

Game $\mathsf{G}_{4.5}$ In game $\mathsf{G}_{4.5}$, we change the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle such that there is no need for the secret key sk'_0. Namely, $H_0^{\mathrm{acc}}(c)$ is returned for the decapsulation of ciphertext c w.r.t. sk'_0. Let $m' = \mathsf{Dec}(\mathsf{sk}_0, c)$, $r' = G_{0r}(m')$ and $\hat{k}' = G_{0\hat{k}}(m')$. Now consider the following two cases:

1. $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$. In this case, the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle returns $H(\hat{k}', c)$ in game G_4 and $H_0^{\mathrm{acc}}(c)$ in game $\mathsf{G}_{4.5}$. Hence, it is not hard to see that we have $H(\hat{k}', c) = H_0^{\mathrm{acc}}(c)$ in G_4, since the query $(\hat{k}', c)$ falls under "category (1)" w.r.t. oracle H. Therefore, $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracles of games G_4 and $\mathsf{G}_{4.5}$ return the same value $H_0^{\mathrm{acc}}(c)$.
2. $\mathsf{Enc}(\mathsf{pk}_0, m'; r') \neq c$. In this case, the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle returns $H_0^{\mathrm{rej}}(c)$ in game G_4 and $H_0^{\mathrm{acc}}(c)$ in game $\mathsf{G}_{4.5}$. In game G_4, as the random function H_0^{rej} is independent of all other oracles, the output $H_0^{\mathrm{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game $\mathsf{G}_{4.5}$, the only way $\mathcal{A}$ gets prior access to the function H_0^{acc} is if it made a H-query $(\hat{k}'', c)$ such that $\mathsf{Enc}(\mathsf{pk}_0, m''; G_{0r}(m'')) = c$ (and $G_{0\hat{k}}(m'') = \hat{k}''$), where $m'' = \mathsf{Dec}(\mathsf{sk}_0, c)$. But since $\mathsf{Dec}(\mathsf{sk}_0, \cdot)$ is a deterministic function, we have $m'' = m'$ leading to

a contradiction of "$\mathsf{Enc}(\mathsf{pk}_0, m'; r') \neq c$". Therefore, such a prior access is not possible and $H_0^{\text{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle in G_4 and $\mathsf{G}_{4.5}$ are the same in both cases, we have

$$\Pr[\mathsf{G}_{4.5} = 1] = \Pr[\mathsf{G}_4 = 1].$$

Game G_5 In game G_5, we change the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle such that $H_1^{\text{acc}}(c)$ is returned for the decapsulation of *any* ciphertext c w.r.t. sk_1. The analysis here follows quite similarly to that of the previous game-hop except that this simulation of the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle – without the secret key sk'_1 – will fail (w.r.t. case 1 in the above game-hop) if $\mathcal{A}$ asks for the decapsulation of a ciphertext $\hat{c}$ such that $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = \hat{c} = \mathsf{Enc}(\mathsf{pk}_0, m''; G_{0r}(m''))$ and $G_{1\hat{k}}(m') = \hat{k'} = G_{0\hat{k}}(m'')$, where $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$ and $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$. In this peculiar case, $H_0^{\text{acc}}(\hat{c})$ is returned in $\mathrm{G}_{4.5}$ and $H_1^{\text{acc}}(\hat{c})$ is returned in G_5.

We bound the probability of this peculiar event (i.e., $\mathcal{A}$ asking for the decapsulation of such an above ciphertext $\hat{c}$ w.r.t. sk'_1) by the advantage of an adversary $\mathcal{E}$ against the *claw-finding* problem w.r.t. the instance $(G_{0\hat{k}}, G_{1\hat{k}})$. Because note that the pair (m'', m) is a *claw* with $G_{0\hat{k}}(m'') = G_{1\hat{k}}(m')$, where $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$ and $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$. More formally, $\mathcal{E}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game $\mathsf{G}_{4.5}$, by creating the appropriate setup (starting with the generation of two honest key-pairs $(\mathsf{pk}_0, \mathsf{sk}'_0)$ and $(\mathsf{pk}_1, \mathsf{sk}'_1)$).
- Uses three different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r} and G_{1r} respectively, four different $2q_H$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc}, H_1^{rej} and H_2 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also uses the pair of functions $f_0 : \{0,1\}^{256} \to \{0,1\}^{256}$ and $f_1 : \{0,1\}^{256} \to \{0,1\}^{256}$ – which is the instance of the claw-finding problem – to simulate the oracles $G_{0\hat{k}}$ and $G_{1\hat{k}}$ respectively.
- Answers decapsulation queries the same way as in $\mathsf{G}_{4.5}$. Particularly, w.r.t. any query $\hat{c}$ made by $\mathcal{A}$ to the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle, checks if the query satisfies the above described peculiar event. If so, returns the pair (m'', m') as a claw w.r.t. (f_0, f_1), where $m'' = \mathsf{Dec}(\mathsf{sk}_0, \hat{c})$ and $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c})$.

Note that $\mathcal{E}$ makes at most q_G queries to the pair (f_0, f_1). Let $\Pr[\mathcal{P}]$ be the probability of this peculiar event, denoted as $\mathcal{P}$, occurring. We have the games $\mathsf{G}_{4.5}$ and G_5 to be equivalent unless the event $\mathcal{P}$ occurs. From the construction of the claw-finding adversary $\mathcal{E}$ above, it is not hard to see that $\Pr[\mathcal{P}] \leq \frac{\alpha(q_G+1)^3}{2^{256}}$ from Lemma 1. Hence, we have

$$|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_{4.5} = 1]| \leq \Pr[\mathcal{P}] \leq \frac{\alpha(q_G+1)^3}{2^{256}}.$$

Game G_6 In game G_6, we make a further modification to the evaluation of "category (1) and (2)" H-queries (as introduced in the "$\mathsf{G}_3 \to \mathsf{G}_4$" game-hop) of the form $(\hat{k}, c^*)$ as follows, where c^* is the challenge ciphertext computed in the setup: respond to the corresponding "category (1)" queries with $H_3(m)$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$, and the corresponding "category (2)" queries with $H_3(m)$, where $m = \mathsf{Dec}(\mathsf{sk}_1, c)$. Here H_3 is an internal independent random function.

Let $m_0 = \mathsf{Dec}(\mathsf{sk}_0, c^*)$ and $m_1 = \mathsf{Dec}(\mathsf{sk}_1, c^*)$ which additionally satisfy $\mathsf{Enc}(\mathsf{pk}_0, m_0; G_{0r}(m_0)) = c^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G_{1r}(m_1)) = c^*$. So to analyze this change to oracle H, there are only two H-queries worth considering, namely "category (1)" query $(\hat{k}_0, c^*)$ and "category (2)" query $(\hat{k}_1, c^*)$ where $\hat{k}_0 = G_{0\hat{k}}(m_0)$ and $\hat{k}_1 = G_{1\hat{k}}(m_1)$. W.r.t. these two queries, the H oracle would return $H_0^{\text{acc}}(c^*)$, $H_1^{\text{acc}}(c^*)$ respectively in G_5, and $H_3(m_0)$, $H_3(m_1)$ respectively in G_6. Conditional on $m_0 \neq m_1$, the adversary $\mathcal{A}$'s view would be identical even after this change because the random values $H_0^{\text{acc}}(c^*)$, $H_1^{\text{acc}}(c^*)$ are only accessible to $\mathcal{A}$ via the H-oracle in G_5, and in particular, not through the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles since c^* is a forbidden decapsulation query. Hence in G_6, we are effectively replacing two uniformly random values that can only be accessed via the H-oracle by $\mathcal{A}$ with two other uniformly random values. Hence, the output distributions of the H-oracle in the games G_5 and G_6 are equivalent unless we have $m_0 = m_1$, or in other words, the following event occurs w.r.t. two honest $\mathsf{Saber.PKE}$ key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1)$: $\mathsf{Dec}(\mathsf{sk}_0, c^*) = \mathsf{Dec}(\mathsf{sk}_1, c^*) = m'$ *and* $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c^*$, where for $m^* \leftarrow_{\$} \{0,1\}^{256}$ and $b \leftarrow_{\$} \{0,1\}$ we have $c^* = \mathsf{Enc}(\mathsf{pk}_b, m^*; G_{br}(m^*))$ (note that we are not assuming the correctness of $\mathsf{Saber.PKE}$, i.e., m^* may or may not be equal to m').

We can bound the probability of the above event by considering the *sub-event* "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G_{(1-b)r}(m')) = c^*$". Note that in the context of an experiment describing the above event, we have $G_{(1-b)r}(m')$ resulting in uniformly random coins $r' \leftarrow_\$ \{0,1\}^{256}$, since G_{br} is used to compute the ciphertext c^* and $G_{(1-b)r}$ is a random oracle independent to G_{br}. Borrowing the notation used to describe $\mathsf{Saber.PKE}$ (especially the KGen and Enc algorithms), note that the public-key pk_{1-b} results in a uniformly random matrix $A_{1-b} \leftarrow_\$ R_q^{l\times l}$ (in the random oracle model). Similarly the $\mathsf{Saber.PKE}$ ciphertext c^* (specifically, its second component) contains a vector $b^{*\prime} \in R_p^{l\times 1}$. Now considering the re-encryption check "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c^*$" and looking at the Enc algorithm description, note that this implies "$A_{1-b}s' + h \bmod q = b^{*\prime}$" where h is a constant vector, $A_{1-b} \leftarrow_\$ R_q^{l\times l}$ and $s' \leftarrow_\$ \beta_\mu(R_q^{l\times 1})$ (specifically, $s' = \beta_\mu(R_q^{l\times 1}; r)$ for uniformly random $r' \leftarrow_\$ \{0,1\}^{256}$ as discussed above). Since the distribution $(A_{1-b}, A_{1-b}s' + h \bmod q)$ is computationally indistinguishable from (A_{1-b}, b'') for a uniformly random vector $b'' \leftarrow_\$ R_p^{l\times 1}$ based on the hardness of mod-LWR, the probability of the event "$A_{1-b}s' + h \bmod q = b^{*\prime}$" will be negligible. More concretely, it is not hard to obtain the following:

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{1}{2^{256}}$$

where $\mathbf{Adv}^{\text{mod-lwr}}_{m,l,\mu,q,p}(\mathcal{B}_1)$ denotes the advantage of an adversary $\mathcal{B}_1$ in distinguishing between m samples from a mod-LWR distribution from that of a uniform distribution, with corresponding parameters l, μ, q and p. Also from the specification of pSaber, it can be shown that the size of the vector space $R_p^{l\times 1}$ is at least 2^{256}.

Game $\mathsf{G}_{6.5}$ In game $\mathsf{G}_{6.5}$, we change the random oracle G_{0r} such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_0, \mathsf{sk}_0)$, as seen in the proof of Theorem 5. Specifically, denote $\Omega_{G^{\text{good}}_{0r}}$ to be the set of all random functions G^{good}_{0r} such that $G^{\text{good}}_{0r}(m)$ is sampled according to a uniform distribution in $\mathcal{R}_{\text{good}}(\mathsf{pk}_0, \mathsf{sk}_0, m)$. Hence in $\mathsf{G}_{6.5}$, we replace the oracle G_{0r} with G^{good}_{0r}. By using a similar analysis as the game-hop $(\mathsf{G}_1 \rightarrow \mathsf{G}_2)$ in the proof of Theorem 5 (in fact, the analysis would be simpler in this case since we have to consider a single key-pair $(\mathsf{pk}_0, \mathsf{sk}_0)$ instead of two), it is not hard to obtain

$$|\Pr[\mathsf{G}_{6.5} = 1] - \Pr[\mathsf{G}_6 = 1]| \leq 2q_G\sqrt{\delta}$$

Game G_7 In game G_7, we now change the random oracle G_{1r} such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}_1, \mathsf{sk}_1)$. The analysis in this case would be similar (and simpler when compared) to the game-hop $(\mathsf{G}_1 \rightarrow \mathsf{G}_2)$ in the proof of Theorem 5. But a thing worth noting is that the distinguisher $B^{\hat{G}}$ (for $\hat{G} \in \{G_{1r}, G^{\text{good}}_{1r}\}$) – as was used in the $(\mathsf{G}_1 \rightarrow \mathsf{G}_2)$ game-hop in the proof of Theorem 5 – will have a single key-pair $(\mathsf{pk}_1, \mathsf{sk}_1)$ as input, and will need to simulate $\mathcal{A}$'s view in the games $\mathsf{G}_{6.5}$ and G_7. But since the distinguisher $B^{\hat{G}}$ can be *unbounded*, it can simulate the "non-ideal" random oracle G^{good}_{0r} that is used in $\mathsf{G}_{6.5}$ and G_7. Again, it is not hard to obtain

$$|\Pr[\mathsf{G}_7 = 1] - \Pr[\mathsf{G}_{6.5} = 1]| \leq 2q_G\sqrt{\delta}$$

Game G_8 In the setup of game G_8, we generate the value k^* as "$k^* \leftarrow H_3(m^*)$" (as opposed to "$k^* \leftarrow H(\hat{k}^*, c^*)$" in G_7). Also $\hat{k}^*$ is not generated in the setup (i.e., removing the step "$\hat{k}^* \leftarrow G_{b\hat{k}}(m^*)$" in G_7) as the value is not required anymore in the game. Note that G_8 is equivalent to G_7 w.r.t. this change unless the following event occurs: for $b = 1$ if we have $c^* = \mathsf{Enc}(\mathsf{pk}_1, m^*; G_{1r}(m^*))$ and $\hat{k}^* \leftarrow G_{1\hat{k}}(m^*)$ (for $m^* \leftarrow_\$ \{0,1\}^{256}$) in the setup, then $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c^*$ and $G_{0\hat{k}}(m') = \hat{k}^*$, where $\mathsf{Dec}(\mathsf{sk}_0, c^*) = m'$. Note that in this case, the value k^* computed in setup of the games will be equal to $H_3(m')(= H(\hat{k}^*, c^*))$ in G_7 and $H_3(m^*)$ in G_8.

We can bound the probability of such an event by considering the *sub-event* "$G_{0\hat{k}}(m') = G_{1\hat{k}}(m^*)\ (= \hat{k}^*)$". More formally, consider a (hypothetical) experiment which describes the above event as follows. First, it generates (honestly) two $\mathsf{Saber.PKE}$ key-pairs $(\mathsf{pk}_0, \mathsf{sk}'_0)$ and $(\mathsf{pk}_1, \mathsf{sk}'_1)$. Then it uniformly at random samples a message $m^* \leftarrow_\$ \{0,1\}^{256}$ and computes $c^* = \mathsf{Enc}(\mathsf{pk}_1, m^*; G_{1r}(m^*))$, $\hat{k}^* \leftarrow G_{1\hat{k}}(m^*)$; one thing worth noting here is that the hypothetical experiment can simulate the "non-ideal" random oracle $G_{1\hat{r}}$, which only samples "good" random coins, with an unbounded running time. Then it computes $m' = \mathsf{Dec}(\mathsf{sk}_0, c^*)$ and finally checks if "$G_{0\hat{k}}(m') = \hat{k}^*$". Note that in the context of this experiment, since this is the first invocation of the oracle $G_{0\hat{k}}$ (independent to $G_{1\hat{k}}$), $G_{0\hat{k}}(m')$ results in a uniformly random value $\hat{k}' \leftarrow_\$ \{0,1\}^{256}$. Therefore,

the probability of this sub-event, or "$G_{0\hat{k}}(m') = \hat{k}^*$", happening is at most $1/2^{256}$. Hence, it is not hard to see that

$$|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq \frac{1}{2^{256}}$$

Game $\mathsf{G}_{8.5}$ In game $\mathsf{G}_{8.5}$, we replace the random oracles $G_{i\hat{k}}$ ($i \in \{0,1\}$) with $2q_G$-wise independent functions, following Lemma 2. Random polynomials of degree $2q_G - 1$ over the finite field representation of the message space $\{0,1\}^{256}$ are $2q_G$-wise independent. Let Ω_{poly} be the set of all such polynomials. We are then replacing the step "$G_{0\hat{k}}, G_{1\hat{k}} \leftarrow_\$ \Omega_G$" with "$G_{0\hat{k}}, G_{1\hat{k}} \leftarrow_\$ \Omega_{\text{poly}}$" in $\mathsf{G}_{8.5}$. From Lemma 2, as this change is indistinguishable when the oracles $G_{0\hat{k}}, G_{1\hat{k}}$ are queried at most q_G times, we have

$$\Pr[\mathsf{G}_{8.5} = 1] = \Pr[\mathsf{G}_8 = 1]$$

Game G_9 In game G_9, we change the H-oracle such that there is no need for secret keys sk_0, sk_1. Namely, we implicitly divide the H-queries into three disjoint categories: (1) query is of the form $(\hat{k}, c)$ with $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$ and there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial $G_{0\hat{k}}(x) - \hat{k}$ (recall that $G_{0\hat{k}}$ and $G_{1\hat{k}}$ are now polynomials) such that $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$, (2) query is of the form $(\hat{k}, c)$ with $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$ and which do not fall under "category (1)", while at the same time, there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial $G_{1\hat{k}}(x) - \hat{k}$ such that $\mathsf{Enc}(\mathsf{pk}_1, m; G_{1r}(m)) = c$, and (3) the remaining queries. We then respond to queries from the respective categories as follows: (1) return $H_3(m)$ if $c = c^*$, otherwise return $H_0^{\text{acc}}(c)$, (2) return $H_3(m)$ if $c = c^*$, otherwise return $H_1^{\text{acc}}(c)$, and (3) return $H_2(\hat{k}, c)$.

It is not hard to see that the input-output behavior of oracle H in games $\mathsf{G}_{8.5}$ and G_9 is identical. For example, w.r.t. a query $(\hat{k}, c)$ if the oracle H in $\mathsf{G}_{8.5}$ returns $H_0^{\text{acc}}(c)$, then we have $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ ($\neq c^*$) and $G_{0\hat{k}}(m) = \hat{k}$, where $m = \mathsf{Dec}(\mathsf{sk}_0, c)$. This implies that m is the *only* root of the polynomial $G_{0\hat{k}}(x) - \hat{k}$ which satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ (note that there cannot exist some other root m' ($\neq m$) of $G_{0\hat{k}}(x) - \hat{k}$ satisfying $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c$ because, as G_{0r} samples "good" random coins, we must then have $\mathsf{Dec}(\mathsf{sk}_0, c) = m' = m$ – a contradiction), and hence on the same input $(\hat{k}, c)$, oracle H in G_9 outputs the value $H_0^{\text{acc}}(c)$ as well. In the other direction, w.r.t. a query $(\hat{k}, c)$ if the oracle H in G_9 returns $H_0^{\text{acc}}(c)$, then there exists a root m of the polynomial $G_{0\hat{k}}(x) - \hat{k}$ such that it *uniquely* satisfies $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$ ($\neq c^*$). Since G_{0r} samples "good" random coins, we must have $\mathsf{Dec}(\mathsf{sk}_0, c) = m$ with m satisfying $G_{0\hat{k}}(m) = \hat{k}$ and $\mathsf{Enc}(\mathsf{pk}_0, m; G_{0r}(m)) = c$. Therefore, on the same input $(\hat{k}, c)$, oracle H in $\mathsf{G}_{8.5}$ outputs the value $H_0^{\text{acc}}(c)$ as well. A similar reasoning applies to the outputs $H_1^{\text{acc}}(c)$ and $H_2(\hat{k}, c)$ w.r.t. H-queries $(\hat{k}, c)$, and also to queries of the form $(\hat{k}, c^*)$, which finally leads to the equivalence of oracles H in $\mathsf{G}_{8.5}$ and G_9. We thus have

$$\Pr[\mathsf{G}_9 = 1] = \Pr[\mathsf{G}_{8.5} = 1]$$

Game G_{10} In game G_{10}, we reset the random oracles G_{ir} (for $i \in \{0,1\}$) so that they return uniformly random coins from $\{0,1\}^{256}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_6 \to \mathsf{G}_7$, by using a similar analysis, we obtain

$$|\Pr[\mathsf{G}_{10} = 1] - \Pr[\mathsf{G}_9 = 1]| \leq 4q_G\sqrt{\delta}$$

Game G_{11} In the set-up of game G_{11}, we generate the values r^* and k^* such that they are uniformly random values independent of any oracles, i.e., we replace the step "$r^* \leftarrow G_{br}(m^*)$" with "$r^* \leftarrow_\$ \{0,1\}^{256}$" and "$k^* \leftarrow H_3(m^*)$" with "$k^* \leftarrow_\$ \{0,1\}^{256}$". We use Lemma 4 to bound the difference in the success probabilities of $\mathcal{A}$ in G_{10} and G_{11}. Let A be an oracle algorithm that has quantum access to the random oracle $G_r \times H_3$, where $G_r, H_3 \leftarrow_\$ \Omega_G$ and $(G_r \times H_3)(m) = (G_r(m), H_3(m))$. Figure 24 describes $A^{G_r \times H_3}$'s operation on input $(m^*, (r^*, k^*))$. Note that the algorithm $A^{G_r \times H_3}$ makes at most $q_G + q_H$ number of queries to the random oracle $G_r \times H_3$ to respond to $\mathcal{A}$'s G-oracle and H-oracle queries.

Let B be an oracle algorithm that on input m^* does the following: picks $i \leftarrow_\$ \{1, \ldots, q_G + q_H\}$, generates $r^* \leftarrow_\$ \{0,1\}^{256}$ and $k^* \leftarrow_\$ \{0,1\}^{256}$, runs the algorithm $A^{G_r \times H_3}(m^*, (r^*, k^*))$ until the i-th query, measures the argument of the $(G_r \times H_3)$-query in the computational basis and outputs the measurement outcome (if $A^{G_r \times H_3}$ makes less than i queries, B outputs $\perp$). With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_{10} = 1]$ and $P_A^2 = \Pr[\mathsf{G}_{11} = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 4 w.r.t. the algorithm $A^{G_r \times H_3}$. Therefore,

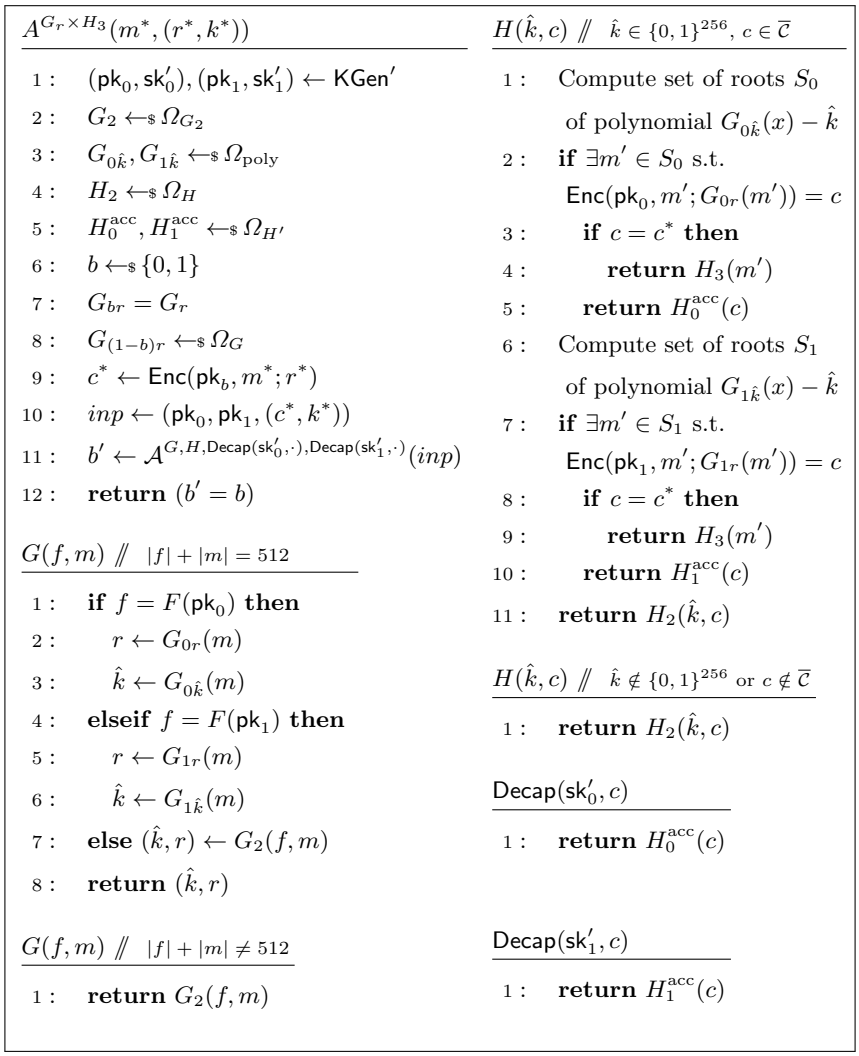


Fig. 24. Algorithm $A^{G_r \times H_3}$ for the proof of Theorem 9.

we now define game G_{12} (see Fig. 23) such that $P_B = \Pr[\mathsf{G}_{12} = 1]$, where P_B is as defined in Lemma 4 w.r.t. the algorithm $B^{G_r \times H_3}$. From Lemma 4, we thus have

$$|\Pr[\mathsf{G}_{10} = 1] - \Pr[\mathsf{G}_{11} = 1]| \leq 2(q_G + q_H)\sqrt{\Pr[\mathsf{G}_{12} = 1]}$$

We now bound the success probability of $\mathcal{A}$ in G_{11} by the advantage of an adversary $\mathcal{B}$ in the ANO-CPA game of $\mathsf{Saber.PKE}$. Upon receiving public-keys pk_0 and pk_1, $\mathcal{B}$ submits a uniformly random message $m^* \leftarrow_{\$} \{0,1\}^{256}$ to the ANO-CPA challenger. It then receives a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ for uniformly random bit $b(\leftarrow_{\$} \{0,1\})$ and randomness $r^*(\leftarrow_{\$} \{0,1\}^{256})$ chosen by the challenger. $\mathcal{B}$ then proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_{11}.
- Uses five different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r}, G_{1r}, $G_{0\hat{k}}$ and $G_{1\hat{k}}$ respectively, four different $2q_H$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc}, H_2 and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. The random oracles G and H are simulated in the same way as in G_{11}.
- Answers decapsulation queries using the oracles H_i^{acc} $(i \in \{0,1\})$ as in G_{11}.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow_{\$} \{0,1\}^{256}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$.
- After obtaining a bit b' from $\mathcal{A}$, forwards b' to its ANO-CPA challenger as the final message.

It is easy to see that $|\Pr[\mathsf{G}_{11} = 1] - \frac{1}{2}| = \mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{B})$. Now we bound the success probability of $\mathcal{A}$ in G_{12} by the advantage of an adversary $\mathcal{C}$ in the OW-CPA game of $\mathsf{Saber.PKE}$. Upon receiving a public-key pk along with a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^*(\leftarrow_{\$} \{0,1\}^{256})$ and randomness $r^*(\leftarrow_{\$} \{0,1\}^{256})$ chosen by the challenger, $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_{12} (e.g., starting with sampling a uniformly random bit $b \leftarrow_{\$} \{0,1\}$).
- Uses five different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r}, G_{1r}, $G_{0\hat{k}}$ and $G_{1\hat{k}}$ respectively, four different $2q_H$-wise independent functions to simulate the random oracles H_0^{acc}, H_1^{acc}, H_2 and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also evaluates $\mathcal{A}$'s G- and H-queries using the oracle $G_{br} \times H_3$; the random oracles G and H are simulated in the same way as in G_{12}.
- Answers decapsulation queries using the oracles H_i^{acc} $(i \in \{0,1\})$ as in G_{12}.
- For $\mathcal{A}$'s challenge query, first sets $\mathsf{pk}_b = \mathsf{pk}$. Then generates a key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}(1^\lambda)$, samples a uniformly random key $k^* \leftarrow_{\$} \{0,1\}^{256}$ and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$. (By doing this, note that we have $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ in $\mathcal{A}$'s view.)
- Selects $i \leftarrow_{\$} \{1, \ldots, q_G + q_H\}$, measures the i-th query to oracle $G_{br} \times H_3$ and returns the outcome $\hat{m}$.

Again, it is not hard to see that $\Pr[\mathsf{G}_{12} = 1] \leq \mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{C})$. Hence by collecting all of the above bounds, we arrive at

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) &\leq \mathbf{Adv}^{\text{ANO-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{B}) + 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{Saber.PKE}}(\mathcal{C})} \\ &+ \mathbf{Coll}^F_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{2}{2^{256}} + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta}\end{aligned}$$

C.10 Proof of Theorem 10

Theorem 10. *Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct, for any* SCFR-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_D *queries to the (classical) decapsulation oracles, at most* q_G *(resp.* q_H*) queries to the quantum random oracle* G *(resp.* H*), we have*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{\alpha(q_H+1)^3}{2^{256}} + \frac{4q_H}{2^{128}}$$

Here α (< 648) *is the constant from Lemma 1.*

Games $\mathsf{G}_0 - \mathsf{G}_5$

1 : $(\mathsf{pk}_0, \mathsf{sk}_0'), (\mathsf{pk}_1, \mathsf{sk}_1') \leftarrow \mathsf{KGen}'$
2 : $G_2 \leftarrow\!\!\$\ \Omega_{G_2}; G_{0r}, G_{1r} \leftarrow\!\!\$\ \Omega_G$
3 : $G_{0\hat{k}}, G_{1\hat{k}} \leftarrow\!\!\$\ \Omega_G$
4 : $H_2 \leftarrow\!\!\$\ \Omega_H; H_0^{\text{rej}}, H_1^{\text{rej}} \leftarrow\!\!\$\ \Omega_{H'}$
5 : $H_0^{\text{acc}}, H_1^{\text{acc}} \leftarrow\!\!\$\ \Omega_{H'}$
6 : $inp \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
7 : $c \leftarrow \mathcal{A}^{G, H, \mathsf{Decap}(\mathsf{sk}_0', \cdot), \mathsf{Decap}(\mathsf{sk}_1', \cdot)}(inp)$
8 : **return** $(\mathsf{Decap}(\mathsf{sk}_0', c) = \mathsf{Decap}(\mathsf{sk}_1', c))$

$\mathsf{Decap}(\mathsf{sk}_0', c)$

1 : **return** $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_{4.5}$ - G_5
2 : Parse $\mathsf{sk}_0' = (\mathsf{sk}_0, s_0, F(\mathsf{pk}_0))$
3 : $m' = \mathsf{Dec}(\mathsf{sk}_0, c)$
4 : $(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_0), m') /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
5 : $r' \leftarrow G_{0r}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_4$
6 : $\hat{k}' \leftarrow G_{0\hat{k}}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_4$
7 : **if** $\mathsf{Enc}(\mathsf{pk}_0, m'; r') = c$ **then**
8 : **return** $H(\hat{k}', c)$
9 : **else return** $H(s_0, c) /\!\!/ \ \mathsf{G}_0$
10 : **else return** $H_0^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1$ - G_4

$\mathsf{Decap}(\mathsf{sk}_1', c)$

1 : **return** $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_5$
2 : Parse $\mathsf{sk}_1' = (\mathsf{sk}_1, s_1, F(\mathsf{pk}_1))$
3 : $m' = \mathsf{Dec}(\mathsf{sk}_1, c)$
4 : $(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_1), m') /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
5 : $r' \leftarrow G_{1r}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_{4.5}$
6 : $\hat{k}' \leftarrow G_{1\hat{k}}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_{4.5}$
7 : **if** $\mathsf{Enc}(\mathsf{pk}_1, m'; r') = c$ **then**
8 : **return** $H(\hat{k}', c)$
9 : **else return** $H(s_1, c) /\!\!/ \ \mathsf{G}_0$
10 : **else return** $H_1^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1$ - $\mathsf{G}_{4.5}$

$G(f, m) /\!\!/ \ |f| + |m| = 512$

1 : **if** $f = F(\mathsf{pk}_0)$ **then** $/\!\!/ \ \mathsf{G}_2$-G_5
2 : $r \leftarrow G_{0r}(m) /\!\!/ \ \mathsf{G}_2$ - G_5
3 : $\hat{k} \leftarrow G_{0\hat{k}}(m) /\!\!/ \ \mathsf{G}_2$ - G_5
4 : **elseif** $f = F(\mathsf{pk}_1)$ **then** $/\!\!/ \ \mathsf{G}_2$-G_5
5 : $r \leftarrow G_{1r}(m) /\!\!/ \ \mathsf{G}_2$ - G_5
6 : $\hat{k} \leftarrow G_{1\hat{k}}(m) /\!\!/ \ \mathsf{G}_2$ - G_5
7 : **else** $(\hat{k}, r) \leftarrow G_2(f, m)$
8 : **return** $(\hat{k}, r)$

$G(f, m) /\!\!/ \ |f| + |m| \neq 512$

1 : **return** $G_2(f, m)$

$H(\hat{k}, c) /\!\!/ \ \hat{k} \in \{0,1\}^{256}, c \in \overline{\mathcal{C}}$

1 : $m' = \mathsf{Dec}(\mathsf{sk}_0, c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
2 : **if** $\mathsf{Enc}(\mathsf{pk}_0, m'; G_{0r}(m')) = c \wedge$
$G_{0\hat{k}}(m') = \hat{k} /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
3 : **return** $H_0^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
4 : $m' = \mathsf{Dec}(\mathsf{sk}_1, c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
5 : **if** $\mathsf{Enc}(\mathsf{pk}_1, m'; G_{1r}(m')) = c \wedge$
$G_{1\hat{k}}(m') = \hat{k} /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
6 : **return** $H_1^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_5$
7 : **return** $H_2(\hat{k}, c)$

$H(\hat{k}, c) /\!\!/ \ \hat{k} \notin \{0,1\}^{256}$ or $c \notin \overline{\mathcal{C}}$

1 : **return** $H_2(\hat{k}, c)$

Fig. 25. Games G_0 – G_5 for the proof of Theorem 10.

Proof. Denote Ω_{G_2}, Ω_G, Ω_H and $\Omega_{H'}$ to be the set of all functions $G_2 : \{0,1\}^* \to \{0,1\}^{512}$, $G : \{0,1\}^{256} \to \{0,1\}^{256}$, $H : \{0,1\}^* \to \{0,1\}^{256}$ and $H' : \overline{\mathcal{C}} \to \{0,1\}^{256}$ respectively, where $\overline{\mathcal{C}}$ is the ciphertext space of Saber.PKE/pSaber.KEM.

Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for pSaber.KEM issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G (resp., q_H) quantum queries to the random oracles G (resp. H).

The structure of the proof is very similar to that of Theorem 9. Basically we do a similar sequence of game-hops as in the proof of Theorem 9 until the point where we can simulate the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ ($i \in \{0,1\}$) without requiring the corresponding secret keys sk'_i.

To be specific, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_5$ as described in Figure 25. By a similar analysis as that of the proof of Theorem 9 w.r.t. these game-hops, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{4q_H}{2^{128}}$$

Note that the game G_0 is exactly the SCFR-CCA game for pSaber.KEM. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A})$$

Coming to the game G_5, note that the adversary $\mathcal{A}$ wins the game if it finally outputs a ciphertext c such that $\mathsf{Decap}(\mathsf{sk}'_0, c) = \mathsf{Decap}(\mathsf{sk}'_1, c)$. Because of the modification of the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles, this winning condition translates to $H^{\text{acc}}_0(c) = H^{\text{acc}}_1(c)$, where H^{acc}_0 and H^{acc}_1 are independent quantum-accessible random functions. Note that in this case, (c, c) is a *claw* w.r.t. the pair of functions $H^{\text{acc}}_0 : \overline{\mathcal{C}} \to \{0,1\}^{256}$ and $H^{\text{acc}}_1 : \overline{\mathcal{C}} \to \{0,1\}^{256}$. Hence we can bound the success probability of $\mathcal{A}$ in G_5 by the advantage of an adversary $\mathcal{D}$ against the *claw-finding* problem w.r.t. the instance (H^{acc}_0, H^{acc}_1). $\mathcal{D}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_5, by creating the appropriate setup (starting with the generation of two honest key-pairs $(\mathsf{pk}_0, \mathsf{sk}'_0)$ and $(\mathsf{pk}_1, \mathsf{sk}'_1)$).
- Uses five different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_{0r}, G_{1r}, $G_{0\hat{k}}$ and $G_{1\hat{k}}$ respectively, four different $2(q_G + q_H)$-wise independent functions to simulate the random oracles G_{0r}, G_{1r}, $G_{0\hat{k}}$ and $G_{1\hat{k}}$ respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also uses the pair of functions $f_0 : \overline{\mathcal{C}} \to \{0,1\}^{256}$ and $f_1 : \overline{\mathcal{C}} \to \{0,1\}^{256}$ – which is the instance of the claw-finding problem – to simulate the oracles H^{acc}_0 and H^{acc}_1 respectively.
- The random oracles G and H are simulated in the same way as in G_5 (e.g., note that H can be simulated as the claw-finding adversary $\mathcal{D}$ possesses the secret keys sk_0 and sk_1).
- Answers decapsulation queries using the oracles $f_i(\cdot)$ ($i \in \{0,1\}$) as in G_5.
- After obtaining a final ciphertext c from $\mathcal{A}$, forwards (c, c) as a claw w.r.t. (f_0, f_1).

Note that $\mathcal{D}$ makes at most q_H queries to the pair (f_0, f_1). It is easy to see that $\Pr[\mathsf{G}_5 = 1] \leq \frac{\alpha(q_H+1)^3}{2^{256}}$ from Lemma 1. Hence, we finally get

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{\alpha(q_H+1)^3}{2^{256}} + \frac{4q_H}{2^{128}}$$

C.11 Proof of Theorem 11

Theorem 11. *Let* $\mathsf{pSaber.PKE}^{hy} = (\mathsf{KGen}', \mathsf{Enc}^{hy}, \mathsf{Dec}^{hy})$ *be a hybrid encryption scheme obtained by composing* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *with a one-time authenticated encryption scheme* $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$. *Given* $\mathsf{Saber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, then for any* ANO-CCA *adversary $\mathcal{A}$ against* $\mathsf{pSaber.PKE}^{hy}$ *issuing at most q_G (resp. q_H) queries to the quantum random oracle G (resp. H), there exist* ANO-CCA *adversary $\mathcal{B}$,* IND-CCA *adversary $\mathcal{C}$ against* pSaber.KEM, INT-CTXT *adversary $\mathcal{E}$ against* DEM *and distinguisher $\mathcal{B}_1$ between l samples from a mod-LWR distribution and a uniform distribution, with corresponding parameters l, μ, q and p, such that*

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.PKE}^{hy}}(\mathcal{A}) \leq\ & \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{C}) + \mathbf{Coll}^F_{\mathsf{Saber.PKE}} \\ & + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta} + \frac{1}{2^{256}}\end{aligned}$$

and the running times of $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{E}$ are the same as that of $\mathcal{A}$. The running time of $\mathcal{B}_1$ is independent (and less than that) of the running time of $\mathcal{A}$.

Proof. The proof is quite similar to that of Theorem 7, except for some initial game-hops. Here we will be focusing on these hops.

Denote Ω_{G_2}, Ω_G and Ω_H to be the set of all functions $G_2 : \{0,1\}^* \to \{0,1\}^{512}$, $G : \{0,1\}^{256} \to \{0,1\}^{256}$ and $H : \{0,1\}^* \to \{0,1\}^{256}$ respectively. Let $\mathcal{A}$ be an adversary in the ANO-CCA game for $\mathsf{pSaber.PKE}^{hy}$ issuing at most q_G (resp. q_H) quantum queries to the random oracles G (resp. H). Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_1$ described in Figure 26.

Game G_0: The game G_0 is equivalent to the ANO-CCA game for $\mathsf{pSaber.PKE}^{hy}$, except for some "cosmetic" changes. Namely, the pair (c_1^*, k^*) resulting from running $\mathsf{Encap}(\mathsf{pk}_b)$ for a uniformly random bit b is generated *before* the adversary $\mathcal{A}$ gets to choose a message $\mathbf{m}$. This change does not affect $\mathcal{A}$'s view in any way. Hence,

$$\left| \Pr[\mathsf{G}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.PKE}^{hy}}(\mathcal{A})$$

Game G_1: In game G_1, we implicitly divide the G-queries into at most three categories: (1) query is of the form (f, m) with $|f|+|m| = 512$ and $f = F(\mathsf{pk}_0)$, (2) query is of the form (f, m) with $|f|+|m| = 512$ and $f = F(\mathsf{pk}_1)$, and (3) the remaining queries. We then respond to the queries from the respective categories with $(G_{0\hat{k}}(m), G_{0r}(m))$, $(G_{1\hat{k}}(m), G_{1r}(m))$ and $G_2(m, c)$ respectively, where $G_{i\hat{k}}$, G_{ir} (for $i \in \{0,1\}$) are internal random functions; note that we say "at most" three categories because of the (unlikely) possibility that $F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$. It is not hard to verify that the output distributions of the G-oracle in games G_0 and G_1 are equivalent. Therefore,

$$\Pr[\mathsf{G}_1 = 1] = \Pr[\mathsf{G}_0 = 1].$$

Game G_2 In game G_2, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\hat{k}^*, r^*$ in setup of the game as "$\hat{k}^* \leftarrow G_{b\hat{k}}(m^*)$" and "$r^* \leftarrow G_{br}(m^*)$" (effectively, replacing the step "$(\hat{k}^*, r^*) \leftarrow G(F(\mathsf{pk}_b), m^*)$" in G_1). We then similarly generate the values $\hat{k}', r'$ w.r.t. the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ ($i \in \{0,1\}$) as "$\hat{k}' \leftarrow G_{i\hat{k}}(m')$" and "$r' \leftarrow G_{ir}(m')$" (replacing the step "$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}_i), m')$" in G_1).

Let bad denote the event where the public-keys pk_0 and pk_1 generated honestly in the setup satisfy the following: $F(\mathsf{pk}_0) = F(\mathsf{pk}_1)$. It is not hard to see that the games G_1 and G_2 are equivalent unless the event "bad" happens. As seen in the proof of Theorem 9 (specifically, the "$\mathsf{G}_2 \to \mathsf{G}_3$" hop), we also have the probability of the event bad occurring to be: $\Pr[\mathsf{bad}] \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}}$. Hence, we get

$$|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_1 = 1]| \leq \Pr[\mathsf{bad}] \leq \mathbf{Coll}^F_{\mathsf{Saber.PKE}}.$$

Game G_3 In game G_3, we modify the oracle $\mathsf{Dec}^{hy}(\mathsf{sk}_{1-b}, \cdot)$ such that if the decryption query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle uses $k^{\text{rej}} (= H(s_{1-b}, c_1^*))$ to decrypt c_2. Here k^{rej} is the key returned if $\mathsf{Decap}(\mathsf{sk}'_{1-b}, c_1^*)$ would have resulted in an "implicit rejection". Thus, it is not hard to see that the games G_2 and G_3 are equivalent unless c_1^* is not (implicitly) rejected by the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ operation, or in other words, if the following event occurs: "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c_1^*$" where for a uniformly random message $m^* \leftarrow_\$ \{0,1\}^{256}$ we have $(\hat{k}^*, r^*) \leftarrow_\$ (G_{b\hat{k}}(m^*), G_{br}(m^*))$, $\mathsf{Enc}(\mathsf{pk}_b, m^*; r^*) = c_1^*$, $\mathsf{Dec}(\mathsf{sk}_{1-b}, c_1^*) = m'$ and $(\hat{k}', r') \leftarrow_\$ (G_{(1-b)\hat{k}}(m'), G_{(1-b)r}(m'))$.

The analysis that follows is quite similar to the "$\mathsf{G}_5 \to \mathsf{G}_6$" game-hop in the proof of Theorem 9. Note that in the context of an experiment describing the above event, we have $G_{(1-b)r}(m')$ resulting in uniformly random coins $r' \leftarrow_\$ \{0,1\}^{256}$, since G_{br} is used to compute the ciphertext c_1^* and $G_{(1-b)r}$ is a random oracle independent to G_{br}. Borrowing the notation used to describe $\mathsf{Saber.PKE}$ (especially the KGen and Enc algorithms), note that the public-key pk_{1-b} consists of a uniformly random matrix $A_{1-b} \leftarrow_\$ R_q^{l \times l}$. Similarly the $\mathsf{Saber.PKE}$ ciphertext c_1^* (specifically, its second component) contains a vector $b^{*\prime} \in R_p^{l \times 1}$. Now considering the re-encryption check "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r') = c_1^*$" and looking at the Enc algorithm description, note that this implies "$A_{1-b}s' + h \bmod q = b^{*\prime}$" where h is a constant vector, $A_{1-b} \leftarrow_\$ R_q^{l \times l}$ and $s' \leftarrow_\$ \beta_\mu(R_q^{l \times 1})$ (specifically, $s' = \beta_\mu(R_q^{l \times 1}; r)$ for uniformly random $r' \leftarrow_\$ \{0,1\}^{256}$ as discussed above). Since the distribution $(A_{1-b}, A_{1-b}s' + h \bmod q)$ is computationally indistinguishable from (A_{1-b}, b'') for a uniformly random vector $b'' \leftarrow_\$ R_p^{l \times 1}$ based on the hardness of mod-LWR, the probability of the event "$A_{1-b}s' + h \bmod q = b^{*\prime}$" will be negligible. More concretely, it is not hard to obtain the following:

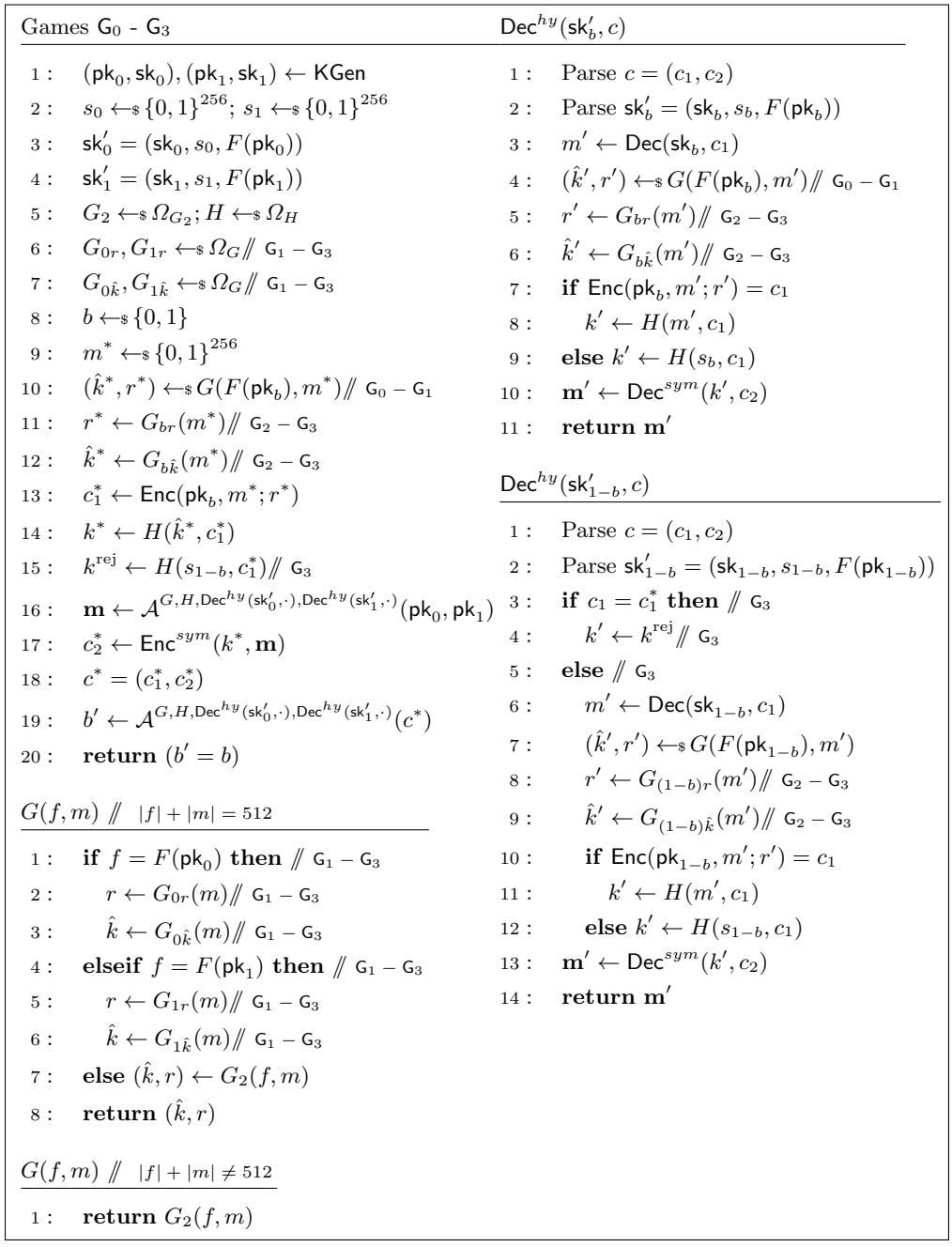


Fig. 26. Games G_0 – G_3 for the proof of Theorem 11.

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{1}{2^{256}}$$

where $\mathbf{Adv}^{\text{mod-lwr}}_{m,l,\mu,q,p}(\mathcal{B}_1)$ denotes the advantage of an adversary $\mathcal{B}_1$ in distinguishing between m samples from a mod-LWR distribution from that of a uniform distribution, with corresponding parameters l, μ, q and p. Also from the specification of pSaber, it can be shown that the size of the vector space $R_p^{l\times 1}$ is at least 2^{256}.

The rest of the proof follows very similarly to that of Theorem 7, where we then do the game-hop "$\mathsf{G}_0 \to \mathsf{G}_1$" of Theorem 7, skip the "$\mathsf{G}_1 \to \mathsf{G}_2$" hop – since effectively this is covered by our above $\mathsf{G}_0 \to \mathsf{G}_3$ hop – and proceed from "G_3" of Theorem 7 from then on, and so on.

Hence, it is not hard to finally arrive at

$$\begin{aligned}\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.PKE}^{hy}}(\mathcal{A}) \leq\ & \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{B}) + 2\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{C}) + \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} \\ & + 2\mathbf{Adv}^{\text{INT-CTXT}}_{\mathsf{DEM}}(\mathcal{E}) + \mathbf{Adv}^{\text{mod-lwr}}_{l,l,\mu,q,p}(\mathcal{B}_1) + \frac{4q_H}{2^{128}} + 8q_G\sqrt{\delta} + \frac{1}{2^{256}}\end{aligned}$$

C.12 Proof of Theorem 12

Theorem 12. *For any* SCFR-CCA *adversary* $\mathcal{A}$ *against the scheme* $\mathsf{Saber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most* q_G *(resp.* q_F*) queries to the quantum random oracle* G *(resp.* F*), we have*

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{Saber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{4\alpha(q_F+1)^3}{2^{256}} + \frac{4q_F}{2^{128}}$$

Here α (< 648) *is the constant from Lemma 1.*

Proof. Denote Ω_{G_2}, Ω_G and $\Omega_{H'}$ to be the set of all functions $G_2 : \{0,1\}^* \to \{0,1\}^{512}$, $G : \{0,1\}^{256} \to \{0,1\}^{256}$ and $H' : \{0,1\}^{256} \to \{0,1\}^{256}$ respectively.

Let $\mathcal{A}$ be an adversary in the SCFR-CCA game for Saber.KEM issuing at most q_G (resp., q_F) quantum queries to the random oracle G (resp. F).

The structure of the proof is very similar to that of Theorem 9. Namely, we do the sequence of game-hops $\mathsf{G}_0 \to \mathsf{G}_3$ as described in Figure 27. Since this sequence is similar to the game-hops "$\mathsf{G}_0 \to \mathsf{G}_3$" in the proof of Theorem 9, by a similar analysis we obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{4q_F}{2^{128}}$$

Note that the game G_0 is exactly the SCFR-CCA game for Saber.KEM. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{Saber.KEM}}(\mathcal{A})$$

Coming to the game G_3, note that the adversary $\mathcal{A}$ wins the game if it finally outputs a ciphertext c such that $\mathsf{Decap}(\mathsf{sk}'_0, c) = \mathsf{Decap}(\mathsf{sk}'_1, c)$. Let $m'_0 = \mathsf{Dec}(\mathsf{sk}_0, c)$, $m'_1 = \mathsf{Dec}(\mathsf{sk}_1, c)$, $\hat{k_0}' \leftarrow G_{0\hat{k}}(m'_0)$ and $\hat{k_1}' \leftarrow G_{1\hat{k}}(m'_1)$. There are four disjoint cases that need to be considered w.r.t. this winning condition:

- $\underline{\mathsf{Decap}(\mathsf{sk}'_0, c) = F(\hat{k_0}', F(c)) \wedge \mathsf{Decap}(\mathsf{sk}'_1, c) = F(\hat{k_1}', F(c))}$:
  - $\underline{\hat{k_0}' \neq \hat{k_1}'}$: The winning condition in this case translates to $F(\hat{k_0}', F(c)) = F(\hat{k_1}', F(c))$, where $\hat{k_0}' \neq \hat{k_1}'$. This implies a collision in the quantum random oracle F. Hence using Lemma 6, we can bound the probability of this sub-event by $\frac{\alpha(q_F+1)^3}{2^{256}}$ via a straightforward reduction to the collision-resistance of the random oracle F.
  - $\underline{\hat{k_0}' = \hat{k_1}'}$: In this sub-case, note that (m'_0, m'_1) is a *claw* w.r.t. the pair of random oracles $G_{0\hat{k}}$ and $G_{1\hat{k}}$. Using Lemma 1, we can bound the probability of this event by $\frac{\alpha(q_G+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance $(G_{0\hat{k}}, G_{1\hat{k}})$.

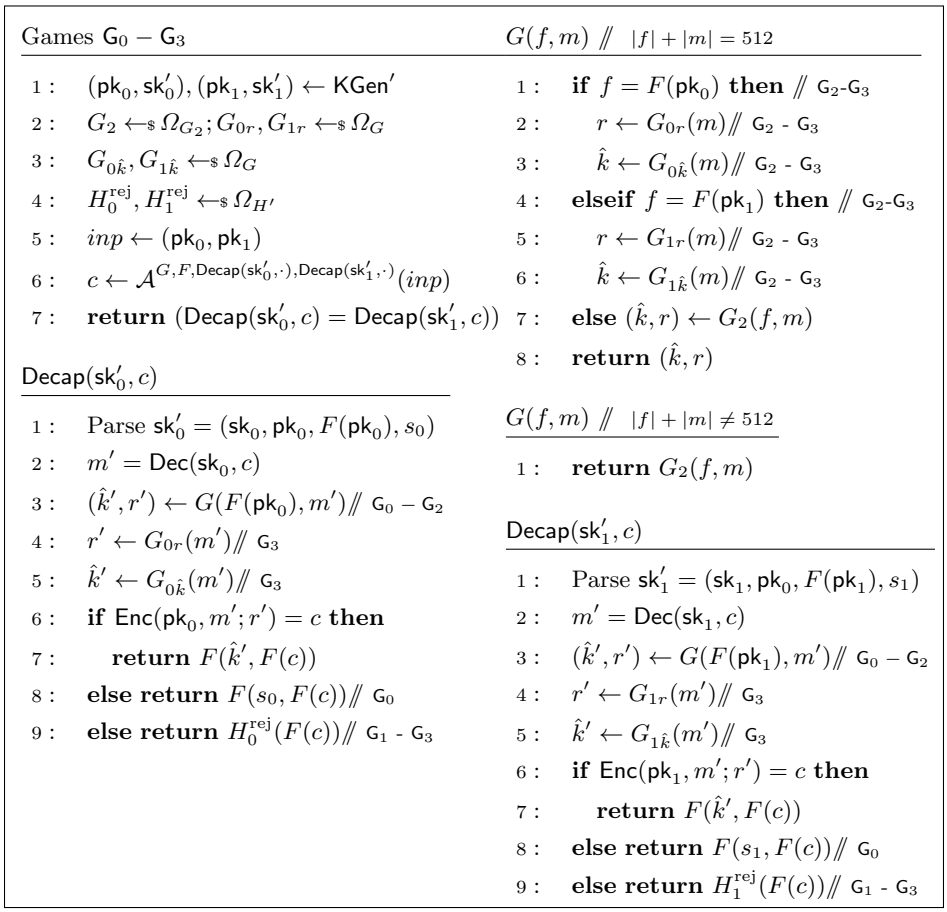


Fig. 27. Games G_0 – G_3 for the proof of Theorem 12.

$\mathsf{Encap}(\mathsf{pk})$		$\mathsf{Decap}(\mathsf{sk}, c)$	
1 :	$m \leftarrow_{\$} \mathcal{M}$	1 :	Parse $c = (c_1, c_2)$
2 :	$c_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, m; G(m))$	2 :	$m' \leftarrow \mathsf{Dec}(\mathsf{sk}, c_1)$
3 :	$c_2 \leftarrow H'(m, c_1)$	3 :	$c_1' \leftarrow \mathsf{Enc}(\mathsf{pk}, m'; G(m'))$
4 :	$c \leftarrow (c_1, c_2)$	4 :	**if** $c_1' = c_1 \wedge H'(m', c_1) = c_2$ **then**
5 :	$k = H(m, c)$	5 :	**return** $H(m', c)$
6 :	**return** (c, k)	6 :	**else return** $\perp$

Fig. 28. The KEM $\mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$.

- $\underline{\mathsf{Decap}(\mathsf{sk}_0', c) = F(\hat{k_0}', F(c)) \wedge \mathsf{Decap}(\mathsf{sk}_1', c) = H_1^{\mathrm{rej}}(F(c))}$: In this case, the winning condition translates to $F(\hat{k_0}', F(c)) = H_1^{\mathrm{rej}}(F(c))$. Note that then $((\hat{k_0}', F(c)), F(c))$ is a claw w.r.t. the pair of random oracles F and H_1^{rej}. Using Lemma 1, we can bound the probability of this event by $\frac{\alpha(q_F+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance (F, H_1^{rej}).
- $\underline{\mathsf{Decap}(\mathsf{sk}_0', c) = H_0^{\mathrm{rej}}(F(c)) \wedge \mathsf{Decap}(\mathsf{sk}_1', c) = F(\hat{k_1}', F(c))}$: The analysis here will be the same as the previous case.
- $\underline{\mathsf{Decap}(\mathsf{sk}_0', c) = H_0^{\mathrm{rej}}(F(c)) \wedge \mathsf{Decap}(\mathsf{sk}_1', c) = H_1^{\mathrm{rej}}(F(c))}$: In this case, the winning condition translates to $H_0^{\mathrm{rej}}(F(c)) = H_1^{\mathrm{rej}}(F(c))$. Note that $(F(c), F(c))$ is then a claw w.r.t. the pair of random oracles H_0^{rej} and H_1^{rej}. Using Lemma 1, we can bound the probability of this event by $\frac{\alpha(q_F+1)^3}{2^{256}}$ via a straightforward reduction to the claw-finding problem w.r.t. the instance $(H_0^{\mathrm{rej}}, H_1^{\mathrm{rej}})$.

From the above analysis, we have $\Pr[\mathsf{G}_3 = 1] \leq \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{4\alpha(q_F+1)^3}{2^{256}}$. Hence, we finally get

$$\mathbf{Adv}^{\text{SCFR-CCA}}_{\mathsf{Saber.KEM}}(\mathcal{A}) \leq \mathbf{Coll}^{F}_{\mathsf{Saber.PKE}} + \frac{\alpha(q_G+1)^3}{2^{256}} + \frac{4\alpha(q_F+1)^3}{2^{256}} + \frac{4q_F}{2^{128}}$$

D Analysis of the $\mathsf{HFO}^{\perp'}$ Transform

For the sake of convenience, we describe our $\mathsf{HFO}^{\perp'}$ transform again in Figure 28.

We now formally state the three theorems that respectively capture the enhancement properties of $\mathsf{HFO}^{\perp'}$ with regards to confidentiality, anonymity and robustness in the QROM.

Theorem 13. *Suppose* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* δ*-correct scheme. Then for any* IND-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\perp} = \mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$ *issuing at most* q_G[8]*,* q_H *and* $q_{H'}$ *queries to the quantum random oracles* G*,* H *and* H' *resp., and at most* q_D *queries to the (classical) decapsulation oracles, there exists an* OW-CPA *adversary* $\mathcal{B}$ *against* PKE *such that:*

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq 2(q_G + q_H) \cdot \sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{B})} + \frac{q_D}{|\overline{\mathcal{C}}_2|} + 4q_G\sqrt{\delta}\,.$$

Here $|\overline{\mathcal{C}}_2|$ *denotes the cardinality of the range of* H'*. Moreover, the running time of* $\mathcal{B}$ *is about the same as that of* $\mathcal{A}$*.*

The proof for this theorem follows very similarly to that of the $\mathsf{HFO}^{\perp}$ transform given in [28, Theorem 2]. We do not discuss it further here. Instead, we focus on the anonymity and robustness enhancing properties of $\mathsf{HFO}^{\perp'}$.

[8] Following [24, 27], we make the convention that the number q_O of queries made by an adversary $\mathcal{A}$ to a random oracle O counts the total number of times O is executed in the corresponding security experiment; i.e., the number of $\mathcal{A}$'s explicit queries to O plus the number of implicit queries to O made by the experiment.

Theorem 14. *Suppose* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a* δ*-correct and* γ*-spread scheme. Then for any* ANO-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\perp} = \mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$ *issuing at most* q_D *queries to the (classical) decapsulation oracles, and at most* q_G*,* q_H *and* $q_{H'}$ *queries to the quantum random oracles* G*,* H *and* H' *resp., there exist* wANO-CPA *adversary* $\mathcal{B}$*,* OW-CPA *adversary* $\mathcal{C}$ *against* PKE *and* SCFR-CPA *adversary* $\mathcal{D}$ *against the deterministic PKE scheme* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ *issuing at most* q_G *queries to* G *such that:*

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_G + q_H + q_{H'})\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})}$$
$$+ q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + \frac{2q_D}{|\overline{\mathcal{C}}_2|} + 2q_G(q_D + 4)\sqrt{2\delta} + 2^{-\gamma}\,.$$

Here $|\overline{\mathcal{C}}_2|$ *denotes the cardinality of the range of* H'*. Moreover, the running times of* $\mathcal{B}$*,* $\mathcal{C}$ *and* $\mathcal{D}$ *are the same as that of* $\mathcal{A}$*.*

First, note that to obtain anonymous (ANO-CCA secure) hybrid PKE schemes from explicit rejection KEMs via the KEM-DEM framework, Theorem 1 requires the KEM to satisfy a weaker notion of anonymity, namely wANO-CCA. In this context, Theorem 14 proves something stronger: the KEM $\mathsf{KEM}^{\perp}$ obtained from the $\mathsf{HFO}^{\perp'}$ transform is *strongly* anonymous, i.e., ANO-CCA secure.

Proof. The structure of the proof is quite similar to that of Theorem 5.

Denote Ω_G, Ω_H, $\Omega_{H'}$, Ω_{H_2}, Ω_{H_3}, $\Omega_{H'_2}$ and $\Omega_{H'_3}$ to be the set of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \overline{\mathcal{C}} \to \mathcal{K}$, $H' : \overline{\mathcal{C}}_1 \to \overline{\mathcal{C}}_2$, $H_2 : \mathcal{M} \times \overline{\mathcal{C}} \to \mathcal{K}$, $H'_2 : \mathcal{M} \times \overline{\mathcal{C}}_1 \to \overline{\mathcal{C}}_2$, $H_3 : \mathcal{M} \to \mathcal{K}$ and $H'_3 : \mathcal{M} \to \overline{\mathcal{C}}_2$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{M}$ is the message space of PKE, $\mathcal{K}$ is the encapsulated key-space of $\mathsf{KEM}^{\perp}$, $\overline{\mathcal{C}}_1$ is the ciphertext space of PKE and $\overline{\mathcal{C}}(= \overline{\mathcal{C}}_1 \times \overline{\mathcal{C}}_2)$ is the ciphertext space of $\mathsf{KEM}^{\perp}$.

Let $\mathcal{A}$ be an adversary in the ANO-CCA game for $\mathsf{KEM}^{\perp}$ issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G, q_H, $q_{H'}$ quantum queries to the random oracles G, H, H' respectively. Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_8$ described in Figure 29. (In Figure 29, w.r.t. changes in the $\mathsf{Decap}(\mathsf{sk}'_b, \cdot)$ oracle, $\mathsf{G}_x = \mathsf{G}_3$ if $b = 0$ and $\mathsf{G}_x = \mathsf{G}_{3.5}$ if $b = 1$. Similarly, w.r.t. changes in the $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ oracle, $\mathsf{G}_y = \mathsf{G}_{3.5}$ if $b = 0$ and $\mathsf{G}_y = \mathsf{G}_3$ if $b = 1$.)

Game G_0 The game G_0 is exactly the ANO-CCA game for $\mathsf{KEM}^{\perp}$ ($= \mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$). Hence,

$$\left|\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}\right| = \mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A})$$

Game G_1 In game G_1, we change the random oracle G such that it uniformly samples "good" random coins w.r.t. the key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$, as seen in the proof of Theorem 5 (game "G_2" to be specific). A similar analysis applies to this particular game-hop as well, and we thus obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq 2q_G\sqrt{2\delta}$$

Game $\mathsf{G}_{1.5}$ In game $\mathsf{G}_{1.5}$, we implicitly divide the H'-queries (m, c_1) into three disjoint categories: (1) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$, (2) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) \neq c_1 = \mathsf{Enc}(\mathsf{pk}_1, m; G(m))$, and (3) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) \neq c_1 \wedge \mathsf{Enc}(\mathsf{pk}_1, m; G(m)) \neq c_1$. We then respond to the queries from the respective categories with $H'_0(c_1)$, $H'_1(c_1)$ and $H'_2(m, c_1)$ respectively, where H'_0 and H'_1 are internal random functions not directly accessible to the adversary $\mathcal{A}$. Because G samples "good" random coins, it is not hard to see that the encryption functions $\mathsf{Enc}(\mathsf{pk}_0, .; G(\cdot))$ and $\mathsf{Enc}(\mathsf{pk}_1, .; G(\cdot))$ are injective, and hence, the output distributions of the H'-oracle in the games G_1 and $\mathsf{G}_{1.5}$ are equivalent. Therefore,

$$\Pr[\mathsf{G}_{1.5} = 1] = \Pr[\mathsf{G}_1 = 1]$$

Game G_2 In game G_2, we implicitly divide the H-queries (m, c) with $c = (c_1, c_2)$ into three disjoint categories again: (1) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$, (2) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) \neq c_1 = \mathsf{Enc}(\mathsf{pk}_1, m; G(m))$, and (3) $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) \neq c_1 \wedge \mathsf{Enc}(\mathsf{pk}_1, m; G(m)) \neq c_1$. We then respond to the queries from the respective categories with $H_0(c)$, $H_1(c)$ and $H_2(m, c)$ respectively, where H_0 and H_1 are internal random functions not directly accessible to the adversary $\mathcal{A}$. Because G samples "good" random coins, the encryption functions $\mathsf{Enc}(\mathsf{pk}_0, .; G(\cdot))$ and $\mathsf{Enc}(\mathsf{pk}_1, .; G(\cdot))$ are injective, and hence, it is not hard to see that the output distributions of the H-oracle in the games $\mathsf{G}_{1.5}$ and G_2 are equivalent. Therefore,

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_{1.5} = 1]$$

Games G_0 - G_8

$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'(1^\lambda)$
$G \leftarrow_\$ \Omega_G$
$G^{\text{good}} \leftarrow_\$ \Omega_{G^{\text{good}}}$
$G = G^{\text{good}}$ // G_1 - $\mathsf{G}_{2.25}$; G_3 - G_5
$H_0, H_1 \leftarrow_\$ \Omega_H; H'_0, H'_1 \leftarrow_\$ \Omega_{H'}$
$H_2 \leftarrow_\$ \Omega_{H_2}; H_3 \leftarrow_\$ \Omega_{H_3}$
$H'_2 \leftarrow_\$ \Omega_{H'_2}; H'_3 \leftarrow_\$ \Omega_{H'_3}$
$b \leftarrow_\$ \{0,1\}$
$m^* \leftarrow_\$ \mathcal{M}$
$r^* \leftarrow G(m^*)$ // $\mathsf{G}_0 - \mathsf{G}_6$
$r^* \leftarrow_\$ \mathcal{R}$ // $\mathsf{G}_7 - \mathsf{G}_8$
$c_1^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$
$c_2^* \leftarrow H'(m^*, c_1^*)$ // $\mathsf{G}_0 - \mathsf{G}_6$
$c_2^* \leftarrow_\$ \overline{\mathcal{C}}_2$ // $\mathsf{G}_7 - \mathsf{G}_8$
$c^* = (c_1^*, c_2^*)$
$k^* \leftarrow H(m^*, c^*)$ // $\mathsf{G}_0 - \mathsf{G}_6$
$k^* \leftarrow_\$ \mathcal{K}$ // $\mathsf{G}_7 - \mathsf{G}_8$
$inp \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$
$i \leftarrow_\$ \{1, \ldots, q_G + q_H + q_{H'}\}$ // G_8
run $\mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}'_0,\cdot),\mathsf{Decap}(\mathsf{sk}'_1,\cdot)}(inp)$ until i-th query to $G \times H_3 \times H'_3$ // G_8
measure the i-th query and let the outcome be $\hat{m}$ // G_8
return $(\hat{m} = m^*)$ // G_8
$b' \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}'_0,\cdot),\mathsf{Decap}(\mathsf{sk}'_1,\cdot)}(inp)$
return $(b' = b)$

$H(m, c)$

if $c = c^*$ **return** $H_3(m)$ // G_5 - G_8
Parse $c = (c_1, c_2)$
if $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$ // G_2-G_8
  return $H_0(c)$ // G_2-G_8
if $\mathsf{Enc}(\mathsf{pk}_1, m; G(m)) = c_1$ // G_2-G_8
  return $H_1(c)$ // G_2-G_8
return $H_2(m, c)$

$H'(m, c_1)$

if $c_1 = c_1^*$ **return** $H'_3(m)$ // $\mathsf{G}_{4.5}$-G_8
if $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$ // $\mathsf{G}_{1.5}$-G_8
  return $H'_0(c_1)$ // $\mathsf{G}_{1.5}$-G_8
if $\mathsf{Enc}(\mathsf{pk}_1, m; G(m)) = c_1$ // $\mathsf{G}_{1.5}$-G_8
  return $H'_1(c_1)$ // $\mathsf{G}_{1.5}$-G_8
return $H'_2(m, c_1)$

$\mathsf{Decap}(\mathsf{sk}'_0, c)$ // $\mathsf{G}_{3.5}$ - G_8

Parse $c = (c_1, c_2)$
if $c_1 = c_1^*$ **return** $\perp$
if $H'_0(c_1) = c_2$ **then**
  return $H_0(c)$
else return $\perp$

$\mathsf{Decap}(\mathsf{sk}'_1, c)$ // G_4 - G_8

Parse $c = (c_1, c_2)$
if $c_1 = c_1^*$ **return** $\perp$
if $H'_1(c_1) = c_2$ **then**
  return $H_1(c)$
else return $\perp$

$\mathsf{Decap}(\mathsf{sk}'_b, c)$ // G_0 - G_x

Parse $c = (c_1, c_2)$
if $c_1 = c_1^*$ **return** $\perp$ // $\mathsf{G}_{2.25}$ - G_x
$m' = \mathsf{Dec}(\mathsf{sk}'_b, c_1)$
if $\mathsf{Enc}(\mathsf{pk}_b, m', G(m')) = c_1 \wedge H'(m', c_1) = c_2$ **then**
  return $H(m', c)$
else return $\perp$

$\mathsf{Decap}(\mathsf{sk}'_{1-b}, c)$ // G_0 - G_y

Parse $c = (c_1, c_2)$
if $c_1 = c_1^*$ **return** $\perp$ // $\mathsf{G}_{2.75}$ - G_y
$m' = \mathsf{Dec}(\mathsf{sk}'_{1-b}, c_{1-b})$
if $\mathsf{Enc}(\mathsf{pk}_{1-b}, m', G(m')) = c_1 \wedge H'(m', c_1) = c_2$ **then**
  return $H(m', c)$
else return $\perp$

Fig. 29. Games G_0 – G_8 for the proof of Theorem 14.

Game $\mathsf{G}_{2.25}$: In game $\mathsf{G}_{2.25}$, we modify the oracle $\mathsf{Decap}(\mathsf{sk}'_b, \cdot)$ such that if the decapsulation query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle returns $\perp$. This is because, querying (c_1^*, c_2) to the (unchanged) oracle $\mathsf{Decap}(\mathsf{sk}'_b, \cdot)$ of game G_2 would not result in a $\perp$ response if and only if $c_2 = c_2^*$; as G samples good random coins, $\mathsf{Dec}(\mathsf{sk}_b, c_1^*) = m^*$, and hence, to get a *non*-$\perp$ response, it must be the case $H'(m^*, c_1^*)(= c_2^*) = c_2$. But since (c_1^*, c_2^*) is a forbidden query, we have the games G_2 and $\mathsf{G}_{2.5}$ to be equivalent. Hence,

$$\Pr[\mathsf{G}_{2.25} = 1] = \Pr[\mathsf{G}_2 = 1]$$

Game $\mathsf{G}_{2.5}$: In game $\mathsf{G}_{2.5}$, we reset G to be an ideal random oracle, i.e., $G(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_0 \to \mathsf{G}_1$, by using a similar analysis, it is not hard to obtain

$$|\Pr[\mathsf{G}_{2.5} = 1] - \Pr[\mathsf{G}_{2.25} = 1]| \leq 2q_G\sqrt{2\delta}$$

Game $\mathsf{G}_{2.75}$: In game $\mathsf{G}_{2.75}$, we modify the oracle $\mathsf{Decap}(\mathsf{sk}'_{1-b}, \cdot)$ such that if the decapsulation query is (c_1, c_2) where $c_1 = c_1^*$, then the oracle returns $\perp$. We can see that the games $\mathsf{G}_{2.5}$ and $\mathsf{G}_{2.75}$ are equivalent unless $\mathsf{Decap}(\mathsf{sk}'_{1-b}, (c_1^*, c_2))$ $(c_2 \neq c_2^*)$ does not result in a $\perp$ in game $\mathsf{G}_{2.5}$, or in other words, if the following event occurs: $\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; G(m')) = c_1^*$ *and* $H'(m', c_1^*) = c_2$ where $\mathsf{Enc}(\mathsf{pk}_b, m^*; G(m^*)) = c_1^*$ and $\mathsf{Dec}(\mathsf{sk}_{1-b}, c_1^*) = m'$ (for $m^* \leftarrow_{\$} \mathcal{M}$).

There are two sub-events to consider w.r.t. the above event:

1. $\underline{m' \neq m^*}$: In this case, the random oracle G on a new query m' will return uniformly random coins $r \leftarrow_{\$} \mathcal{R}$. Since PKE is γ-spread, for the key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b})$ and message m', we have the re-encryption check, namely "$\mathsf{Enc}(\mathsf{pk}_{1-b}, m'; r) = c_1^*$", to hold with probability $\leq 2^{-\gamma}$, for uniformly random r.
2. $\underline{m' = m^*}$: In this case, note that the additional hash check, namely "$H'(m', c_1^*) = c_2$", succeeds if and only if $c_2 = c_2^*$ (since $m' = m^*$, we have $H'(m^*, c_1^*) = c_2^*$). But because (c_1^*, c_2^*) is a forbidden decapsulation query, the probability of this sub-event occurring is zero.

Hence,

$$|\Pr[\mathsf{G}_{2.75} = 1] - \Pr[\mathsf{G}_{2.5} = 1]| \leq 2^{-\gamma}$$

Game G_3: In game G_3, we (re)-modify the random oracle G such that it uniformly samples "good" random coins w.r.t. the key-pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$. A similar analysis as the $\mathsf{G}_0 \to \mathsf{G}_1$ hop shows that

$$|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_{2.75} = 1]| \leq 2q_G\sqrt{2\delta}$$

Game $\mathsf{G}_{3.5}$ In game $\mathsf{G}_{3.5}$, we change the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle such that there is no need for the secret key sk'_0. When $\mathcal{A}$ queries the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle on $c = (c_1, c_2)$ $(c_1 \neq c_1^*)$, the key $k = H_0(c)$ is returned if the check "$H'_0(c_1) = c_2$" is satisfied; otherwise $\perp$ is returned. Let $m' = \mathsf{Dec}(\mathsf{sk}_0, c_1)$. Consider the following three cases:

1. $\mathsf{Enc}(\mathsf{pk}_0, m'; G(m')) = c_1 \wedge H'(m', c_1) = c_2$. In this case, it is not hard to verify that the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracles in games G_3 and $\mathsf{G}_{3.5}$ return the same value $H_0(c)$. (Note that in this case, $H'(m', c_1) = H'_0(c_1)$.)
2. $\mathsf{Enc}(\mathsf{pk}_0, m'; G(m')) = c_1 \wedge H'(m', c_1) \neq c_2$. In this case, it is again not hard to verify that the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracles in games G_3 and $\mathsf{G}_{3.5}$ return $\perp$.
3. $\mathsf{Enc}(\mathsf{pk}_0, m'; G(m')) \neq c_1$. In game G_3, the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle returns $\perp$. In game $\mathsf{G}_{3.5}$, the only way $\mathcal{A}$ gets prior access to the function H'_0 is if it made a H'-query (m'', c_1) such that $\mathsf{Enc}(\mathsf{pk}_0, m''; G(m'')) = c_1$. But because G samples good random coins, we have $\mathsf{Dec}(\mathsf{sk}_0, c_1) = m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}_0, m'; G(m')) \neq c_1$". Therefore, such a prior access is not possible and $H'_0(c_1)$ will be a uniformly random value in $\mathcal{A}$'s view. As a result, the probability that the "$H'_0(c_1) = c_2$" is satisfied is $\frac{1}{\mathcal{C}_2}$, and hence, the $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ oracle in $\mathsf{G}_{3.5}$ returns $\perp$ with probability $1 - \frac{1}{\mathcal{C}_2}$.

From applying a union bound over (at most) q_D number of decapsulation queries made by $\mathcal{A}$, we obtain

$$|\Pr[\mathsf{G}_{3.5} = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{q_D}{\mathcal{C}_2}$$

Game G_4 In game G_4, we change the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle such when $\mathcal{A}$ queries the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle on $c = (c_1, c_2)$ $(c_1 \neq c_1^*)$, the key $k = H_1(c)$ is returned if the check "$H'_1(c_1) = c_2$" is satisfied; otherwise $\perp$ is returned. The analysis here follows quite similarly to that of the previous game-hop except that the games $\mathsf{G}_{3.5}$ and G_4 could *additionally* differ if, w.r.t. the analogous cases of 1 and 2 above (namely "$\mathsf{Enc}(\mathsf{pk}_1, m'; G(m')) = c_1 \wedge H'(m', c_1) = c_2$" and "$\mathsf{Enc}(\mathsf{pk}_1, m'; G(m')) = c_1 \wedge H'(m', c_1) \neq c_2$" respectively, where $m' = \mathsf{Dec}(\mathsf{sk}_1, c_1)$), $\mathcal{A}$ asks for the decapsulation of a ciphertext $\hat{c} = (\hat{c}_1, \hat{c}_2)$ such that $m' = \mathsf{Dec}(\mathsf{sk}_1, \hat{c}_1)$ and $\mathsf{Enc}(\mathsf{pk}_0, m'; G(m')) = \mathsf{Enc}(\mathsf{pk}_1, m'; G(m') = \hat{c}_1$; in such a peculiar event, $H'(m', \hat{c}_1) = H'_0(\hat{c}_1)$, and hence, has no clear relation with the check "$H'_1(\hat{c}_1) = \hat{c}_1$" in game G_4. We bound the probability of this peculiar event (i.e., $\mathcal{A}$ asking for the decapsulation of such an above ciphertext $\hat{c}$ w.r.t. sk'_1) by the advantage of an SCFR-CPA adversary $\mathcal{D}$ against the deterministic scheme $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$, as was done similarly in the "$(\mathsf{G}_{3.5} \to \mathsf{G}_4)$" hop in the proof of Theorem 5. Hence, by a similar analysis, it is not hard to see that

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_{3.5} = 1]| \leq q_D \cdot (\mathbf{Adv}^{\text{SCFR-CPA}}_{\text{PKE}_1}(\mathcal{D}) + 2q_G\sqrt{2\delta}) + \frac{q_D}{\overline{\mathcal{C}}_2}$$

where the additional term "$\frac{q_D}{\overline{\mathcal{C}}_2}$" in the above bound is because of the difference in games $\mathsf{G}_{3.5}$ and G_4 w.r.t. the analogous case of 3 above (namely "$\mathsf{Enc}(\mathsf{pk}_1, m'; G(m')) \neq c_1$", where $m' = \mathsf{Dec}(\mathsf{sk}_1, c_1)$, and the $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$ oracle returns $\perp$ in $\mathsf{G}_{3.5}$, but does not return a $\perp$ in G_4 with probability $\frac{1}{\overline{\mathcal{C}}_2}$.)

Game $\mathsf{G}_{4.5}$ In game $\mathsf{G}_{4.5}$, we answer H'-queries of the form (m, c_1^*) with $H'_3(m)$, where H'_3 is an independent random function. Since G samples good randomness, there are at most two H'-queries worth considering, namely (m_0, c_1^*) and (m_1, c_1^*), where $\mathsf{Enc}(\mathsf{pk}_0, m_0; G(m_0)) = c_1^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G(m_1)) = c_1^*$ (for the other H'-queries (m', c_1^*), where $m' \notin \{m_0, m_1\}$. we are replacing the oracle outputs $H'_2(m', c_1^*)$ in G_4 with $H'_3(m')$ in $\mathsf{G}_{4.5}$). W.r.t. these two queries, the H' oracle would return $H'_0(c_1^*)$, $H'_1(c_1^*)$ respectively in G_4, and $H'_3(m_0)$, $H'_3(m_1)$ respectively in $\mathsf{G}_{4.5}$. The adversary $\mathcal{A}$'s view would be identical even after this change because the random values $H'_0(c_1^*)$, $H'_1(c_1^*)$ are only accessible to $\mathcal{A}$ via the H'-oracle in G_4, and in particular, not through the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles since decapsulation queries of the form (c_1^*, c_2) result in a $\perp$. Hence in $\mathsf{G}_{4.5}$, we are effectively replacing (at most) two uniformly random values that can only be accessed via the H'-oracle by $\mathcal{A}$ with two other uniformly random values (the simpler case of $m_0 = m_1$ would follow similarly). Since the output distributions of the H'-oracle in the games G_4 and $\mathsf{G}_{4.5}$ are equivalent, we have

$$\Pr[\mathsf{G}_{4.5} = 1] = \Pr[\mathsf{G}_4 = 1]$$

Game G_5 In game G_5, we answer H-queries of the form (m, c^*) (where $c^* = (c_1^*, c_2^*)$) with $H_3(m)$, where H_3 is an independent random function. Since G samples only good randomness, there are at most two H-queries worth considering, namely (m_0, c^*) and (m_1, c^*), where $\mathsf{Enc}(\mathsf{pk}_0, m_0; G(m_0)) = c_1^*$ and $\mathsf{Enc}(\mathsf{pk}_1, m_1; G(m_1)) = c_1^*$ (for the other H-queries (m', c^*), where $m' \notin \{m_0, m_1\}$. we are replacing the oracle outputs $H_2(m', c^*)$ in $\mathsf{G}_{4.5}$ with $H_3(m')$ in G_5). W.r.t. these two queries, the H oracle would return $H_0(c^*)$, $H_1(c^*)$ respectively in $\mathsf{G}_{4.5}$, and $H_3(m_0)$, $H_3(m_1)$ respectively in G_5. The adversary $\mathcal{A}$'s view would be identical even after this change because the random values $H_0(c^*)$, $H_1(c^*)$ are only accessible to $\mathcal{A}$ via the H-oracle in $\mathsf{G}_{4.5}$, and in particular, not through the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles since c^* is a forbidden decapsulation query. Hence in G_5, we are effectively replacing (at most) two uniformly random values that can only be accessed via the H-oracle by $\mathcal{A}$ with two other uniformly random values (the simpler case of $m_0 = m_1$ would follow similarly). Since the output distributions of the H-oracle in the games $\mathsf{G}_{4.5}$ and G_5 are equivalent, we have

$$\Pr[\mathsf{G}_5 = 1] = \Pr[\mathsf{G}_{4.5} = 1]$$

Game G_6 In game G_6, we reset G to be an ideal random oracle, i.e., $G(m)$ now returns uniformly random coins from $\mathcal{R}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop $\mathsf{G}_0 \to \mathsf{G}_1$, by using a similar analysis, we obtain

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq 2q_G\sqrt{2\delta}$$

Game G_7 In the setup of game G_7, we replace the hash evaluations "$r^* \leftarrow G(m^*)$", "$c_2^* \leftarrow H'(m^*, c_1^*)(= H'_3(m^*))$" and "$k^* \leftarrow H(m^*, c^*)(= H_3(m^*))$" with "$r^* \leftarrow_\$ \mathcal{R}$", "$c_2^* \leftarrow_\$ \overline{\mathcal{C}}_2$" and "$k^* \leftarrow_\$ \mathcal{K}$" respectively. That is, r^*, k^* and c_2^* are now uniformly random values that are generated independently of the random oracles G, H_3 and H'_3. We use Lemma 4 to bound the difference in the success probabilities of $\mathcal{A}$ in G_6 and G_7. Let A be

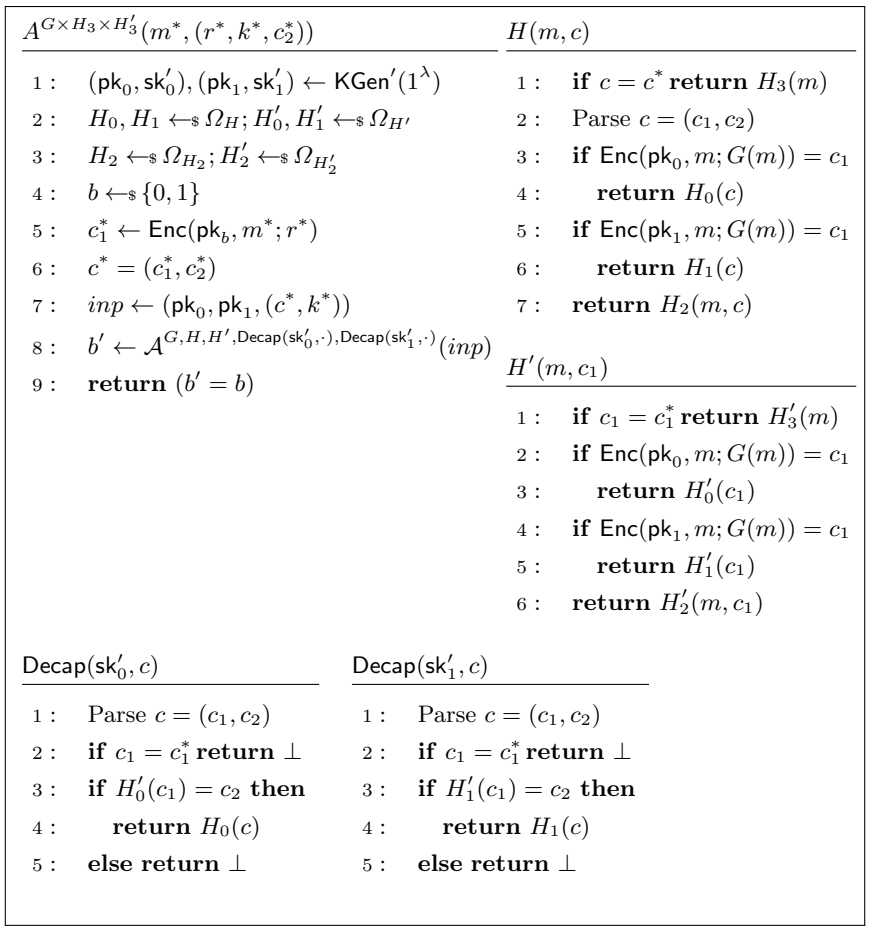


Fig. 30. Algorithm $A^{G\times H_3\times H_3'}$ for the proof of Theorem 14.

an oracle algorithm that has quantum access to the random oracle $G \times H_3 \times H'_3$, where $(G \times H_3 \times H'_3)(m) = (G(m), H_3(m), H_3(m))$. Figure 30 describes $A^{G \times H_3 \times H'_3}$'s operation on input $(m^*, (r^*, k^*, c_2^*))$. Note that the algorithm $A^{G \times H_3 \times H'_3}$ makes at most $q_G + q_H + q_{H'}$ number of queries to the random oracle $G \times H_3 \times H'_3$ to respond to $\mathcal{A}$'s oracle queries[9].

Let B be an oracle algorithm that on input m^* does the following: picks $i \leftarrow_{\$} \{1, \ldots, q_G + q_H + q_{H'}\}$, generates $r^* \leftarrow_{\$} \mathcal{R}$, $k^* \leftarrow_{\$} \mathcal{K}$ and $c_2^* \leftarrow_{\$} \overline{\mathcal{C}}_2$, runs $A^{G \times H_3 \times H'_3}(m^*, (r^*, k^*, c_2^*))$ until the i-th query, measures the argument of the $(G \times H_3 \times H'_3)$-query in the computational basis and outputs the measurement outcome (if $A^{G \times H_3 \times H'_3}$ makes less than i queries, B outputs $\perp$). With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_6 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_7 = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 4 w.r.t. the algorithm $A^{G \times H_3 \times H'_3}$. Therefore, we now define game G_8 (see Fig. 29) such that $P_B = \Pr[\mathsf{G}_8 = 1]$, where P_B is as defined in Lemma 4 w.r.t. the algorithm $B^{G \times H_3 \times H'_3}$. From Lemma 4, we thus have

$$|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]| \leq 2(q_G + q_H + q_{H'})\sqrt{\Pr[\mathsf{G}_8 = 1]}$$

We now bound the success probability of $\mathcal{A}$ in G_7 by the advantage of an adversary $\mathcal{B}$ in the wANO-CPA game of PKE. Upon receiving public-keys pk_0 and pk_1 along with the ciphertext c_1^*, where $c_1^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ for uniformly random bit $b (\leftarrow_{\$} \{0,1\})$, (secret) message $m^* (\leftarrow_{\$} \mathcal{M})$ and randomness $r^* (\leftarrow_{\$} \mathcal{R})$ chosen by the challenger, $\mathcal{B}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_7.
- Uses a $2q_G$-wise independent function to simulate the random oracle G, uses four different $2q_H$-wise independent functions to simulate the random oracles H_0, H_1, H_2, H_3 respectively, and uses four different $2q_{H'}$-wise independent functions to simulate the random oracles H'_0, H'_1, H'_2, H'_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. The random oracles H and H' are simulated in the same way as in G_7.
- Answers decapsulation queries the same way as in G_7 by using the oracles H_i, H'_i $(i \in \{0,1\})$.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow_{\$} \mathcal{K}$ and a ciphertext component $c_2^* \leftarrow_{\$} \overline{\mathcal{C}}_2$, and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$ where $c^* = (c_1^*, c_2^*)$.
- After obtaining a bit b' from $\mathcal{A}$, forwards b' to its wANO-CPA challenger as the final message.

It is easy to see that $|\Pr[\mathsf{G}_7 = 1] - \frac{1}{2}| = \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B})$. Now we bound the success probability of $\mathcal{A}$ in G_8 by the advantage of an adversary $\mathcal{C}$ in the OW-CPA game of PKE. Upon receiving a public-key pk along with a ciphertext c_1^*, where $c_1^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^* (\leftarrow_{\$} \mathcal{M})$ and randomness $r^* (\leftarrow_{\$} \mathcal{R})$ chosen by the challenger, $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_8.
- Uses a $2q_G$-wise independent function to simulate the random oracle G, uses four different $2q_H$-wise independent functions to simulate the random oracles H_0, H_1, H_2, H_3 respectively, and uses four different $2q_{H'}$-wise independent functions to simulate the random oracles H'_0, H'_1, H'_2, H'_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also evaluates $\mathcal{A}$'s G-, H- and H'-queries using the oracle $G \times H_3 \times H'_3$; the random oracles H and H' are simulated in the same way as in G_8.
- Answers decapsulation queries the same way as in G_8 by using the oracles H_i, H'_i $(i \in \{0,1\})$.
- For $\mathcal{A}$'s challenge query, first samples a uniformly random bit $b \leftarrow_{\$} \{0,1\}$ and sets $\mathsf{pk}_b = \mathsf{pk}$. Then generates a key-pair $(\mathsf{pk}_{1-b}, \mathsf{sk}_{1-b}) \leftarrow \mathsf{KGen}(1^\lambda)$, samples a uniformly random key $k^* \leftarrow_{\$} \mathcal{K}$, a ciphertext component $c_2^* \leftarrow_{\$} \overline{\mathcal{C}}_2$, and responds with $(\mathsf{pk}_0, \mathsf{pk}_1, (c^*, k^*))$ where $c^* = (c_1^*, c_2^*)$. (By doing this, note that we have $c_1^* \leftarrow \mathsf{Enc}(\mathsf{pk}_b, m^*; r^*)$ in $\mathcal{A}$'s view.)
- Selects $i \leftarrow_{\$} \{1, \ldots, q_G + q_H + q_{H'}\}$, measures the i-th query to oracle $G \times H_3 \times H'_3$ and returns the outcome $\hat{m}$.

Again, it is not hard to see that $\Pr[\mathsf{G}_8 = 1] \leq \mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})$. Hence by collecting all of the above bounds, we arrive at

$$\mathbf{Adv}^{\text{ANO-CCA}}_{\mathsf{KEM}^\perp}(\mathcal{A}) \leq \mathbf{Adv}^{\text{wANO-CPA}}_{\mathsf{PKE}}(\mathcal{B}) + 2(q_G + q_H + q_{H'})\sqrt{\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{PKE}}(\mathcal{C})}$$
$$+ q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{D}) + \frac{2q_D}{|\overline{\mathcal{C}}_2|} + 2q_G(q_D + 4)\sqrt{2\delta} + 2^{-\gamma}$$

[9] For example, if $A^{G \times H_3 \times H'_3}$ wants to respond to $\mathcal{A}$'s H-query, then $A^{G \times H_3 \times H'_3}$ prepares a uniform superposition of all states in the output registers corresponding to G and H'_3 (see [38] for particulars of this "trick").

Regarding robustness of KEMs constructed using $\mathsf{HFO}^{\perp'}$, we have the following.

Theorem 15. *Suppose* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ*-correct. Then for any* SROB-CCA *adversary* $\mathcal{A}$ *against* $\mathsf{KEM}^{\perp} = \mathsf{HFO}^{\perp'}[\mathsf{PKE}, G, H, H']$ *issuing at most* q_D *queries to the (classical) decapsulation oracles, and at most* q_G, q_H, $q_{H'}$ *queries to the quantum random oracles* G, H, H' *resp., there exists an* SCFR-CPA *adversary* $\mathcal{B}$ *against the deterministic PKE scheme* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, G]$ *issuing at most* q_G *queries to* G *such that:*

$$\mathbf{Adv}^{\text{SROB-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{C(q_{H'}+1)^3}{|\overline{\mathcal{C}}_2|} + \frac{2q_D}{|\overline{\mathcal{C}}_2|} + 2q_G(q_D+2)\sqrt{2\delta}\,.$$

Here $|\overline{\mathcal{C}}_2|$ *denotes the cardinality of the range of* H' *and* C *is the constant from Lemma 1. Moreover, the running time of* $\mathcal{B}$ *is the same as that of* $\mathcal{A}$.

Games G_0 - G_4

$(\mathsf{pk}_0, \mathsf{sk}'_0), (\mathsf{pk}_1, \mathsf{sk}'_1) \leftarrow \mathsf{KGen}'(1^\lambda)$
$G \leftarrow\!\!\$\ \Omega_G$
$G^{\text{good}} \leftarrow\!\!\$\ \Omega_{G^{\text{good}}}$
$G = G^{\text{good}}$ // G_1 - G_3
$H_0, H_1 \leftarrow\!\!\$\ \Omega_H$
$H'_0, H'_1 \leftarrow\!\!\$\ \Omega_{H'}$
$H_2 \leftarrow\!\!\$\ \Omega_{H_2}$; $H'_2 \leftarrow\!\!\$\ \Omega_{H'_2}$
$inp \leftarrow (\mathsf{pk}_0, \mathsf{pk}_1)$
$c \leftarrow \mathcal{A}^{G,H,H',\mathsf{Decap}(\mathsf{sk}'_0,\cdot),\mathsf{Decap}(\mathsf{sk}'_1,\cdot)}(inp)$
return $(\mathsf{Decap}(\mathsf{sk}'_0, c) \neq \perp \wedge \mathsf{Decap}(\mathsf{sk}'_1, c) \neq \perp)$

$H(m, c)$

Parse $c = (c_1, c_2)$
if $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$ // G_2-G_4
  return $H_0(c)$ // G_2-G_4
if $\mathsf{Enc}(\mathsf{pk}_1, m; G(m)) = c_1$ // G_2-G_4
  return $H_1(c)$ // G_2-G_4
return $H_2(m, c)$

$H'(m, c_1)$

if $\mathsf{Enc}(\mathsf{pk}_0, m; G(m)) = c_1$ // $\mathsf{G}_{1.5}$-G_4
  return $H'_0(c_1)$ // $\mathsf{G}_{1.5}$-G_4
if $\mathsf{Enc}(\mathsf{pk}_1, m; G(m)) = c_1$ // $\mathsf{G}_{1.5}$-G_4
  return $H'_1(c_1)$ // $\mathsf{G}_{1.5}$-G_4
return $H'_2(m, c_1)$

$\mathsf{Decap}(\mathsf{sk}'_0, c)$

Parse $c = (c_1, c_2)$
if $H'_0(c_1) = c_2$ **then** // $\mathsf{G}_{2.5}$ - G_4
  return $H_0(c)$ // $\mathsf{G}_{2.5}$ - G_4
else return $\perp$ // $\mathsf{G}_{2.5}$ - G_4
$m' = \mathsf{Dec}(\mathsf{sk}'_0, c_1)$
if $\mathsf{Enc}(\mathsf{pk}_0, m', G(m')) = c_1 \wedge H'(m', c_1) = c_2$ **then**
  return $H(m', c)$
else return $\perp$

$\mathsf{Decap}(\mathsf{sk}'_1, c)$

Parse $c = (c_1, c_2)$
if $H'_1(c_1) = c_2$ **then** // G_3 - G_4
  return $H_1(c)$ // G_3 - G_4
else return $\perp$ // G_3 - G_4
$m' = \mathsf{Dec}(\mathsf{sk}'_1, c_1)$
if $\mathsf{Enc}(\mathsf{pk}_1, m', G(m')) = c_1 \wedge H'(m', c_1) = c_2$ **then**
  return $H(m', c)$
else return $\perp$

Fig. 31. Games G_0 – G_4 for the proof of Theorem 15.

Proof. Denote Ω_G, Ω_H, $\Omega_{H'}$, Ω_{H_2}, $\Omega_{H'_2}$ to be the set of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \overline{\mathcal{C}} \to \mathcal{K}$, $H' : \overline{\mathcal{C}}_1 \to \overline{\mathcal{C}}_2$, $H_2 : \mathcal{M} \times \overline{\mathcal{C}} \to \mathcal{K}$, $H'_2 : \mathcal{M} \times \overline{\mathcal{C}}_1 \to \overline{\mathcal{C}}_2$ respectively, where $\mathcal{R}$ is the set of random coins used in Enc, $\mathcal{M}$ is the message space of PKE, $\mathcal{K}$ is the encapsulated key-space of $\mathsf{KEM}^{\perp}$, $\overline{\mathcal{C}}_1$ is the ciphertext space of PKE and $\overline{\mathcal{C}}(= \overline{\mathcal{C}}_1 \times \overline{\mathcal{C}}_2)$ is the ciphertext space of $\mathsf{KEM}^{\perp}$.

Let $\mathcal{A}$ be an adversary in the SROB-CCA game for $\mathsf{KEM}^{\perp}$ issuing at most q_D (classical) queries to the oracles $\mathsf{Decap}(\mathsf{sk}'_0, \cdot)$ and $\mathsf{Decap}(\mathsf{sk}'_1, \cdot)$, and q_G, q_H, $q_{H'}$ quantum queries to the random oracles G, H, H' respectively.

The structure of the proof is quite similar to that of Theorem 14. Basically we do a similar sequence of game-hops as in the proof of Theorem 14 until the point where we can simulate the decapsulation oracles $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ ($i \in \{0, 1\}$) without requiring the corresponding secret keys sk'_i. In the final game-hop, we reset G to be an ideal random oracle.

To be specific, the sequence of game-hops $\mathsf{G}_0 \rightarrow \mathsf{G}_4$ as described in Figure 31 is similar to the sequence "$\mathsf{G}_0 \rightarrow \mathsf{G}_2$", "$\mathsf{G}_{3.5} \rightarrow \mathsf{G}_4$" and "$\mathsf{G}_5 \rightarrow \mathsf{G}_6$" w.r.t. the proof of Theorem 14 (i.e., we do not consider the intermediate sequence "$\mathsf{G}_2 \rightarrow \mathsf{G}_{3.5}$"). By a similar analysis as that of the proof of Theorem 14 w.r.t. these game-hops, it is not hard to obtain

$$|\Pr[\mathsf{G}_0 = 1] - \Pr[\mathsf{G}_4 = 1]| \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{2q_D}{\overline{\mathcal{C}}_2} + 2q_G(q_D + 2)\sqrt{2\delta}$$

Note that the game G_0 is exactly the SROB-CCA game for $\mathsf{KEM}^{\perp}$. Hence, we have

$$\Pr[\mathsf{G}_0 = 1] = \mathbf{Adv}^{\text{SROB-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A})$$

Coming to the game G_4, note that the adversary $\mathcal{A}$ wins the game if it finally outputs a ciphertext c such that $\mathsf{Decap}(\mathsf{sk}'_0, c) \neq \perp$ and $\mathsf{Decap}(\mathsf{sk}'_1, c) \neq \perp$. Because of the modification of the $\mathsf{Decap}(\mathsf{sk}'_i, \cdot)$ oracles, this winning condition translates to $H'_0(c_1) = c_2$ and $H'_1(c_1) = c_2$, or in other words, $H'_0(c_1) = H'_1(c_1)$ where H'_0 and H'_1 are independent quantum-accessible random functions. Note that in this case, (c_1, c_1) is a *claw* w.r.t. the pair of functions $H'_0 : \overline{\mathcal{C}}_1 \rightarrow \overline{\mathcal{C}}_2$ and $H'_1 : \overline{\mathcal{C}}_1 \rightarrow \overline{\mathcal{C}}_2$. Hence we can bound the success probability of $\mathcal{A}$ in G_4 by the advantage of an adversary $\mathcal{C}$ against the *claw-finding* problem w.r.t. the instance (H'_0, H'_1). $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_4.
- Uses a $2q_G$-wise independent function to perfectly simulate the random oracle G, uses three different $2q_H$-wise independent functions to perfectly simulate the random oracles H_0, H_1 and H_2 respectively, and uses a $2q_{H'}$-wise independent function to perfectly simulate the random oracle H'_2 in $\mathcal{A}$'s view, as noted in Lemma 2. Also uses the pair of functions $f_0 : \overline{\mathcal{C}}_1 \rightarrow \overline{\mathcal{C}}_2$ and $f_1 : \overline{\mathcal{C}}_1 \rightarrow \overline{\mathcal{C}}_1$ – which is the instance of the claw-finding problem – to simulate the oracles H'_0 and H'_1 respectively.
- Answers decapsulation queries the same way as in G_4 using the oracles $f_i(\cdot), H_i(\cdot)$ ($i \in \{0, 1\}$).
- After obtaining a final ciphertext c ($= (c_1, c_2)$) from $\mathcal{A}$, forwards (c_1, c_1) as a claw w.r.t. (f_0, f_1).

Note that $\mathcal{C}$ makes at most $q_{H'}$ queries to the pair (f_0, f_1). It is easy to see that $\Pr[\mathsf{G}_4 = 1] \leq \frac{C(q_{H'}+1)^3}{|\overline{\mathcal{C}}_2|}$ from Lemma 1. Hence, we finally get

$$\mathbf{Adv}^{\text{SROB-CCA}}_{\mathsf{KEM}^{\perp}}(\mathcal{A}) \leq q_D \cdot \mathbf{Adv}^{\text{SCFR-CPA}}_{\mathsf{PKE}_1}(\mathcal{B}) + \frac{C(q_{H'}+1)^3}{|\overline{\mathcal{C}}_2|} + \frac{2q_D}{\overline{\mathcal{C}}_2} + 2q_G(q_D + 2)\sqrt{2\delta}$$

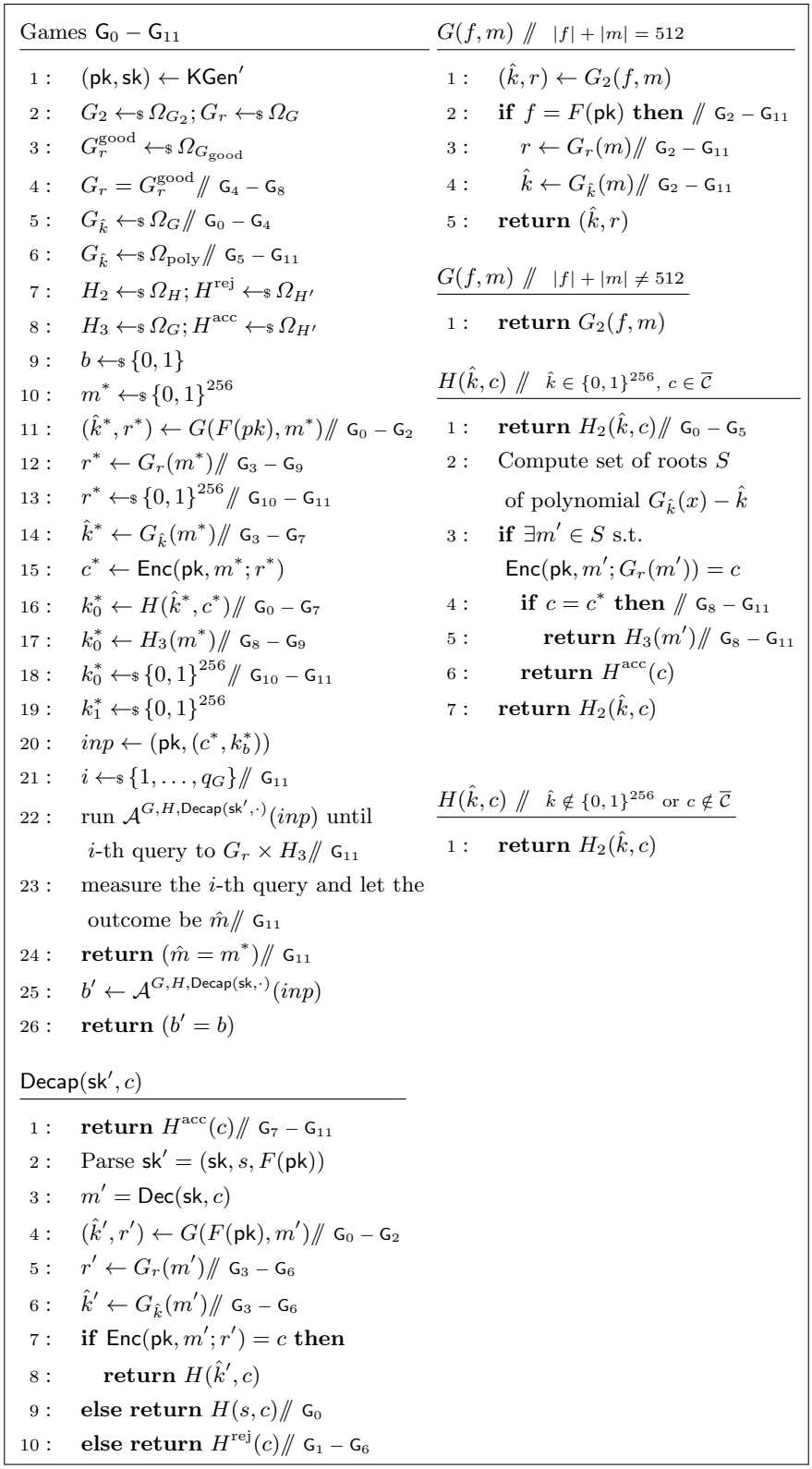
E An IND-CCA Security Proof of proto-Saber in the QROM

We restate the IND-CCA security theorem for proto-Saber in the QROM from [15].

Theorem 16. *Given* $\mathsf{pSaber.PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *is δ-correct, for any* IND-CCA *adversary $\mathcal{A}$ against* $\mathsf{pSaber.KEM} = (\mathsf{KGen}', \mathsf{Encap}, \mathsf{Decap})$ *issuing at most q_D queries to the decapsulation oracles, at most q_G (resp. q_H) queries to the quantum random oracle G (resp. H), there exists an* IND-CPA *adversary $\mathcal{B}$ against* pSaber.PKE *such that*

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{pSaber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q_H}{2^{128}} + 4q_G\sqrt{\delta}$$

and the running time of $\mathcal{B}$ is that of $\mathcal{A}$.



Games $\mathsf{G}_0 - \mathsf{G}_{11}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}'$
$G_2 \leftarrow_\$ \Omega_{G_2}; G_r \leftarrow_\$ \Omega_G$
$G_r^{\text{good}} \leftarrow_\$ \Omega_{G_{\text{good}}}$
$G_r = G_r^{\text{good}} /\!\!/ \ \mathsf{G}_4 - \mathsf{G}_8$
$G_{\hat{k}} \leftarrow_\$ \Omega_G /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_4$
$G_{\hat{k}} \leftarrow_\$ \Omega_{\text{poly}} /\!\!/ \ \mathsf{G}_5 - \mathsf{G}_{11}$
$H_2 \leftarrow_\$ \Omega_H; H^{\text{rej}} \leftarrow_\$ \Omega_{H'}$
$H_3 \leftarrow_\$ \Omega_G; H^{\text{acc}} \leftarrow_\$ \Omega_{H'}$
$b \leftarrow_\$ \{0,1\}$
$m^* \leftarrow_\$ \{0,1\}^{256}$
$(\hat{k}^*, r^*) \leftarrow G(F(pk), m^*) /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
$r^* \leftarrow G_r(m^*) /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_9$
$r^* \leftarrow_\$ \{0,1\}^{256} /\!\!/ \ \mathsf{G}_{10} - \mathsf{G}_{11}$
$\hat{k}^* \leftarrow G_{\hat{k}}(m^*) /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_7$
$c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$
$k_0^* \leftarrow H(\hat{k}^*, c^*) /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_7$
$k_0^* \leftarrow H_3(m^*) /\!\!/ \ \mathsf{G}_8 - \mathsf{G}_9$
$k_0^* \leftarrow_\$ \{0,1\}^{256} /\!\!/ \ \mathsf{G}_{10} - \mathsf{G}_{11}$
$k_1^* \leftarrow_\$ \{0,1\}^{256}$
$inp \leftarrow (\mathsf{pk}, (c^*, k_b^*))$
$i \leftarrow_\$ \{1, \ldots, q_G\} /\!\!/ \ \mathsf{G}_{11}$
run $\mathcal{A}^{G,H,\mathsf{Decap}(\mathsf{sk}',\cdot)}(inp)$ until i-th query to $G_r \times H_3 /\!\!/ \ \mathsf{G}_{11}$
measure the i-th query and let the outcome be $\hat{m} /\!\!/ \ \mathsf{G}_{11}$
return $(\hat{m} = m^*) /\!\!/ \ \mathsf{G}_{11}$
$b' \leftarrow \mathcal{A}^{G,H,\mathsf{Decap}(\mathsf{sk},\cdot)}(inp)$
return $(b' = b)$

$\mathsf{Decap}(\mathsf{sk}', c)$

return $H^{\text{acc}}(c) /\!\!/ \ \mathsf{G}_7 - \mathsf{G}_{11}$
Parse $\mathsf{sk}' = (\mathsf{sk}, s, F(\mathsf{pk}))$
$m' = \mathsf{Dec}(\mathsf{sk}, c)$
$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}), m') /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_2$
$r' \leftarrow G_r(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_6$
$\hat{k}' \leftarrow G_{\hat{k}}(m') /\!\!/ \ \mathsf{G}_3 - \mathsf{G}_6$
if $\mathsf{Enc}(\mathsf{pk}, m'; r') = c$ **then**
  return $H(\hat{k}', c)$
else return $H(s, c) /\!\!/ \ \mathsf{G}_0$
else return $H^{\text{rej}}(c) /\!\!/ \ \mathsf{G}_1 - \mathsf{G}_6$

$G(f, m) \ /\!\!/ \ |f| + |m| = 512$

$(\hat{k}, r) \leftarrow G_2(f, m)$
if $f = F(\mathsf{pk})$ **then** $/\!\!/ \ \mathsf{G}_2 - \mathsf{G}_{11}$
  $r \leftarrow G_r(m) /\!\!/ \ \mathsf{G}_2 - \mathsf{G}_{11}$
  $\hat{k} \leftarrow G_{\hat{k}}(m) /\!\!/ \ \mathsf{G}_2 - \mathsf{G}_{11}$
return $(\hat{k}, r)$

$G(f, m) \ /\!\!/ \ |f| + |m| \neq 512$

return $G_2(f, m)$

$H(\hat{k}, c) \ /\!\!/ \ \hat{k} \in \{0,1\}^{256}, \ c \in \overline{\mathcal{C}}$

return $H_2(\hat{k}, c) /\!\!/ \ \mathsf{G}_0 - \mathsf{G}_5$
Compute set of roots S of polynomial $G_{\hat{k}}(x) - \hat{k}$
if $\exists m' \in S$ s.t. $\mathsf{Enc}(\mathsf{pk}, m'; G_r(m')) = c$
  if $c = c^*$ **then** $/\!\!/ \ \mathsf{G}_8 - \mathsf{G}_{11}$
    return $H_3(m') /\!\!/ \ \mathsf{G}_8 - \mathsf{G}_{11}$
  return $H^{\text{acc}}(c)$
return $H_2(\hat{k}, c)$

$H(\hat{k}, c) \ /\!\!/ \ \hat{k} \notin \{0,1\}^{256}$ or $c \notin \overline{\mathcal{C}}$

return $H_2(\hat{k}, c)$

Fig. 32. Games G_0 – G_{11} for the proof of Theorem 16.

The proof that follows is structurally similar to that of [27, Theorem 1]. But the key component of this proof that overcomes the barrier described in Subsection 5.2 is encapsulated in the "$\mathsf{G}_5 \to \mathsf{G}_8$" game-hops.

Proof. Denote Ω_{G_2}, Ω_G, Ω_H and $\Omega_{H'}$ to be the set of all functions $G_2 : \{0,1\}^* \to \{0,1\}^{512}$, $G : \{0,1\}^{256} \to \{0,1\}^{256}$, $H : \{0,1\}^* \to \{0,1\}^{256}$ and $H' : \overline{\mathcal{C}} \to \{0,1\}^{256}$ respectively, where $\overline{\mathcal{C}}$ is the ciphertext space of $\mathsf{pSaber.PKE}/\mathsf{pSaber.KEM}$.

Let $\mathcal{A}$ be an adversary in the IND-CCA game for $\mathsf{pSaber.KEM}$ issuing at most q_D (classical) queries to the oracle $\mathsf{Decap}(\mathsf{sk}, \cdot)$, and q_G (resp., q_H) quantum queries to the random oracles G (resp. H). Consider the sequence of games $\mathsf{G}_0 - \mathsf{G}_{11}$ described in Figure 32.

Game G_0 The game G_0 is exactly the IND-CCA game for $\mathsf{pSaber.KEM}$. Hence,

$$\left| \Pr[\mathsf{G}_0 = 1] - \frac{1}{2} \right| = \mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}).$$

Game G_1 In game G_1, we modify the decapsulation oracle $\mathsf{Decap}(\mathsf{sk}', \cdot)$ such that $H^{\text{rej}}(c)$ is returned instead of $H(s, c)$ for an invalid ciphertext c. Since this change is quite similar to the game-hop "$\mathsf{G}_0 \to \mathsf{G}_1$" in the proof of [27, Theorem 1], it is not hard to obtain

$$|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_0 = 1]| \leq \frac{2q_H}{\sqrt{2^{256}}}$$

(note that the message space of $\mathsf{pSaber.PKE}$ is $\{0,1\}^{256}$).

Game G_2 In game G_2, we implicitly divide the G-queries into two categories: (1) query is of the form (f, m) with $|f| + |m| = 512$ and $f = F(\mathsf{pk})$ and (2) the remaining queries. We then respond to the queries from the respective categories with $(G_{\hat{k}}(m), G_r(m))$ and $G_2(m, c)$ respectively, where $G_{\hat{k}}$, G_r are internal random functions. It is not hard to verify that the output distributions of the G-oracle in games G_1 and G_2 are equivalent. Therefore,

$$\Pr[\mathsf{G}_2 = 1] = \Pr[\mathsf{G}_1 = 1].$$

Game G_3 In game G_3, we make the following changes w.r.t. the G-oracle evaluation. First, we generate the values $\hat{k}^*, r^*$ in setup of the game as "$\hat{k}^* \leftarrow G_{\hat{k}}(m^*)$" and "$r^* \leftarrow G_r(m^*)$" (effectively, replacing the step "$(\hat{k}^*, r^*) \leftarrow G(F(\mathsf{pk}), m^*)$" in G_2). We then similarly generate the values $\hat{k}', r'$ w.r.t. the decapsulation oracle $\mathsf{Decap}(\mathsf{sk}', \cdot)$ as "$\hat{k}' \leftarrow G_{\hat{k}}(m')$" and "$r' \leftarrow G_r(m')$" (replacing the step "$(\hat{k}', r') \leftarrow G(F(\mathsf{pk}), m')$" in G_2).

Since these changes are "cosmetic" in nature following our modification to oracle G in game G_2, we have

$$\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_2 = 1].$$

Game G_4 In game G_4, we change the random oracle G_r such that it uniformly samples "good" random coins w.r.t. the key-pair $(\mathsf{pk}, \mathsf{sk})$. To be specific, given a PKE key-pair $(\mathsf{pk}, \mathsf{sk})$ and a message $m \in \mathcal{M}$, define

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

Denote $\Omega_{G_r^{\text{good}}}$ to be the set of all random functions G_r^{good} such that $G_r^{\text{good}}(m)$ is sampled according to a uniform distribution in $\mathcal{R}_{\text{good}}(\mathsf{pk}, \mathsf{sk}, m)$. Hence in G_4, we replace the oracle G_r with G_r^{good}. Again, this change is quite similar to the game-hop "$\mathsf{G}_1 \to \mathsf{G}_2$" in the proof of [27, Theorem 1]. Hence, it is not hard to obtain

$$|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq 2q_G\sqrt{\delta}.$$

Game G_5 In game G_5, we replace the random oracle $G_{\hat{k}}$ with a $2q_G$-wise independent function, following Lemma 2. Random polynomials of degree $2q_G - 1$ over the finite field representation of the message space $\{0,1\}^{256}$ are $2q_G$-wise independent. Let Ω_{poly} be the set of all such polynomials. We are then replacing the step "$G_{\hat{k}} \leftarrow_{\$} \Omega_G$" with "$G_{\hat{k}} \leftarrow_{\$} \Omega_{\text{poly}}$" in G_5. From Lemma 2, as this change is indistinguishable when the oracle $G_{\hat{k}}$ is queried at most q_G times, we have

$$\Pr[\mathsf{G}_5 = 1] = \Pr[\mathsf{G}_4 = 1].$$

Game G_6 In game G_6, we implicitly divide the H-queries into two disjoint categories: (1) query is of the form $(\hat{k}, c)$ with $\hat{k} \in \{0,1\}^{256}$, $c \in \overline{\mathcal{C}}$ and there exists $m \in \{0,1\}^{256}$ which is a root of the polynomial

$G_{\hat{k}}(x) - \hat{k}$ (recall that $G_{\hat{k}}$ is now a polynomial) such that $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c$ and (2) the remaining queries. We then respond to queries from the respective categories with $H^{\mathrm{acc}}(c)$ and $H_2(\hat{k}, c)$, where H^{acc} is an internal random function not directly accessible to the adversary $\mathcal{A}$.

Focusing on H-queries in "category (1)", note that it is not possible for two distinct queries $(\hat{k}', c)$ and $(\hat{k}'', c)$ to result in the same output $H^{\mathrm{acc}}(c)$. The reason is, as G_r now samples "good" random coins, there can exist at most one value m that satisfies $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c$. And since $G_{\hat{k}}(\cdot)$ is a deterministic function, the above follows. Therefore, the output distributions of the H-oracle in the games G_5 and G_6 are equivalent, and we get

$$\Pr[\mathsf{G}_6 = 1] = \Pr[\mathsf{G}_5 = 1].$$

Game G_7 In game G_7, we change the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle such that there is no need for the secret key sk'. Namely, $H^{\mathrm{acc}}(c)$ is returned for the decapsulation of any ciphertext c w.r.t. sk'. Let $m' = \mathsf{Dec}(\mathsf{sk}, c)$, $r' = G_r(m')$ and $\hat{k}' = G_{\hat{k}}(m')$. Now consider the following two cases:

1. $\mathsf{Enc}(\mathsf{pk}, m'; r') = c$. In this case, the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle returns $H(\hat{k}', c)$ in game G_6 and $H^{\mathrm{acc}}(c)$ in game G_7. It is not hard to see that we have $H(\hat{k}', c) = H^{\mathrm{acc}}(c)$ in G_6, since the query $(\hat{k}', c)$ falls under "category (1)" w.r.t. oracle H. Therefore, $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracles of games G_6 and G_7 return the same value $H^{\mathrm{acc}}(c)$.
2. $\mathsf{Enc}(\mathsf{pk}, m'; r') \neq c$. In this case, the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle returns $H^{\mathrm{rej}}(c)$ in game G_6 and $H^{\mathrm{acc}}(c)$ in game G_7. In game G_6, as the random function H^{rej} is independent of all other oracles, the output $H^{\mathrm{rej}}(c)$ is uniformly random in the adversary $\mathcal{A}$'s view. In game G_7, the only way $\mathcal{A}$ gets prior access to the value $H^{\mathrm{acc}}(c)$ is if it made a H-query $(\hat{k}'', c)$ such that $\mathsf{Enc}(\mathsf{pk}, m''; G_r(m'')) = c$ (and $G_{\hat{k}}(m'') = \hat{k}''$). But since G_r samples "good" random coins, we have $\mathsf{Dec}(\mathsf{sk}, c) = m'' = m'$ leading to a contradiction of "$\mathsf{Enc}(\mathsf{pk}, m'; r') \neq c$". Therefore, such a prior access is not possible and $H^{\mathrm{acc}}(c)$ will also be a uniformly random value in $\mathcal{A}$'s view.

As the output distributions of the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle in G_6 and G_7 are the same in both cases, we have

$$\Pr[\mathsf{G}_7 = 1] = \Pr[\mathsf{G}_6 = 1].$$

Game G_8 In game G_8, we make a further modification to the evaluation of "category (1)" H-queries of the form $(\hat{k}, c^*)$ as follows, where c^* is the challenge ciphertext computed in the setup: respond to the corresponding "category (1)" query with $H_3(m)$, where m is a (lexicographically minimal) root of polynomial $G_{\hat{k}}(x) - \hat{k}$ that satisfies $\mathsf{Enc}(\mathsf{pk}, m; G_r(m)) = c^*$. Here H_3 is another internal independent random function.

Since we established in the "$\mathsf{G}_5 \to \mathsf{G}_6$" game-hop that there cannot be two distinct "category (1)" H-queries $(\hat{k}^*, c^*)$ and $(\hat{k}', c^*)$, this further change to the H-oracle only affects the H-query $(\hat{k}^*, c^*)$, where $\hat{k}^* = G_{\hat{k}}(m^*)$ for the secret message m^* sampled uniformly at random in the setup (and $\mathsf{Enc}(\mathsf{pk}, m^*; G_r(m^*)) = c^*$). W.r.t. this query, the H oracle would return $H^{\mathrm{acc}}(c^*)$ in G_7, and $H_3(m^*)$ in G_8. The adversary $\mathcal{A}$'s view would be identical even after this change because the random value $H^{\mathrm{acc}}(c^*)$ is only accessible to $\mathcal{A}$ via the H-oracle in G_7, and in particular, not through the $\mathsf{Decap}(\mathsf{sk}', \cdot)$ oracle since c^* is a forbidden decapsulation query. Hence in G_8, we are effectively replacing a uniformly random value that can only be accessed via the H-oracle by $\mathcal{A}$ with another uniformly random value. Hence, the output distributions of the H-oracle in the games G_7 and G_8 are equivalent. Therefore, we have

$$\Pr[\mathsf{G}_8 = 1] = \Pr[\mathsf{G}_7 = 1].$$

Following the above modification, we make a "cosmetic" change in the setup where the "real" key k_0^* defined in the setup is now generated as "$k_0^* \leftarrow H_3(m^*)$" (instead of "$k_0^* \leftarrow H(\hat{k}^*, c^*)$"). This change does not affect the game in any way.

Game G_9 In game G_9, we reset the random oracle G_r so that it returns uniformly random coins from $\{0,1\}^{256}$ instead of returning only "good" random coins. Since this change, in a sense, is the "inverse" of the game-hop "$\mathsf{G}_3 \to \mathsf{G}_4$", by using a similar analysis, we obtain

$$|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_8 = 1]| \leq 2q_G\sqrt{\delta}.$$

Game G_{10} In the set-up of game G_{10}, we generate the values r^* and k_0^* such that they are uniformly random values independent of any oracles, i.e., we replace the step "$r^* \leftarrow G_r(m^*)$" with "$r^* \leftarrow_{\$} \{0,1\}^{256}$"

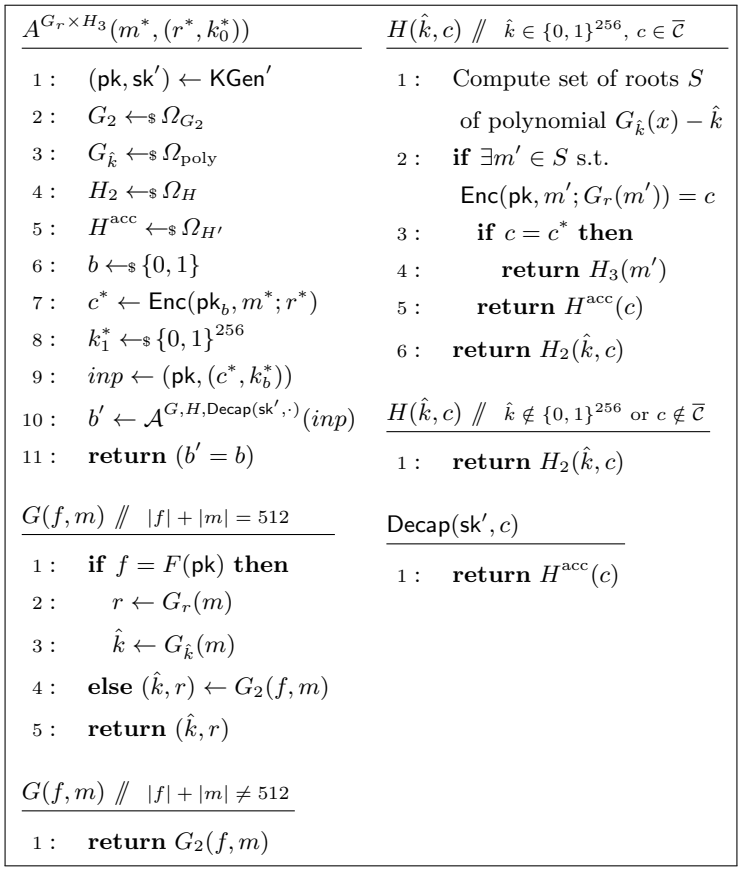


Fig. 33. Algorithm $A^{G_r \times H_3}$ for the proof of Theorem 16.

and "$k_0^* \leftarrow H_3(m^*)$" with "$k_0^* \leftarrow_{\$} \{0,1\}^{256}$". Note that in this game, both the "real" and "random" keys are sampled uniformly at random from $\{0,1\}^{256}$ (i.e., both keys have the exact same distribution). Hence, the challenge bit b is independent from $\mathcal{A}$'s view and we get

$$\Pr[\mathsf{G}_{10} = 1] = \frac{1}{2}.$$

Now we use Lemma 4 to bound the difference in the success probabilities of $\mathcal{A}$ in G_9 and G_{10}. Let A be an oracle algorithm that has quantum access to the random oracle $G_r \times H_3$, where $G_r, H_3 \leftarrow_{\$} \Omega_G$ and $(G_r \times H_3)(m) = (G_r(m), H_3(m))$. Figure 33 describes $A^{G_r \times H_3}$'s operation on input $(m^*, (r^*, k_0^*))$. Note that the algorithm $A^{G_r \times H_3}$ makes at most $q_G + q_H$ number of queries to the random oracle $G_r \times H_3$ to respond to $\mathcal{A}$'s G-oracle and H-oracle queries.[10].

Let B be an oracle algorithm that on input m^* does the following: picks $i \leftarrow_{\$} \{1, \ldots, q_G + q_H\}$, generates $r^* \leftarrow_{\$} \{0,1\}^{256}$ and $k_0^* \leftarrow_{\$} \{0,1\}^{256}$, runs the algorithm $A^{G_r \times H_3}(m^*, (r^*, k_0^*))$ until the i-th query, measures the argument of the $(G_r \times H_3)$-query in the computational basis and outputs the measurement outcome (if $A^{G_r \times H_3}$ makes less than i queries, B outputs $\perp$). With this construction of A, note that $P_A^1 = \Pr[\mathsf{G}_9 = 1]$ and $P_A^2 = \Pr[\mathsf{G}_{10} = 1]$, where P_A^1 and P_A^2 are as defined in Lemma 4 w.r.t. the algorithm $A^{G_r \times H_3}$. Therefore, we now define game G_{11} (see Fig. 32) such that $P_B = \Pr[\mathsf{G}_{11} = 1]$, where P_B is as defined in Lemma 4 w.r.t. the algorithm $B^{G_r \times H_3}$. From Lemma 4, we thus have

$$|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_{10} = 1]| \leq 2(q_G + q_H)\sqrt{\Pr[\mathsf{G}_{11} = 1]}$$

We now bound the success probability of $\mathcal{A}$ in G_{11} by the advantage of an adversary $\mathcal{C}$ in the OW-CPA game of $\mathsf{pSaber.PKE}$. Upon receiving a public-key pk along with a ciphertext c^*, where $c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*; r^*)$ for uniformly random (secret) message $m^*(\leftarrow_{\$} \{0,1\}^{256})$ and randomness $r^*(\leftarrow_{\$} \{0,1\}^{256})$ chosen by the challenger, $\mathcal{C}$ proceeds as follows:

- Runs $\mathcal{A}$ as a subroutine as in game G_{11} (e.g., starting with sampling a uniformly random bit $b \leftarrow_{\$} \{0,1\}$).
- Uses three different $2q_G$-wise independent functions to perfectly simulate the random oracles G_2, G_r, and $G_{\hat{k}}$ respectively, three different $2q_H$-wise independent functions to simulate the random oracles H^{acc}, H_2 and H_3 respectively in $\mathcal{A}$'s view, as noted in Lemma 2. Also evaluates $\mathcal{A}$'s G- and H-queries using the oracle $G_r \times H_3$; the random oracles G and H are simulated in the same way as in G_{11}.
- Answers decapsulation queries using the oracle H^{acc} as in G_{11}.
- For $\mathcal{A}$'s challenge query, samples a uniformly random key $k^* \leftarrow_{\$} \{0,1\}^{256}$ and responds with $(\mathsf{pk}, (c^*, k^*))$.
- Selects $i \leftarrow_{\$} \{1, \ldots, q_G + q_H\}$, measures the i-th query to oracle $G_r \times H_3$ and returns the outcome $\hat{m}$.

Again, it is not hard to see that $\Pr[\mathsf{G}_{11} = 1] \leq \mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{pSaber.PKE}}(\mathcal{C})$. Since we know that IND-CPA security of a PKE scheme with a sufficiently large message space also implies its OW-CPA security, corresponding to adversary $\mathcal{C}$, there exists an IND-CPA adversary $\mathcal{B}$ against $\mathsf{pSaber.PKE}$ such that

$$\mathbf{Adv}^{\text{OW-CPA}}_{\mathsf{pSaber.PKE}}(\mathcal{C}) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{pSaber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}$$

where the running time of $\mathcal{B}$ is that of $\mathcal{C}$, and $\frac{1}{2^{256}}$ is the message space of $\mathsf{pSaber.PKE}$.

Hence by collecting all of the above bounds, we finally arrive at

$$\mathbf{Adv}^{\text{IND-CCA}}_{\mathsf{pSaber.KEM}}(\mathcal{A}) \leq 2(q_G + q_H)\sqrt{\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{pSaber.PKE}}(\mathcal{B}) + \frac{1}{2^{256}}} + \frac{2q_H}{2^{128}} + 4q_G\sqrt{\delta}$$

[10] For example, if $A^{G_r \times H_3}$ wants to respond to $\mathcal{A}$'s H-query, then $A^{G_r \times H_3}$ prepares a uniform superposition of all states in the output register corresponding to G_r (see [38] for particulars of this "trick").