IND-CCA-secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited*

Haodong Jiang[1,2], Zhenfeng Zhang[2,3], Long Chen[2,3], Hong Wang[1], and Zhi Ma[1,4]

[1] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou, Henan, China
[2] TCA Laboratory, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, Beijing, China
[4] CAS Center for Excellence and Synergetic Innovation Center in Quantum information and Quantum Physics,USTC, Hefei, Anhui, China
hdjiang13@gmail.com, {chenlong,zfzhang}@tca.iscas.ac.cn,
{wfallmoon,ma_zhi}@163.com

Abstract. With the gradual progress of NIST's post-quantum cryptography standardization, the Round-1 KEM proposals have been posted for public to discuss and evaluate. Among the IND-CCA-secure KEM constructions, mostly, an IND-CPA-secure (or OW-CPA-secure) public-key encryption (PKE) scheme is first introduced, then some generic transformations are applied to it. All these generic transformations are constructed in the random oracle model (ROM). To fully assess the post-quantum security, security analysis in the quantum random oracle model (QROM) is preferred. However, current works either lacked a QROM security proof or just followed Targhi and Unruh's proof technique (TCC-B 2016) and modified the original transformations by adding an additional hash to the ciphertext to achieve the QROM security.

In this paper, by using a novel proof technique, we present QROM security reductions for two widely used generic transformations without suffering any ciphertext overhead. Meanwhile, the security bounds are much tighter than the ones derived by utilizing Targhi and Unruh's proof technique. Thus, our QROM security proofs not only provide a solid post-quantum security guarantee for NIST Round-1 KEM schemes, but also simplify the constructions and reduce the ciphertext sizes. We also provide QROM security reductions for Hofheinz-Hövelmanns-Kiltz modular transformations (TCC 2017), which can help to obtain a variety of combined transformations with different requirements and properties.

Keywords: quantum random oracle model · key encapsulation mechanism · IND-CCA security · generic transformation

* An earlier version of this paper appeared with title "Post-quantum IND-CCA-secure KEM without Additional Hash".

1 Introduction

As a foundational cryptography primitive, key encapsulation mechanism (KEM) is efficient and versatile. It can be used to construct, in a black-box manner, PKE (the KEM-DEM paradigm [1]), key exchange and authenticated key exchange [2, 3]. Compared with designing a full PKE scheme, the KEM construction is usually somewhat easier or more efficient. In December 2016, National Institute of Standards and Technology (NIST) announced a competition with the goal to standardize post-quantum cryptographic (PQC) algorithms including digital-signature, public-key encryption (PKE), and KEM (or key exchange) with security against quantum adversaries [4]. Among the 69 Round-1 algorithm submissions, posted in December 2017 by NIST for public to discuss and evaluate [4], there are 39 proposals for KEM constructions.

Indistinguishability against chosen-ciphertext attacks (IND-CCA) [5] is widely accepted as a standard security notion for many cryptography applications. However, the security is usually much more difficult to prove than IND-CPA (and OW-CPA) security, i.e., indistinguishability (and one-way) against chosen-plaintext attacks. Mostly, generic transformations [6, 7] are used to create an IND-CCA-secure KEM from some weakly secure (OW-CPA or IND-CPA) PKEs.

Recently, considering the drawbacks of previous analysis of Fujisaki-Okamoto (FO) transformation [8, 9], such as a non-tight security reduction and the need for a perfectly correct scheme, Hofheinz, Hövelmanns and Kiltz [7] revisited the KEM version of FO transformation [6] and provided a fine-grained and modular toolkit of transformations $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}^{\not\perp}_{m}$, $\mathrm{U}^{\perp}_{m}$, $\mathrm{QU}^{\not\perp}_{m}$ and $\mathrm{QU}^{\perp}_{m}$ (In what follows, these transformations will be categorized as modular FO transformations for brevity), where m (without m) means $K = H(m)$ $(K = H(m, c))$, $\not\perp$ $(\perp)$ means implicit (explicit) rejection[5] and Q means adding an additional hash to the ciphertext. Combing these modular transformations, they obtained several variants of FO transformation $\mathrm{FO}^{\not\perp}$, $\mathrm{FO}^{\perp}$, $\mathrm{FO}^{\not\perp}_{m}$, $\mathrm{FO}^{\perp}_{m}$, $\mathrm{QFO}^{\not\perp}_{m}$ and $\mathrm{QFO}^{\perp}_{m}$ (These transformations will be categorized as FO transformations in the following).

All the (modular) FO transformations are in the random oracle model (ROM) [10]. When the KEM scheme is instantiated, the random oracle is usually replaced by a hash function, which a quantum adversary may evaluate on a quantum superposition of inputs. As a result, to fully assess post-quantum security, we should analyze security in the quantum random oracle model (QROM), as introduced in [11]. However, proving security in the QROM is quite challenging, as many classical ROM proof techniques will be invalid [11].

In [7], Hofheinz et al. presented QROM security reductions for $\mathrm{QU}^{\not\perp}_{m}$, $\mathrm{QU}^{\perp}_{m}$, $\mathrm{QFO}^{\not\perp}_{m}$ and $\mathrm{QFO}^{\perp}_{m}$. For these transformations, there is an additional hash in the ciphertext, which plays an important role in their reductions. The security reductions for $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}^{\not\perp}_{m}$, $\mathrm{U}^{\perp}_{m}$, $\mathrm{FO}^{\not\perp}$, $\mathrm{FO}^{\perp}$, $\mathrm{FO}^{\not\perp}_{m}$ and $\mathrm{FO}^{\perp}_{m}$ are just presented in the ROM.

[5] In implicit (explicit) rejection, a pseudorandom key (an abnormal symbol $\perp$) is returned for an invalid ciphertext.

Among the 39 KEM submissions, there are 35 schemes that take IND-CCA as the security goal. Particularly, 25 IND-CCA-secure KEM schemes are constructed by utilizing above transformations (see Table 1) from different PKE schemes, with different security notions (e.g., IND-CPA vs OW-CPA), and underlying hardness of certain problems over lattice, code theory and isogeny. In the submissions of LAC, Odd Manhattan, LEDAkem and SIKE, the QROM security is not considered. In the 16 submissions including FrodoKEM etc., $\mathrm{QFO}^{\not\perp}$[6], $\mathrm{QFO}^{\perp}$, $\mathrm{QFO}_m^{\not\perp}$ and $\mathrm{QFO}_m^{\perp}$ are used, where an additional hash is appended to the ciphertext. In the other 5 submissions including CRYSTALS-Kyber, LIMA, SABER, ThreeBears and Classic McEliece, the additional hash is removed according to Saito, Xagawa, and Yamakawa's work [12] and our work in previous version [13].

For the (modular) FO transformations, the underlying PKE schemes differ in the following aspects including additional hash, correctness, determinacy, and security.

- **Additional hash.** Additional hash here is a length-preserving hash function (that has the same domain and range size) appended to the ciphertext, which was first introduced by Targhi and Unruh [14] to prove the QROM security of the variants of FO transformation [8, 9] and OAEP transformation [15, 16]. Following Targhi and Unruh's trick, Hofheinz et al. gave the transformations $\mathrm{QU}_m^{\not\perp}$, $\mathrm{QU}_m^{\perp}$, $\mathrm{QFO}_m^{\not\perp}$ and $\mathrm{QFO}_m^{\perp}$ by adding an additional hash to the corresponding ROM constructions, and presented the QROM security reductions for them.
  Among NIST Round-1 submissions of an IND-CCA-secure KEM, 16 proposals use this trick to achieve QROM security. Intuitively, for 128-bit post-quantum security, this additional hash merely increases the ciphertext size by 256 bits [17]. However, we note that the QROM security proof in [7, 14] requires the additional hash to be length-preserving. Thus, for some schemes where the message space is strictly larger than the output space of the hash function, the increasement of ciphertext size is significant. Hülsing et al. [18] tried several ways to circumvent this issue, unfortunately all straight forward approaches failed. For their specific NTRU-based KEM, additional 1128 bits are needed, which accounts for 11% of the final encapsulation size.
  In the ROM, this additional hash is clearly redundant for the constructions of an IND-CCA-secure KEM [6, 7]. Some proposals, e.g., ThreeBears [19], believe this additional hash adds no security. To accomplish the QROM security proof, this additional hash was deliberately introduced, which increased the ciphertext size and complicated the implementation. Thus, a natural question is that: can we improve the QROM security proofs without suffering any ciphertext overhead for these constructions?
- **Correctness error.** For many practical post-quantum PKE schemes, e.g., DXL [20], Peikert [21], BCNS [22], New hope [23], Frodo [24], Lizard [25], Kyber [26], NTRUEncrypt [27], NTRU Prime [28], CAKE [29] and QC-MDPC

[6] $\mathrm{QFO}^{\perp}$ ($\mathrm{QFO}^{\not\perp}$) is the same as $\mathrm{QFO}_m^{\perp}$ ($\mathrm{QFO}_m^{\not\perp}$) except that $K = H(m, c)$. Its security proof can be easily obtained from the one for $\mathrm{QFO}_m^{\perp}$ ($\mathrm{QFO}_m^{\not\perp}$) in [7].

  [30], there exists a small correctness error δ, i.e., the probability of decryption failure in a legitimate execution of the scheme. Specially, among the KEM submissions in Table 1, there are 18 proposals that have a correctness error issue.

  From a security point of view, it turns out that correctness errors not only influence the validity of a security proof, but also leak information on the private key [31]. Particularly, the chosen-ciphertext attacks by exploiting the gathered correctness errors [31, 32] were demonstrated for CCA versions of NTRUEncrypt and QC-MDPC obtained by using generic transformations, whose securities were proved assuming the underlying PKEs perfectly correct. Additionally, recently, Bernstein et al. [33] showed that the HILA5 KEM [34] does not provide IND-CCA security by demonstrating a key-recovery attack in the standard IND-CCA attack model using the information obtained from the correctness errors.

  To date, it is not clear how highly these correctness errors can affect the CCA security of these KEM schemes and how high these correctness errors should be to achieve a fixed security strength. To the best of our knowledge, for all previous security analyses about (modular) FO transformations except the work [7], perfect correctness, i.e., $\delta = 0$, is assumed. Therefore, QROM security analyses of above (modular) FO transformations with correctness errors into consideration are preferred.

- **Determinacy.** According to the works [7, 35], an IND-CCA-secure KEM in the ROM can be easily constructed by applying the transformation $\mathrm{U}_m^{\perp}$ (or $\mathrm{U}_m^{\not\perp}$) to a deterministic PKE (DPKE). Saito et al. [12] showed that a DPKE can be constructed based on the concepts of the GPV trapdoor function for LWE [36], NTRU [27], the McEliece PKE [37], and the Niederreiter PKE [38]. However, the popular LWE cryptosystem and variants [39–42] are probabilistic encryption, which are referred by CRYSTALS-Kyber, EMBLEM and R.EMBLEM, FrodoKEM, KINDI, LAC, Lepton, LIMA, Lizard, NewHope, Round2, SABER and ThreeBears [4]. Particularly, of the underlying PKEs in the KEM proposals in Table 1, DPKEs just account for 28%.
- **Security notion.** IND-CPA security and OW-CPA security are widely accepted as standard security notions for PKE. In the KEM submissions in Table 1, all the underlying PKE schemes satisfy the OW-CPA security. The IND-CPA security is taken as a security goal of a PKE/KEM scheme during NIST's PQC standardization, and satisfied for most latticed-based and isogeny-based PKE schemes. FO transformations are widely used as they just require the PKE schemes to have the standard CPA security.

  There are also some non-standard security notions, e.g., one-way against plaintext checking attacks (OW-PCA), one-way against validity checking attacks (OW-VA), one-way against plaintext and validity checking attacks (OW-PVCA) for PKE [6, 7] and disjoint simulatability (DS) for DPKE [12]. According to [7, 12], if the underlying PKE satisfies these non-standard securities, modular FO transformations can be used to construct an IND-CCA-secure KEM with a tighter security reduction. Particularly, saito et al. [12] presented a tight security proof for $\mathrm{U}_m^{\not\perp}$ with stronger assumptions for underlying DPKE scheme, DS security and perfect correctness, which are satisfied by Classical McEliece in Table 1.

To accurately evaluate the CCA security of the KEM proposals in Table 1 in the QROM, taking correctness error into account, we revisit the QROM security of above (modular) FO transformations without additional hash and with different assumptions for the underlying PKE scheme in terms of determinacy and security.

1.1 Our Contributions

1. For any correctness error δ $(0 \leq \delta < 1)$, we prove the QROM security of two generic transformations, $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$ in [7], by reducing the standard OW-CPA security of the underlying PKE to the IND-CCA security of KEM, see Table 2.
The obtained security bounds are both $\epsilon' \approx q\sqrt{\delta} + q\sqrt{\epsilon}$, where ϵ' is the success probability of an adversary against the IND-CCA security of the resulting KEM, ϵ is the success probability of another adversary against the OW-CPA security of the underlying PKE, and q is the total number of $\mathcal{B}$'s queries to various oracles. Our security bounds are much better than $\epsilon' \approx q\sqrt{q^2\delta + q\sqrt{\epsilon}}$, achieved by [7]. Meanwhile, the additional hash is not required as it is redundant for our security proofs. In [12], saito et al. also obtained a same tight security bound $\epsilon' \approx q\sqrt{\epsilon}$ for a variant of $\mathrm{FO}_m^{\not\perp}$, $\mathrm{FO}'^{\not\perp}_m = \mathrm{TPunc} \circ \mathrm{U}_m^{\not\perp}$[7], by assuming the underlying PKE scheme IND-CPA-secure and perfectly correct (i.e., $\delta = 0$).
With our tighter QROM security proofs, 16 KEM constructions including FrodoKEM etc., where $\mathrm{QFO}^{\not\perp}$, $\mathrm{QFO}^{\perp}$, $\mathrm{QFO}_m^{\not\perp}$ and $\mathrm{QFO}_m^{\perp}$ are used, can be simplified by cutting off the additional hash and improved in performance with respect to speed and sizes. Additionally, although LAC and SIKE are constructed by using $\mathrm{FO}^{\not\perp}$ without the additional hash, the QROM security proof is not considered in their proposals. Thus, our proofs also provide a solid post-quantum security guarantee for these two KEM schemes without any additional ciphertext overhead.
2. For modular FO transformations including $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}_m^{\not\perp}$ and $\mathrm{U}_m^{\perp}$ in [7], we provide QROM security reductions without additional hash for any correctness error δ $(0 \leq \delta < 1)$, see Table 3.
Specifically, we first define the quantum version of OW-PCA and OW-PVCA by one-way against quantum plaintext checking attacks (OW-qPCA) and one-way against quantum plaintext and (classical) validity checking attacks (OW-qPVCA) (quantum plaintext checking attacks mean that the adversary can make quantum queries to the plaintext checking oracle). For any correctness error δ $(0 \leq \delta < 1)$, we provide QROM security reductions for, $\mathrm{U}^{\not\perp}$ from OW-qPCA, $\mathrm{U}^{\perp}$ from OW-qPVCA, $\mathrm{U}_m^{\not\perp}$ from OW-CPA (and DS), $\mathrm{U}_m^{\perp}$ from OW-VA, to IND-CCA without additional hash.

[7] TPunc is a variant of T in [7]

Table 1: List of KEM submissions based on (modular) FO transformations.

Proposals	Transformations	Correctness error	DPKE?	QROM consideration?
CRYSTALS-Kyber	$\mathsf{FO}^{\not\perp}$	Y	N	Y
EMBLEM and R.EMBLEM	$\mathsf{QFO}^{\perp}$	Y	N	Y
FrodoKEM	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
KINDI	$\mathsf{QFO}^{\not\perp}_m$	Y	N	Y
LAC	$\mathsf{FO}^{\not\perp}$	Y	N	N
Lepton	$\mathsf{QFO}^{\perp}$	Y	N	Y
LIMA	$\mathsf{FO}^{\perp}_m$	N[a]	N	Y
Lizard	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
NewHope	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
NTRU-HRSS-KEM	$\mathsf{QFO}^{\perp}_m$	N	N	Y
Odd Manhattan	$\mathsf{U}^{\perp}_m$	N	N	N
OKCN-AKCN-CNKE	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
Round2	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
SABER	$\mathsf{FO}^{\not\perp}$	Y	N	Y
ThreeBears	$\mathsf{FO}^{\perp}_m$	Y	N	Y
Titanium	$\mathsf{QFO}^{\not\perp}$	Y	N	Y
BIG QUAKE	$\mathsf{QFO}^{\perp}$	N	N	Y
Classic McEliece	$\mathsf{U}^{\not\perp}$	N	Y	Y
DAGS	$\mathsf{QFO}^{\perp}_m$	N	N	Y
HQC	$\mathsf{QFO}^{\perp}$	Y	N	Y
LEDAkem	$\mathsf{U}^{\not\perp}_m$	Y	Y	N
LOCKER	$\mathsf{QFO}^{\perp}$	Y	N	Y
QC-MDPC	$\mathsf{QFO}^{\perp}_m$	Y	N	Y
RQC	$\mathsf{QFO}^{\perp}$	N	N	Y
SIKE	$\mathsf{FO}^{\not\perp}$	N	N	N

[a] In the round-1 submission, the LIMA team uses rejection sampling in encryption to avoid correctness errors. But they claim that they will replace the rejection sampling in encryption with a "standard" analysis of correctness errors to fix a mistake in previous analysis if LIMA survives until the second round [43].

Table 2: FO transformations from standard security assumptions.

Transformation	Underlying security	Security bound	Additional hash	Perfectly correct?
$\mathrm{QFO}_m^{\not\perp}$ and $\mathrm{QFO}_m^{\perp}$ [7]	OW-CPA	$q\sqrt{q^2\delta + q\sqrt{\epsilon}}$	Y	N
$FO'^{\not\perp}_m$ [12]	IND-CPA	$q\sqrt{\epsilon}$	N	Y
$\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$ Our work	OW-CPA	$q\sqrt{\delta} + q\sqrt{\epsilon}$	N	N

Table 3: Modular FO transformations from non-standard security assumptions.

Transformation	Underlying security	Security bound	Additional hash	DPKE	Perfectly correct?
$\mathrm{QU}_m^{\perp}$ [7]	OW-PCA	$q\sqrt{\epsilon}$	Y	N	N
$\mathrm{QU}_m^{\not\perp}$ [7]	OW-PCA	$q\sqrt{\epsilon}$	Y	N	N
$\mathrm{U}_m^{\not\perp}$[12]	DS	ϵ	N	Y	Y
$\mathrm{U}^{\not\perp}$ Our work	OW-qPCA	$q\sqrt{\epsilon}$	N	N	N
$\mathrm{U}^{\perp}$ Our work	OW-qPVCA	$q\sqrt{\epsilon}$	N	N	N
$\mathrm{U}_m^{\not\perp}$ Our work	OW-CPA	$q\sqrt{\delta} + q\sqrt{\epsilon}$	N	Y	N
$\mathrm{U}_m^{\not\perp}$ Our work	DS	$q\sqrt{\delta} + \epsilon$	N	Y	N
$\mathrm{U}_m^{\perp}$ Our work	OW-VA	$q\sqrt{\delta} + q\sqrt{\epsilon}$	N	Y	N

OW-qPCA (OW-qPVCA) security is just a proof artefact for simulating H. Compared with the DS security notion introduced by [12], the OW-qPCA security is less restrained and weaker. We note that the DS security notion is defined for the DPKE scheme which satisfies (1) statistical disjointness and (2) ciphertext-indistinguishability. Actually, all the DPKE schemes satisfy the OW-qPCA security as the plaintext checking oracle can be simulated by re-encryption in a quantum computer. Therefore, all the instantiations of DS-secure DPKE in [12] are also OW-qPCA-secure. Particularly, the OW-qPCA security is not restrained to the DPKE scheme. Many post-quantum PKE schemes satisfy OW-qPCA security, e.g., NTRU [27], McEliece [37], and Niederreiter [38]. Additionally, we show that the resulting PKE scheme achieved by applying the transformation T to a OW-CPA-secure PKE [7] is also OW-qPCA-secure.

Our security reductions preserve the tightness of the ones in [7, 12] without additional hash for any correctness error δ ($0 \leq \delta < 1$), see Table 3. Our QROM security analyses not only provide post-quantum security guarantees for the KEM schemes constructed by using these modular FO transformations, e.g., Odd Manhattan, Classic McEliece and LEDAkem, but also can help to obtain a variety of combined transformations with different requirements and properties.

1.2 Techniques

Remove the additional hash As explained by Targhi and Unruh [14], their proof technique strongly relies on the additional hash. In their paper, they discussed the QROM security of a variant of FO transformation from a OW-CPA-secure PKE to an IND-CCA-secure PKE. To implement the security reduction, one needs to simulate the decryption oracle without possessing the secret key. In classical proof, a RO-query list is used to simulate such an oracle. In the QROM, the simulator has no way to learn the actual content of adversarial RO queries, therefore such a RO-query list does not exist. Targhi and Unruh circumvented this issue by adding an additional length-preserving hash (modeled as a RO) to the ciphertext. In the security reduction, this additional RO is simulated by a k-wise independent function. For every output of this RO, the simulator can recover the corresponding input by inverting this function. Thereby, the simulator can answer the decryption queries without a secret key.

When considering the generic transformations from a weakly secure PKE to an IND-CCA-secure KEM, one needs to simulate the decapsulation oracle Decaps without the secret key. Indeed, obviously, we can modify the transformations by adding an additional length-preserving hash to the ciphertext so that the simulator can carry out the decryption. Thus, using the key-derivation-function (KDF, modeled as a random oracle H), he can easily simulate the Decaps oracle.

In [11, Theorem 6], Boneh et al. proved the QROM security of a generic hybrid encryption scheme [10], built from an injective trapdoor function and symmetric key encryption scheme. Inspired by their proof idea, we present a novel approach to simulate the Decaps oracle[8].

The high level idea is that we associate the random oracle H (KDF in the KEM) with a secret random function H' by setting $H = H' \circ g$ such that $H'(\cdot) = \textsc{Decaps}(sk, \cdot)$. We demand that the function g should be indistinguishable from an injective function for any efficient quantum adversary. Thus, in the view of the adversary against the IND-CCA security of KEM, H is indeed a random oracle. Meanwhile, we can simulate the Decaps oracle just by using H'. Note that in our simulation of the Decaps oracle, we circumvent the decryption computation. Thereby, there is no need to read the content of adversarial RO queries, which makes it unnecessary to add an additional length-preserving hash to the ciphertext.

Tighten the security bound When proving the IND-CCA security of KEM from the OW-CPA security of underlying PKE for $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$, reprogramming the random oracles G and H is a natural approach. In quantum setting, the one-way to hiding (OW2H) lemma [44, Lemma 6.2] is a practical tool to argue the indistinguishability between games where the random oracles are reprogrammed. However, the OW2H lemma inherently incurs a quadratic security loss.

[8] This method is also used by a concurrent and independent work [12].

To tighten the security bounds, we have to decrease the times of the usage of the OW2H lemma. [7] analyzed the QROM security of $\mathrm{QFO}_m^{\not\perp}$ (and $\mathrm{QFO}_m^{\perp}$) by two steps. First, they presented a QROM security reduction from the OW-CPA security of the underlying PKE to the OW-PCA security of an intermediate scheme PKE$'$. In this step, the random oracle G was reprogrammed, thus by using the OW2H lemma they obtained that $\epsilon'' \leq q^2\delta + q\sqrt{\epsilon}$, where ϵ'' is the success probability of an adversary against the OW-PCA security of PKE$'$. In the second step, they reduced the OW-PCA security of PKE$'$ to the IND-CCA security of KEM, where the random oracles H and H'' (the additional hash) were reprogrammed. Again, by using the OW2H lemma, they gained $\epsilon' \leq q\sqrt{\epsilon''}$. Finally, combing above two bounds, they obtained the security bound of KEM, $\epsilon' \leq q\sqrt{q^2\delta + q\sqrt{\epsilon}}$. Direct combination of the modular analyses leads to twice utilization of the OW2H lemma, which makes the security bound highly non-tight.

When considering the QROM security of $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$, instead of modular analysis, we choose to reduce the OW-CPA security of underlying PKE to the IND-CCA security of KEM directly without introducing an intermediate scheme PKE$'$. In this way, G and H are reprogrammed simultaneously, thus the OW2H lemma is used only once in our reductions.

We also find that the order of the games can highly affect the tightness of the security bound. If we reprogram G and H before simulating the Decaps oracle with the secret random function H', the obtained security bound will be $q\sqrt{\epsilon + q\sqrt{\delta}}$, where the ϵ term has quadratic loss and the δ term has quartic loss. Therefore, we choose to simulate the Decaps oracle with H' before reprogramming G and H. But, in this way, when using the OW2H lemma to argue the indistinguishability between games where G and H are reprogrammed, one has to guarantee the consistency of H and H'. We solve this by generalizing the OW2H lemma to the case where the reprogrammed oracle and other redundant oracle can be sampled simultaneously according to some joint distribution (for complete description of the generalized OW2H lemma, see Lemma 3).

Finally, our derived security bound is $q\sqrt{\delta} + q\sqrt{\epsilon}$, which is much tighter than the bound $q\sqrt{q^2\delta + q\sqrt{\epsilon}}$ obtained by [7].

1.3 Discussion

Tightness. Having a tight security reduction is a desirable property for practice cryptography, especially in large-scale scenarios. In the ROM, if we assume that the underlying PKE scheme in $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$ is IND-CPA-secure, we can obtain a tight reduction from the IND-CPA security of underlying PKE to IND-CCA security of resulting KEM [7]. Specially, if the PKE scheme in $\mathrm{FO}_m^{\not\perp}$ is instantiated with a Ring-LWE-based PKE scheme [41], the security of the underlying Ring-LWE problem can be reduced to the IND-CCA security of KEM [45]. In [12], saito et al. presented a tight security reduction for $\mathrm{U}_m^{\not\perp}$ by assuming a stronger underlying DPKE, which is only satisfied by Classic McEliece in Table 1. For the widely used $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$, quadratic security loss still exists even assuming the IND-CPA security of the underlying PKE scheme, see Table 2. For the tight ROM security reductions in [45, 7], the simulators need to make an elaborate analysis of the RO-query inputs and determine which one of the query inputs can be used to break the IND-CPA security of the underlying PKE scheme [7] or solve a decision Ring-LWE problem [45]. However, in the QROM, such a proof technique will be invalid for the reason that there is no way for the simulators to learn the RO-query inputs [46, 47]. Thus, in the QROM, it is still an important open problem that whether one can develop a novel proof technique to obtain a tight reduction for $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$ assuming standard IND-CPA security of the underlying PKE.

Implicit rejection. For most of the previous generic transformations from a OW-CPA-secure (or IND-CPA-secure) PKE to an IND-CCA-secure KEM, explicit rejection is adopted. In [7], Hofheinz et al. presented several transformations with implicit rejection. These two different versions (explicit rejection and implicit rejection) have their own merits. The transformation with implicit rejection [7] does not require the underlying PKE scheme to be γ-spread [8, 9] (meaning that the ciphertexts generated by the probabilistic encryption algorithm have sufficiently large entropy), which may allow choosing better system parameters for the same security level. Whereas, the ones with explicit rejection have a relatively simple decapsulation algorithm.

In our paper, we just give QROM security reductions for the transformations with implicit rejection. It is not obvious how to extend our QROM security proofs for the transformations with explicit rejection, since the simulator has no way to tell if the submitted ciphertext is valid. In classical ROM, we usually assume the underlying PKE is γ-spread. Then, we can recognize invalid ciphertexts just by testing if they are in the RO-query list, as the probability that the adversary makes queries to the decapsulation oracle with a valid ciphertext which is not in the RO-query list is negligible [45, 7–9]. Unfortunately, in the QROM, the adversary makes quantum queries to the RO, above RO-query list does not exist. Thus, the ROM proof technique for the recognition of invalid ciphertexts is invalid in the QROM. Here, we leave it as an open problem to prove the QROM security of the transformations $\mathrm{FO}^{\not\perp}$ and $\mathrm{FO}_m^{\not\perp}$ with explicit rejection.

2 Preliminaries

Symbol description. Denote $\mathcal{K}$, $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{R}$ as key space, message space, ciphertext space and randomness space, respectively. For a finite set X, we denote the sampling of a uniform random element x by $x \overset{\$}{\leftarrow} X$, and we denote the sampling according to some distribution D by $x \leftarrow D$. By $x = ?y$ we denote the integer that is 1 if $x = y$, and otherwise 0. $\Pr[P : G]$ is the probability that the predicate P holds true where free variables in P are assigned according to the program in G. Denote deterministic (probabilistic) computation of an algorithm

A on input x by $y := A(x)$ ($y \leftarrow A(x)$). A^H means that the algorithm A gets access to the oracle H.

2.1 Quantum Random Oracle Model

In the ROM [10], we assume the existence of a random function H, and give all parties oracle access to this function. The algorithms comprising any cryptographic protocol can use H, as can the adversary. Thus we modify the security games for all cryptographic systems to allow the adversary to make random oracle queries.

When a random oracle scheme is implemented, some suitable hash function H is included in the specification. Any algorithm (including the adversary) replaces oracle queries with evaluations of this hash function. In quantum setting, because a quantum algorithm can evaluate H on an arbitrary superposition of inputs, we must allow the quantum adversary to make quantum queries to the random oracle. We call this the quantum random oracle model [11]. Unless otherwise specified, the queries to random oracles are quantum in our paper.

Tools. Next we state four lemmas that we will use throughout the paper. The first two lemmas have been proved in other works, and we prove the last two in Appendixes B and C. Most of the background in quantum computation needed to understand this paper is just for above two proofs. Therefore, we present the necessary background in Appendix A. Here, we just recall two basic facts about quantum computation.

- Fact 1. Any classical computation can be implemented on a quantum computer.

- Fact 2. Any function that has an efficient classical algorithm computing it can be implemented efficiently as a quantum-accessible oracle.

Lemma 1 (Simulating the random oracle [48, Theorem 6.1]). *Let H 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 queries to H has in distinguishing H from a truly random function is identically 0.*

Lemma 2 (Generic search problem [49, 50]). *Let $\gamma \in [0, 1]$. Let Z be a finite set. $N_1 : Z \to \{0, 1\}$ is the following function: For each 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 $\forall z : N_2(z) = 0$. If an oracle algorithm A makes at most q quantum queries to N_1 (or N_2), then*

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

Particularly, the probability of A finding a z such that $N_1(z) = 1$ is at most $2q\sqrt{\gamma}$, i.e., $\Pr[N_1(z) = 1 : z \leftarrow A^{N_1}] \leq 2q\sqrt{\gamma}$.

Note. [49, Lemma 37] and [50, Theorem 1] just consider the specific case where all p_zs are equal to γ. But in our security proof, we need to consider the case where $p_z \leq \gamma$ and p_zs are in general different from each other. Fortunately, it is not difficult to verify that the proof of [49, Lemma 37] can be extended to this generic case.

The one-way to hiding (OW2H) lemma [44, Lemma 6.2] is a useful tool for reducing a hiding (i.e., indistinguishability) property to a guessing (i.e., one-wayness) property in the security proof. Roughly speaking, the lemma states that if there exists an oracle algorithm A who issuing at most q_1 queries to random oracle $\mathcal{O}_1$ can distinguish $(x, \mathcal{O}_1(x))$ from (x, y), where y is chosen uniformly at random, we can construct another oracle algorithm B who can find x by running A and measuring one of A's query. However, in our security proof, the oracle $\mathcal{O}_1$ is not a perfect random function and A can have access to other oracle $\mathcal{O}_2$ associated to $\mathcal{O}_1$. Therefore, we generalize the OW2H lemma.

Lemma 3 (One-way to hiding, with redundant oracle). *Let oracles $\mathcal{O}_1$, $\mathcal{O}_2$, input parameter inp and x be sampled from some joint distribution D, where $x \in \{0,1\}^n$ (the domain of $\mathcal{O}_1$) and $\mathcal{O}_1(x)$ is uniformly distributed on $\{0,1\}^m$ (the codomain of $\mathcal{O}_1$) conditioned on any fixed $\mathcal{O}_1(x')$ for all $x' \neq x$, $\mathcal{O}_2$, inp and x, and independent from $\mathcal{O}_2$.*

Consider an oracle algorithm $A^{\mathcal{O}_1,\mathcal{O}_2}$ that makes at most q_1 queries to $\mathcal{O}_1$ and q_2 queries to $\mathcal{O}_2$. Denote E_1 as the event that $A^{\mathcal{O}_1,\mathcal{O}_2}$ on input $(inp, x, \mathcal{O}_1(x))$ outputs 1. Reprogram $\mathcal{O}_1$ at x and replace $\mathcal{O}_1(x)$ by a uniformly random y from $\{0,1\}^m$. Denote E_2 as the event that $A^{\mathcal{O}'_1,\mathcal{O}_2}$ on input (inp, x, y) outputs 1 after $\mathcal{O}_1$ is reprogrammed, where $\mathcal{O}'_1$ is denoted as the reprogrammed $\mathcal{O}_1$. Let $B^{\mathcal{O}_1,\mathcal{O}_2}$ be an oracle algorithm that on input (inp, x) does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_1\}$ and $y \xleftarrow{\$} \{0,1\}^m$, run $A^{\mathcal{O}'_1,\mathcal{O}_2}(inp, x, y)$ until the i-th query to $\mathcal{O}'_1$, measure the argument of the query in the computational basis, and output the measurement outcome. (When A makes less than i queries, B outputs $\perp \notin \{0,1\}^n$.) Let

$$\Pr[E_1] = \Pr[b' = 1 : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, b' \leftarrow A^{\mathcal{O}_1,\mathcal{O}_2}(inp, x, \mathcal{O}_1(x))]$$
$$\Pr[E_2] = \Pr[b' = 1 : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, y \xleftarrow{\$} \{0,1\}^m, b' \leftarrow A^{\mathcal{O}'_1,\mathcal{O}_2}(inp, x, y]$$
$$P_B := \Pr[x' = x : (\mathcal{O}_1, \mathcal{O}_2, inp, x) \leftarrow D, x' \leftarrow B^{\mathcal{O}_1,\mathcal{O}_2}(inp, x)].$$

Then

$$|\Pr[E_1] - \Pr[E_2]| \leq 2q_1\sqrt{P_B}.$$

Note that $\mathcal{O}_2$ is unchanged during the reprogramming of $\mathcal{O}_1$ at x. Thus, intuitively, $\mathcal{O}_2$ is redundant and unhelpful for A distinguishing $(x, \mathcal{O}_1(x))$ from (x, y). The complete proof of Lemma 3 is similar to the proof of the OW2H lemma [44, Lemma 6.2] and we present it in Appendix B.

Lemma 4. *Let Ω_H ($\Omega_{H'}$) be the set of all functions $H : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ ($H' : \{0,1\}^{n_2} \to \{0,1\}^m$). Let $H \xleftarrow{\$} \Omega_H$, $H' \xleftarrow{\$} \Omega_{H'}$, $x \xleftarrow{\$} \{0,1\}^{n_1}$. Let $F_0 = H(x, \cdot)$, $F_1 = H'(\cdot)$ Consider an oracle algorithm A^{H,F_i} that makes at* *most q queries to H and F_i ($i \in \{0,1\}$). If x is independent from the A^{H,F_i}'s view,*

$$\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq 2q\frac{1}{\sqrt{2^{n_1}}}.$$

We now sketch the proof of Lemma 4. The complete proof is in Appendix C.

Proof sketch. In classical setting, it is obvious that $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right|$ can be bounded by the probability that A performs an H-query with input $(x,*)$. As x is independent from A^{H,F_i}'s view, $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq q\frac{1}{2^{n_1}}$. In quantum setting, it is not well-defined that $\mathcal{A}$ queries $(x,*)$ from H, since H can be queried in superposition. To circumvent this problem, we follow Unruh's proof technique in [44, Lemma 6.2] and define a new adversary B who runs A, but at some random query stops and measures the query input. Let P_B be the probability that B measures x. Similarly to [44, Lemma 6.2], we can bound $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right|$ by $2q\sqrt{P_B}$. Since x is independent from the A^{H,F_i}'s view, $P_B = \frac{1}{2^{n_1}}$. Thus, $\left|\Pr[1 \leftarrow A^{H,F_0}] - \Pr[1 \leftarrow A^{H,F_1}]\right| \leq 2q\frac{1}{\sqrt{2^{n_1}}}$.

2.2 Cryptographic Primitives

Definition 1 (Public-key encryption). *A public-key encryption scheme* PKE = (Gen, Enc, Dec) *consists of a triple of polynomial time (in the security parameter λ) algorithms and a finite message space $\mathcal{M}$. Gen, the key generation algorithm, is a probabilistic algorithm which on input 1^λ outputs a public/secret key-pair (pk, sk). The encryption algorithm Enc, on input pk and a message $m \in \mathcal{M}$, outputs a ciphertext $c \leftarrow Enc(pk, m)$. If necessary, we make the used randomness of encryption explicit by writing $c := Enc(pk, m; r)$, where $r \overset{\$}{\leftarrow} \mathcal{R}$ ($\mathcal{R}$ is the randomness space). Dec, the decryption algorithm, is a deterministic algorithm which on input sk and a ciphertext c outputs a message $m := Dec(sk, c)$ or a special symbol $\perp \notin \mathcal{M}$ to indicate that c is not a valid ciphertext.*

Definition 2 (Correctness [7]). *A* PKE *is* δ-correct *if*

$$E[\max_{m \in \mathcal{M}} \Pr[Dec(sk, c) \neq m : c \leftarrow Enc(pk, m)]] \leq \delta,$$

where the expectation is taken over $(pk, sk) \leftarrow Gen$.

We now define four security notions for public-key encryption: one-way against chosen plaintext attacks (OW-CPA), one-way against validity checking attacks (OW-VA), one-way against quantum plaintext checking attacks (OW-qPCA) and one-way against quantum plaintext and (classical) validity checking attacks (OW-qPVCA).

Definition 3 (OW-ATK-secure PKE). *Let* PKE = (Gen, Enc, Dec) *be a public-key encryption scheme with message space $\mathcal{M}$. For* ATK $\in$ {CPA, VA, qPCA, qPVCA}, *we define OW-ATK games as in Fig. 1, where*

$$O_{ATK} := \begin{cases} \perp & \text{ATK} = \text{CPA} \\ \textsc{Val}(\cdot) & \text{ATK} = \text{VA} \\ \textsc{Pco}(\cdot,\cdot) & \text{ATK} = \text{qPCA} \\ \textsc{Pco}(\cdot,\cdot), \textsc{Val}(\cdot) & \text{ATK} = \text{qPVCA}. \end{cases}$$

Define the OW-ATK advantage function of an adversary $\mathcal{A}$ against PKE as $\mathtt{Adv}^{\text{OW-ATK}}_{\text{PKE}}(\mathcal{A}) := \Pr[\text{OW-ATK}^{\mathcal{A}}_{\text{PKE}} = 1]$.

```
Game OW-ATK
    $(pk, sk) \leftarrow Gen$
    $m^* \xleftarrow{\$} \mathcal{M}$
    $c^* \leftarrow Enc(pk, m^*)$
    $m' \leftarrow \mathcal{A}^{O_{\text{ATK}}}(pk, c^*)$
    return $m' =?m^*$
```

```
Pco(m,c)
    if $m \notin \mathcal{M}$
        return $\perp$
    else return
        $Dec(sk, c) =?m$
```

```
Val(c)
    $m := Dec(sk, c)$
    if $m \in \mathcal{M}$
        return 1
    else return 0
```

Fig. 1: Games OW-ATK (ATK ∈ {CPA, VA, qPCA, qPVCA}) for PKE, where O_{ATK} is defined in Definition 3. In games qPCA and qPVCA, the adversary $\mathcal{A}$ can query the Pco oracle with quantum state.

Remark. We note that the security game OW-qPCA (OW-qPVCA) is the same as OW-PCA (OW-PVCA) except the adversary $\mathcal{A}$'s queries to the Pco oracle. In OW-qPCA (OW-qPVCA) game, $\mathcal{A}$ can make quantum queries to the Pco oracle, while in OW-PCA (OW-PVCA) game only the classical queries are allowed. These two new security notations will be used in the security analysis of modular FO transformations in Sec. 4.

Definition 4 (DS-secure DPKE[12]). *Let $D_{\mathcal{M}}$ denote an efficiently sampleable distribution on $\mathcal{M}$. A DPKE scheme (Gen,Enc,Dec) with plaintext and ciphertext spaces $\mathcal{M}$ and $\mathcal{C}$ is $D_{\mathcal{M}}$-disjoint simulatable if there exists a PPT algorithm S that satisfies (1) Statistical disjointness:* $\textsc{Disj}_{\text{PKE},S} := \max_{pk} \Pr[c \in Enc(pk, \mathcal{M}) : c \leftarrow S(pk)]$ *is negligible. (2) Ciphertext-indistinguishability: For any PPT adversary $\mathcal{A}$,* $\mathtt{Adv}^{\text{DS-IND}}_{\text{PKE},D_{\mathcal{M}},S}(\mathcal{A}) := |\Pr[\mathcal{A}(pk, c^*) \rightarrow 1 : (pk, sk) \leftarrow Gen; m^* \leftarrow D_{\mathcal{M}}; c^* := Enc(pk, m^*)] - \Pr[\mathcal{A}(pk, c^*) \rightarrow 1 : (pk, sk) \leftarrow Gen; c^* \leftarrow S(pk)]|$ *is negligible.*

Definition 5 (Key encapsulation). *A key encapsulation mechanism KEM consists of three algorithms Gen, Encaps and Decaps. The key generation algorithm Gen outputs a key pair (pk, sk). The encapsulation algorithm Encaps, on input pk, outputs a tuple (K, c) where c is said to be an encapsulation of the key K which is contained in key space $\mathcal{K}$. The deterministic decapsulation algorithm Decaps, on input sk and an encapsulation c, outputs either a key*

$K := Decaps(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.

```
Game IND-CCA
    $(pk, sk) \leftarrow Gen$
    $b \xleftarrow{\$} \{0,1\}$
    $(K_0^*, c^*) \leftarrow Encaps(pk)$
    $K_1^* \xleftarrow{\$} \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\textsc{Decaps}}(pk, c^*, K_b^*)$
    return $b' =?b$

$\textsc{Decaps}(sk, c)$
    if $c = c^*$
        return $\perp$
    else return
        $K := Decaps(sk, c)$
```

Fig. 2: IND-CCA game for KEM.

We now define a security notion for KEM: indistinguishability against chosen ciphertext attacks (IND-CCA).

Definition 6 (IND-CCA-secure KEM). *We define the IND-CCA game as in Fig. 2 and the IND-CCA advantage function of an adversary $\mathcal{A}$ against* KEM *as* $\mathtt{Adv}_{\mathrm{KEM}}^{\text{IND-CCA}}(\mathcal{A}) := \left|\Pr[\text{IND-CCA}_{\mathrm{KEM}}^{\mathcal{A}} = 1] - \frac{1}{2}\right|$.

We also define OW-ATK security of PKE, DS security of DPKE and IND-CCA security of KEM in the QROM, where adversary $\mathcal{A}$ can make quantum queries to random oracles. Following the work [7], we also make the convention that the number q_H of adversarial queries to a random oracle H counts the total number of times H is executed in the experiment. That is, the number of $\mathcal{A}$'s explicit queries to H plus the number of implicit queries to H made by the experiment.

3 Security Proofs for Two Generic KEM Constructions in the QROM

In this section, we revisit two generic transformations, FO$^{\not\perp}$ and FO$^{\not\perp}_m$, see Fig. 3 and Fig. 4. These two transformations are widely used in the post-quantum IND-CCA-secure KEM constructions, see Table 1. But, there are no QROM security proofs for them. To achieve QROM security, some proposals, e.g., FrodoKEM, followed Hofheinz et al.'s work [7] and modified FO$^{\not\perp}$ and FO$^{\not\perp}_m$ by adding an additional length-preserving hash function to the ciphertext. Here, we present two QROM security proofs for FO$^{\not\perp}$ and FO$^{\not\perp}_m$ respectively without suffering any ciphertext overhead.

```
$Gen'$
$(pk, sk) \leftarrow Gen$
$s \xleftarrow{\$} \mathcal{M}$
$sk' := (sk, s)$
return $(pk, sk')$

$Encaps(pk)$
$m \xleftarrow{\$} \mathcal{M}$
$c = Enc(pk, m; G(m))$
$K := H(m, c)$
return $(K, c)$

$Decaps(sk', c)$
Parse $sk' = (sk, s)$
$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
    return $K := H(m', c)$
else return
    $K := H(s, c)$
```

Fig. 3: IND-CCA-secure KEM-I=$\mathrm{FO}^{\not\perp}$[PKE,G,H]

```
$Gen'$
$(pk, sk) \leftarrow Gen$
$k \xleftarrow{\$} \mathcal{K}^{prf}$
$sk' := (sk, k)$
return $(pk, sk')$

$Encaps(pk)$
$m \xleftarrow{\$} \mathcal{M}$
$c = Enc(pk, m; G(m))$
$K := H(m)$
return $(K, c)$

$Decaps(sk', c)$
Parse $sk' = (sk, k)$
$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
    return $K := H(m')$
else return
    $K := f(k, c)$
```

Fig. 4: IND-CCA-secure KEM-II=$\mathrm{FO}^{\not\perp}_m$[PKE,$G$,$H$,$f$]

To a public-key encryption scheme PKE = (Gen, Enc, Dec) with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, hash functions $G : \mathcal{M} \to \mathcal{R}$, $H : \{0,1\}^* \to \{0,1\}^n$ and a pseudorandom function (PRF) f with key space $\mathcal{K}^{prf}$, we associate KEM-I=$\mathrm{FO}^{\not\perp}$[PKE,G,H] and KEM-II=$\mathrm{FO}^{\not\perp}_m$[PKE,$G$,$H$,$f$][9] shown in Fig. 3 and Fig. 4, respectively. The following two theorems establish that IND-CCA securities of KEM-I and KEM-II can both reduce to the OW-CPA security of PKE, in the QROM.

Theorem 1 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-I IND-CCA). *If* PKE *is δ-*correct*, for any IND-CCA $\mathcal{B}$ against KEM-I, issuing at most q_D queries to the decapsulation oracle* Decaps*, at most q_G queries to the random oracle G and at most q_H queries to the random oracle H, there exists a OW-CPA adversary $\mathcal{A}$ against* PKE *such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2(q_G + q_H) \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{A})}$ *and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

Proof. Given (pk, sk) and $m \in \mathcal{M}$, let

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

[9] $\mathrm{FO}^{\not\perp}_m$ here is the generic version of $\mathrm{FO}^{\not\perp}_m$ in [7]. In their work, such a pseudorandom function f is instantiated with $H(s, \cdot)$ (s is a random seed and contained in the secret key sk').

denote the set of "bad" randomness and let $\mathcal{R}_{\text{good}}(pk, sk, m) = \mathcal{R} \backslash \mathcal{R}_{\text{bad}}(pk, sk, m)$ denote the set of "good" randomness. Define

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

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

Let G' be a random function such that $G'(m)$ is sampled according to the uniform distribution in $\mathcal{R}_{\text{good}}(pk, sk, m)$. Let $\Omega_{G'}$ be the set of all functions G'. Denote Ω_G, Ω_H and $\Omega_{H'}$ as the sets of all functions $G : \mathcal{M} \to \mathcal{R}$, $H : \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ and $H' : \mathcal{C} \to \mathcal{K}$, respectively.

Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-I, issuing at most q_D queries to Decaps, at most q_G queries to G and at most q_H queries to H. Consider the games in Fig. 5 and Fig. 9.

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

$$\left| \Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2} \right| = \mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-I}}(\mathcal{B}).$$

```
GAMES $G_0 - G_6$
$(pk, sk') \leftarrow Gen'; G \xleftarrow{\$} \Omega_G$
$G' \xleftarrow{\$} \Omega_{G'}; G := G'$   $//G_2 - G_4$
$H_1, H_2 \xleftarrow{\$} \Omega_{H'}; H_3 \xleftarrow{\$} \Omega_H$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* := G(m^*)$
$r^* \xleftarrow{\$} \mathcal{R}$   $//G_6$
$c^* := Enc(pk, m^*; r^*)$
$k_0^* := H(m^*, c^*)$
$k_0^* \xleftarrow{\$} \mathcal{K}$   $//G_6$
$k_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0, 1\}$
$b' \leftarrow \mathcal{B}^{G, H, \text{Decaps}}(pk, c^*, k_b^*)$
return $b' =?b$
```

```
$H(m, c)$
if $Enc(pk, m; G(m)) = c$   $//G_3 - G_6$
    return $H_1(c)$   $//G_3 - G_6$
return $H_3(m, c)$
```

```
Decaps $(c \neq c^*)$ $//G_0 - G_3$
Parse $sk' = (sk, s)$
$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
    return $K := H(m', c)$
else return
    $K := H(s, c)$   $//G_0$
    $K := H_2(c)$   $//G_1 - G_3$
```

```
Decaps $(c \neq c^*)$   $//G_4 - G_6$
return $K := H_1(c)$
```

Fig. 5: Games G_0-G_6 for the proof of Theorem 1

```
$A^{H,F_i}$
    $(pk, sk) \leftarrow Gen; G \xleftarrow{\$} \Omega_G$
    $m^* \xleftarrow{\$} \mathcal{M}$
    $r^* := G(m^*)$
    $c^* := Enc(pk, m^*; r^*)$
    $k_0^* := H(m^*, c^*); k_1^* \xleftarrow{\$} \mathcal{K}$
    $b \xleftarrow{\$} \{0, 1\}$
    $b' \leftarrow \mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
    return $b' =?b$

Decaps $(c \neq c^*)$
    $m' := Dec(sk, c)$
    if $Enc(pk, m'; G(m')) = c$
        return $K := H(m', c)$
    else return
        $K := F_i(c)$
```

Fig. 6: A^{H,F_i} for the proof of Theorem 1.

```
$A^N(pk, sk)$
    Pick $2q_G$-wise function $f$
    $b'' \leftarrow B^{\widetilde{G}}(pk, sk)$
    return $b''$

$\widetilde{G}(m)$
    if $N(m) = 0$
        $\widetilde{G}(m) = Sample(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
    else
        $\widetilde{G}(m) = Sample(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
    return $\widetilde{G}(m)$
```

Fig. 7: A^N for the proof of Theorem 1

Game G_1. In game G_1, we change the Decaps oracle that $H_2(c)$ is returned instead of $H(s, c)$ for an invalid encapsulation c. Define an oracle algorithm A^{H,F_i} $(i \in \{0, 1\})$, see Fig. 6. Let $H = H_3$, $F_0(\cdot) = H_3(s, \cdot)$ $(s \xleftarrow{\$} \mathcal{M})$ and $F_1 = H_2$, where H_2 and H_3 are chosen in the same way as G_0 and G_1. Then, $\Pr[G_i^{\mathcal{B}} \Rightarrow 1] = \Pr[1 \leftarrow A^{H,F_i}]$. Since the uniform secret s is chosen independently from A^{H,F_i}'s view, we can use Lemma 4 to obtain

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_H \cdot \frac{1}{\sqrt{|\mathcal{M}|}}.$$

Game G_2. In game G_2, we replace G by G' that uniformly samples from "good" randomness at random, i.e., $G' \xleftarrow{\$} \Omega_{G'}$. Note that the distinguishing problem between G_1 and G_2 is essentially the distinguishing problem between G and G'. Specifically, we can construct an adversary $B^{\widetilde{G}}(pk, sk)$ against the distinguishing problem between G and G' by taking the accessible oracle $\widetilde{G}$ as G, simulating $\mathcal{B}$'s view and outputting in the same way as G_1 and G_2. Then, for any fixed

(pk, sk) that is generated by Gen, if $\widetilde{G} = G$, $B^{\widetilde{G}}(pk, sk)$ perfectly simulates G_1 and $\Pr[1 \leftarrow B^{G} : (pk, sk)] = \Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]$. If $\widetilde{G} = G'$, $B^{\widetilde{G}}(pk, sk)$ perfectly simulates G_2 and $\Pr[\Pr[1 \leftarrow B^{G'} : (pk, sk)] = \Pr[G_2^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]$. Thus,

$$\begin{aligned} &\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \\ = &\left|\Pr[1 \leftarrow B^{G} : (pk, sk)] - \Pr[1 \leftarrow B^{G'} : (pk, sk)]\right|. \end{aligned}$$

Next, we will show that any algorithm that distinguishes G from G' can be converted into an algorithm that distinguishes N_1 from N_2, where N_1 is a function such that $N_1(m)$ is sampled from the Bernoulli distribution $B_{\delta(pk,sk,m)}$, i.e., $\Pr[N_1(m) = 1] = \delta(pk, sk, m)$ and $\Pr[N_1(m) = 0] = 1 - \delta(pk, sk, m)$, and N_2 is a constant function that always outputs 0 for any input.

For any adversary $B^{\widetilde{G}}(pk, sk)$, we can construct an adversary $A^{N}(pk, sk)$ as in Fig. 7. $Sample(\mathcal{Y})$ is a probabilistic algorithm that returns a uniformly distributed $y \xleftarrow{\$} \mathcal{Y}$. $Sample(\mathcal{Y}; f(m))$ denotes the deterministic execution of $Sample(\mathcal{Y})$ using explicitly given randomness $f(m)$.

Note that $\widetilde{G} = G$ if $N = N_1$ and $\widetilde{G} = G'$ if $N = N_2$. Thus, for any fixed (pk, sk) that is generated by Gen, $\Pr[1 \leftarrow A^{N_1} : (pk, sk)] = \Pr[1 \leftarrow B^{G} : (pk, sk)]$ and $\Pr[1 \leftarrow A^{N_2} : (pk, sk)] = \Pr[1 \leftarrow B^{G'} : (pk, sk)]$. Conditioned on a fixed (pk, sk) we obtain by Lemma 2

$$\begin{aligned} &\left|\Pr[1 \leftarrow B^{G} : (pk, sk)] - \Pr[1 \leftarrow B^{G'} : (pk, sk)]\right| \\ = &\left|\Pr[1 \leftarrow A^{N_1} : (pk, sk)] - \Pr[1 \leftarrow A^{N_2} : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}. \end{aligned}$$

Note that $\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right|$ can be bounded by the maximum distinguishing probability between G and G' for $B^{\widetilde{G}}(pk, sk)$. Thus,

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}.$$

By averaging over $(pk, sk) \leftarrow Gen$ we finally obtain

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

Game G_3. Note that in game G_2, $H(m, c) = H_3(m, c)$. In game G_3, if H-query input (m, c) satisfies $g(m) = c$, the response is replaced by $H_1^{g}(m) = H_1 \circ g(m) = H_1(g(m)) = H_1(c)$, where

$$g(\cdot) = Enc(pk, \cdot; G(\cdot)).$$

Note that g in G_3 is an injective function since G in G_3 only samples from "good" randomness. Thus, the distributions of H in G_2 and G_3 are identical. Therefore,

$$\Pr[G_2^{\mathcal{B}} \Rightarrow 1] = \Pr[G_3^{\mathcal{B}} \Rightarrow 1].$$

GAME G_4. In game G_4, the DECAPS oracle is changed that it makes no use of the secret key sk' any more. When $\mathcal{B}$ queries the DECAPS oracle on c ($c \neq c^*$), $K := H_1(c)$ is returned as the response. Let $m' := Dec(sk, c)$ and consider the following two cases.

Case 1: $Enc(pk, m'; G(m')) = c$. In this case, $H(m', c) = H_1(c)$. Thus, both DECAPS oracles in G_3 and G_4 return the same value.

Case 2: $Enc(pk, m'; G(m')) \neq c$. Random values $H_2(c)$ and $H_1(c)$ are returned in G_3 and G_4 respectively. In G_3, H_2 is a random function independent of the oracles G and H, thus $H_2(c)$ is uniform at random in $\mathcal{B}$'s view. In G_4, $\mathcal{B}$'s queries to H can only help him get access to H_1 at $\hat{c}$ such that $g(\hat{m}) = \hat{c}$ for some $\hat{m}$. Consequently, $H_1(c)$ is also a fresh random key just like $H_2(c)$ in $\mathcal{B}$'s view due to the fact that G in G_4 only samples from "good" randomness and there exits no an m'' such that $g(m'') = c$. Hence, in this case, the output distributions of the DECAPS oracles in G_3 and G_4 are same in $\mathcal{B}$'s view.

As a result, the output distributions of G_3 and G_4 are statistically indistinguishable and we have

$$\Pr[G_3^{\mathcal{B}} \Rightarrow 1] = \Pr[G_4^{\mathcal{B}} \Rightarrow 1].$$

GAME G_5. In game G_5, we replace G' by G, that is, G in this game is reset to be an ideal random oracle. Then, similar to the case of G_1 and G_2, the distinguishing problem between G_4 and G_5 also can be converted to the distinguishing problem between G and G'. Particularly, we can construct an adversary $B'^{\widetilde{G}}(pk, sk)$ against the distinguishing problem between G and G' by simulating G with the accessible oracle $\widetilde{G}$, simulating $\mathcal{B}$'s view and outputting in the same way as G_4 and G_5. Then, for any fixed (pk, sk) that is generated by Gen, G_4 is perfectly simulated when $\widetilde{G} = G'$ and G_5 is perfectly simulated when $\widetilde{G} = G$. Thus,

$$\begin{aligned} &\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \\ =\ &\left|\Pr[1 \leftarrow B'^{G} : (pk, sk)] - \Pr[1 \leftarrow B'^{G'} : (pk, sk)]\right|. \end{aligned}$$

Using the same analysis in game G_2, we can obtain that

$$\left|\Pr[1 \leftarrow B'^{G} : (pk, sk)] - \Pr[1 \leftarrow B'^{G'} : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}.$$

Then,

$$\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]\right| \leq 2q_G\sqrt{\delta(pk, sk)}.$$

Finally, averaging over $(pk, sk)\leftarrow Gen$, we have

$$\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G\sqrt{\delta}.$$

GAME G_6. In game G_6, r^* and k_0^* are chosen uniformly at random from $\mathcal{R}$ and $\mathcal{K}$, respectively. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_6^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

Note that in this game we reprogram the oracles G and H on inputs m^* and (m^*, c^*) respectively. In classical setting, this will be unnoticed unless the event QUERY that $\mathcal{B}$ queries G on m^* or H on (m^*, c^*) happens. Then we can argue that G_5 and G_6 are indistinguishable until QUERY happens. In quantum setting, due to the quantum queries to G and H, the case is complicated and we will use Lemma 3 to bound $\left|\Pr[G_5^{\mathcal{B}} \Rightarrow 1] - \Pr[G_6^{\mathcal{B}} \Rightarrow 1]\right|$. Note that (m^*, c^*) is a valid plaintext-ciphertext pair, i.e., $g(m^*) = c^*$. Therefore, $H(m^*, c^*) = H_1(c^*) = H_1^g(m^*)$. Actually, we just reprogram G and H_1^g at m^*.

Let $(G \times H_1^g)(x) := (G(x), H_1^g(x))$[10]. H_1^g and H_3 are internal random oracles that $\mathcal{B}$ can have access to only by querying the oracle H. Then, the number of total queries to $G \times H_1^g$ is at most $q_G + q_H$. Let H_1' be the function such that $H_1'(g(m^*)) = \perp$ and $H_1' = H_1$ everywhere else. H_1' is exactly the DECAPS oracle in G_5 and G_6 and unchanged during the reprogramming of $G \times H_1^g$.

Let $A^{G \times H_1^g, H_1'}$ be an oracle algorithm that has quantum access to $G \times H_1^g$ and H_1', see Fig. 8. Sample G, H_1, H_1^g and pk in the same way as G_5 and G_6, i.e., $(pk, sk') \leftarrow Gen', G \xleftarrow{\$} \Omega_G, H_1 \xleftarrow{\$} \Omega_{H'}, H_1^g := H_1 \circ g$. Let $m^* \xleftarrow{\$} \mathcal{M}$.

Then, if $r^* := G(m^*)$ and $k_0^* := H_1^g(m^*)$, $A^{G \times H_1^g, H_1'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_5. And, if $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{G \times H_1^g, H_1'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_6. Let $B^{G \times H_1^g, H_1'}$ be an oracle algorithm that on input (pk, m^*) does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{G \times H_1^g, H_1'}(pk, m^*, (r^*, k_0^*))$ until the i-th query to $G \times H_1^g$, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{G \times H_1^g, H_1'}$ makes less than i queries, output $\perp$). Define game G_7 as in Fig. 9. Then, $\Pr[B^{G \times H_1^g, H_1'} \Rightarrow m^*] = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = G \times H_1^g$, $\mathcal{O}_2 = H_1'$, $inp = pk$, $x = m^*$ and $y = (r^*, k_0^*)$, we have

$$\left|\Pr[G_5^{\mathcal{B}} \Rightarrow 1] - \Pr[G_6^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_7^{\mathcal{B}} \Rightarrow 1]}.$$

[10] Note that if one wants to make queries to G (or H_1^g) by accessing to $G \times H_1^g$, he just needs to prepare a uniform superposition of all states in the output register responding to H_1^g (or G). This trick [51, 52, 14] has been used to ignore part of the output of an oracle.

```
$A^{G\times H_1^g, H_1'}(pk, m^*, (r^*, k_0^*))$
    $H_3 \xleftarrow{\$} \Omega_H$
    $c^* := Enc(pk, m^*; r^*)$
    $k_1^* \xleftarrow{\$} \mathcal{K}$
    $b \xleftarrow{\$} \{0,1\}$
    $b' \leftarrow \mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
    return $b' =?b$

$H(m, c)$
    if $g(m) = c$
        return $H_1^g(m)$
    else return $H_3(m, c)$

$\textsc{Decaps}\ (c \neq c^*)$
    return $K := H_1'(c)$
```

Fig. 8: $A^{G\times H_1^g, H_1'}$ for the proof of Theorem 1.

```
GAMES $G_7$
    $i \xleftarrow{\$} \{1, \dots, q_G + q_H\}, (pk, sk') \leftarrow Gen', G \xleftarrow{\$} \Omega_G$
    $H_1 \xleftarrow{\$} \Omega_{H'}, H_3 \xleftarrow{\$} \Omega_H$
    $m^* \xleftarrow{\$} \mathcal{M}, r^* \xleftarrow{\$} \mathcal{R}$
    $c^* := Enc(pk, m^*; r^*)$
    $k_0^*, k_1^* \xleftarrow{\$} \mathcal{K}$
    $b \xleftarrow{\$} \{0,1\}$
    run $\mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$ until the $i-$th query to $G \times H_1^g$
    measure the argument $\hat{m}$
    return $\hat{m} =?m^*$

$H(m, c)$
    if $Enc(pk, m; G(m)) = c$
        return $H_1(c)$
    else return $H_3(m, c)$

$\textsc{Decaps}\ (c \neq c^*)$
    return $K := H_1(c)$
```

Fig. 9: Game G_7 for the proof of Theorem 1

Next, we construct an adversary $\mathcal{A}$ against the OW-CPA security of the PKE scheme such that $\mathtt{Adv}_{\mathrm{PKE}}^{\mathrm{OW\text{-}CPA}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input (1^λ, pk, c) does the following:

1. Run the adversary $\mathcal{B}$ in game G_7.
2. Use a $2q_G$-wise independent function and two different $2q_H$-wise independent functions to simulate the random oracles G, H_1 and H_3 respectively. The random oracle H is simulated in the same way as the one in game G_7.
3. Answer the decapsulation queries by using the DECAPS oracle in Fig. 9.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c, k^*).

5. Select $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, measure the argument $\hat{m}$ of i-th query to $G \times H_1^g$ and output $\hat{m}$.

According to Lemma 1, $\mathtt{Adv}_{\mathrm{PKE}}^{\mathrm{OW\text{-}CPA}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. Finally, combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}_{\mathrm{KEM\text{-}I}}^{\mathrm{IND\text{-}CCA}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 4q_G\sqrt{\delta} + 2(q_G + q_H) \cdot \sqrt{\mathtt{Adv}_{\mathrm{PKE}}^{\mathrm{OW\text{-}CPA}}(\mathcal{A})}.$$

□

Theorem 2 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-II IND-CCA). *If* PKE *is* δ-correct*, for any IND-CCA $\mathcal{B}$ against KEM-II, issuing at most q_D classical queries to the decapsulation oracle* DECAPS *and at most q_G (q_H) queries to random oracle G (H), there exist a quantum OW-CPA adversary $\mathcal{A}$ against* PKE *and an adversary $\mathcal{A}'$ against the security of* PRF *with at most q_D classical queries such that $\mathtt{Adv}_{\mathrm{KEM\text{-}II}}^{\mathrm{IND\text{-}CCA}}(\mathcal{B}) \leq \mathtt{Adv}_{\mathrm{PRF}}(\mathcal{A}') + 4q_G\sqrt{\delta} + 2(q_H + q_G) \cdot \sqrt{\mathtt{Adv}_{\mathrm{PKE}}^{\mathrm{OW\text{-}CPA}}(\mathcal{A})}$ and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

The only difference between KEM-I and KEM-II is the KDF function. In KEM-I, $K = H(m, c)$, while $K = H(m)$ in KEM-II. Note that given pk and random oracle G, c is determined by m. The proof of Theorem 2 is similar to the one of Theorem 1 and we present it in Appendix D.

4 Modular Analysis of FO transformation in the QROM

In [7], Hofheinz et al. introduced seven modular transformations T, $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}_m^{\not\perp}$, $\mathrm{U}_m^{\perp}$, $\mathrm{QU}_m^{\not\perp}$ and $\mathrm{QU}_m^{\perp}$. But, they just presented QROM security reductions for the transformations T, $\mathrm{QU}_m^{\not\perp}$ and $\mathrm{QU}_m^{\perp}$. Different from the transformations $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}_m^{\not\perp}$ and $\mathrm{U}_m^{\perp}$, the transformations $\mathrm{QU}_m^{\not\perp}$ and $\mathrm{QU}_m^{\perp}$ have an additional length-preserving hash in the ciphertext, thus they can follow the proof technique in [53, 14] to give QROM security reductions for them. As they pointed [14], their QROM security reductions quite rely on this additional hash. And, QROM security reductions for $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}_m^{\not\perp}$ and $\mathrm{U}_m^{\perp}$ are missing in [7]. In [12], saito et al. presented a tight QROM security reduction for $\mathrm{U}_m^{\not\perp}$ with stronger assumptions for underlying DPKE scheme, DS-security and perfect correctness.

In this section, we revisit the transformations $\mathrm{U}^{\not\perp}$, $\mathrm{U}^{\perp}$, $\mathrm{U}_m^{\not\perp}$ and $\mathrm{U}_m^{\perp}$, and argue their QROM security without any modification to the constructions and with correctness error into consideration. [7] has shown that the transformation T can turn a OW-CPA-secure PKE into a OW-PCA-secure PKE in the QROM. In Section 4.1, we first show that the resulting PKE scheme by applying T to a OW-CPA-secure PKE is also OW-qPCA-secure. The QROM security reduction for $\mathrm{U}^{\not\perp}$ ($\mathrm{U}^{\perp}$) from the OW-qPCA (OW-qPVCA) security of PKE to the IND-CCA security of KEM is given in Section 4.2 (4.3). In Section 4.4, we show that $\mathrm{U}_m^{\not\perp}$ ($\mathrm{U}_m^{\perp}$) transforms any OW-CPA-secure or DS-secure (OW-VA-secure) DPKE into an IND-CCA-secure KEM in the QROM.

4.1 T: from OW-CPA to OW-qPCA in the QROM

To a public-key encryption PKE=(Gen, Enc, Dec) with message space $\mathcal{M}$ and randomness space R, and a hash function $G : \mathcal{M} \to \mathcal{R}$, we associate $\text{PKE}' = T[\text{PKE}, G]$. The algorithms of PKE′=(Gen,Enc',Dec') are defined in Fig. 10.

Theorem 3 (PKE OW-CPA $\overset{QROM}{\Rightarrow}$ PKE′ OW-qPCA). *If* PKE *is* δ-correct, *for any OW-qPCA* $\mathcal{B}$ *against* PKE′, *issuing at most* q_G *quantum queries to the random oracle* G *and at most* q_P *quantum queries to the plaintext checking oracle* Pco, *there exists a OW-CPA adversary* $\mathcal{A}$ *against PKE such that* $\texttt{Adv}_{\text{PKE}'}^{\text{OW}-q\text{PCA}}(\mathcal{B}) \leq 2q_G \cdot \sqrt{\delta} + (1 + 2q_G) \cdot \sqrt{\texttt{Adv}_{\text{PKE}}^{\text{OW-CPA}}(\mathcal{A})}$ *and the running time of* $\mathcal{A}$ *is about that of* $\mathcal{B}$.

The proof is essentially the same as the one of [7, Theorem 4.4] except the argument about the difference in $\mathcal{B}$'s success probability between game G_0 and game G_1. Game G_0 is exactly the original OW-qPCA game. In game G_1, the Pco oracle is replaced by a simulation where $Enc(pk, m; G(m)) = ?c$ is returned for the query input (m, c). As pk is public and G is a quantum random oracle, such a Pco simulation can be queried on a quantum superposition of inputs. Note that G_0 and G_1 are indistinguishable unless there exits an adversary who issuing at most q_G queries to G can distinguish N_1 from a constant function N_2 that always outputs 0 for any input, where $N_1(m) = 0$ if $Dec(sk, Enc(pk, m; G(m))) = m$, and otherwise $N_1(m) = 1$. Thus, using Lemma 2, we can obtain that $\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_G \cdot \sqrt{\delta}$. Then, following the security proof of [7, Theorem 4.4], we can easily prove Theorem 3.

```
Enc'(pk, m)
    $c = Enc(pk, m; G(m))$
    return $c$
```

```
Dec'(sk, c)
    $m' := Dec(sk, c)$
    if $Enc(pk, m'; G(m')) = c$
        return $m'$
    else return $\perp$
```

Fig. 10: OW-qPCA-secure $\text{PKE}' = T[\text{PKE}, G]$

4.2 $\text{U}^{\not\perp}$: from OW-qPCA to IND-CCA in the QROM

To a public-key encryption PKE′=(Gen', Enc', Dec') and a hash function H, we associate KEM-III = $U^{\not\perp}[\text{PKE}', H]$. The algorithms of KEM-III=(Gen,$Encaps$,$Decaps$) are defined in Fig. 11.

```
$Gen$
    $(pk, sk) \leftarrow Gen'$
    $s \xleftarrow{\$} \mathcal{M}$
    $sk' := (sk, s)$
    **return** $(pk, sk')$
```

```
$Encaps(pk)$
    $m \xleftarrow{\$} \mathcal{M}$
    $c \leftarrow Enc'(pk, m)$
    $K := H(m, c)$
    **return** $(K, c)$
```

```
$Decaps(sk', c)$
    Parse $sk' = (sk, s)$
    $m' := Dec'(sk, c)$
    **if** $m' = \perp$
        **return** $K := H(s, c)$
    **else return**
        $K := H(m', c)$
```

Fig. 11: IND-CCA-secure KEM-III = $U^{\not\perp}[\text{PKE}', H]$

Theorem 4 (PKE′ OW-qPCA $\overset{QROM}{\Rightarrow}$ KEM-III IND-CCA). *If* PKE′ *is* δ-correct*, for any IND-CCA $\mathcal{B}$ against KEM-III, issuing at most q_D (classical) queries to the decapsulation oracle* Decaps *and at most q_H queries to the quantum random oracle H, there exists a quantum OW-qPCA adversary $\mathcal{A}$ against PKE′ that makes at most q_H queries to the* Pco *oracle such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-III}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW}-q\text{PCA}}_{\text{PKE}'}(\mathcal{A})}$ *and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

The proof skeleton of Theorem 4 is essentially the same as the one of Theorem 1. Here, we briefly state the main differences. The complete proof is presented in Appendix E.

In KEM-I, the randomness used in the encryption algorithm is determined by the random oracle G. Given a plaintext m, we can deterministically evaluate the ciphertext $c = Enc(pk, m; G(m))$. Thus, we can divide H-query inputs (m, c) into two categories by judging if (m, c) is a matching plaintex-ciphertext pair (i.e., $c = Enc(pk, m; G(m))$) or not. In KEM-III, the encryption algorithm may be probabilistic, thus the above method will be invalid. Instead, we can query the Pco oracle to judge whether (m, c) is a matching plaintex-ciphertext pair. If $\text{Pco}(m, c) = 1$, the random oracle H returns $H_1(c)$, otherwise $H_3(m, c)$. To simulate the random oracle H, we make quantum queries to Pco (this is the reason why we require the scheme PKE′ to be OW-qPCA-secure). Note that it is impossible that $\text{Pco}(m_1, c) = \text{Pco}(m_2, c) = 1$ for $m_1 \neq m_2$. Thus, H is perfectly simulated without introducing the δ term. As $\mathcal{B}$'s queries to H can only help him get access to H_1 at c such that $Dec'(sk, c) = \hat{m}$ for some $\hat{m} \neq \perp$, the Decaps oracle can be perfectly simulated by H_1. Therefore, different from the security bounds obtained in Theorem 1 and Theorem 2, the δ term is removed with the OW-qPCA security of underlying PKE.

```
Gen
    $(pk, sk) \leftarrow Gen'$
    return $(pk, sk)$

Encaps(pk)
    $m \xleftarrow{\$} \mathcal{M}$
    $c \leftarrow Enc'(pk, m)$
    $K := H(m, c)$
    return $(K, c)$

Decaps^⊥(sk, c)
    $m' := Dec'(sk, c)$
    if $m' = \perp$
        return $\perp$
    else return
        $K := H(m', c)$
```

Fig. 12: IND-CCA-secure KEM-IV = $U^{\perp}[\text{PKE}', H]$

4.3 U$^{\perp}$: from OW-qPVCA to IND-CCA in the QROM

To a public-key encryption PKE$'$=(Gen', Enc', Dec') and a hash function H, we associate KEM-IV = $U^{\perp}[\text{PKE}', H]$. We remark that U$^{\perp}$ is essentially the transformation [6, Table 2], a KEM variant of the REACT/GEM transformations [54, 55]. The algorithms of KEM-IV=(Gen,$Encaps$,$Decaps^{\perp}$) are defined in Fig. 12.

Theorem 5 (PKE$'$ OW-qPVCA $\overset{QROM}{\Rightarrow}$ KEM-IV IND-CCA). *If* PKE$'$ *is* δ-correct*, for any IND-CCA $\mathcal{B}$ against KEM-IV, issuing at most q_D (classical) queries to the decapsulation oracle* Decaps *and at most q_H queries to the quantum random oracle H, there exists a OW-qPVCA adversary $\mathcal{A}$ against PKE$'$ that makes at most q_H queries to the* Pco *oracle and at most q_D queries to the* Val *oracle such that $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-IV}}(\mathcal{B}) \leq 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW}-q\text{PVCA}}_{\text{PKE}'}(\mathcal{A})}$ and the running time of $\mathcal{A}$ is about that of $\mathcal{B}$.*

The only difference between KEM-III and KEM-IV is the response to the invalid ciphertext in the decapsulation algorithm. When the ciphertext c is invalid, the decapsulation algorithm in KEM-III returns a pseudorandom key related to c. In this way, whatever the ciphertext (valid or invalid) is submitted, the return values have the same distribution. As a result, $\mathcal{A}$ can easily simulate the decapsulation oracle Decaps without recognition of the invalid ciphertexts. While the decapsulation algorithm in KEM-IV returns $\perp$ when the submitted c is invalid. Thus, in order to simulate Decaps, $\mathcal{A}$ needs to judge if the ciphertext c is valid. As we assume that the scheme PKE$'$ is OW-qPVCA-secure, $\mathcal{A}$ can query the Val oracle to fulfill such a judgement. Then, it is easy to verify that by using the same proof method in Theorem 4 we can obtain the desired security bound.

4.4 U$^{\not\perp}_m$/$U^{\perp}_m$: from OW-CPA/OW-VA to IND-CCA for Deterministic Encryption in the QROM

The transformation U$^{\not\perp}_m$ (U$^{\perp}_m$) is a variant of U$^{\not\perp}$ (U$^{\perp}$) that derives the KEM key as $K = H(m)$ instead of $K = H(m, c)$. To a deterministic public-key encryption scheme PKE$'$ = (Gen', Enc', Dec') with message space $\mathcal{M}$, a hash function

$H : \mathcal{M} \to \mathcal{K}$, and a pseudorandom function f with key space $\mathcal{K}^{prf}$, we associate KEM-V=$\mathrm{U}^{\not\perp}_{m}$[PKE$'$,$H$,$f$] and KEM-VI=$\mathrm{U}^{\perp}_{m}$[PKE$'$,$H$] shown in Fig. 13 and Fig. 14, respectively.

```
Gen
    $(pk, sk) \leftarrow Gen'$
    $k \xleftarrow{\$} \mathcal{K}^{prf}$
    $sk' := (sk, k)$
    return $(pk, sk')$

Encaps(pk)
    $m \xleftarrow{\$} \mathcal{M}$
    $c := Enc'(pk, m)$
    $K := H(m)$
    return $(K, c)$

Decaps(sk', c)
    Parse $sk' = (sk, k)$
    $m' := Dec'(sk, c)$
    if $Enc'(pk, m') = c$
        return $K := H(m')$
    else return
        $K := f(k, c)$
```

Fig. 13: IND-CCA-secure KEM-V=$\mathrm{U}^{\not\perp}_{m}$[PKE$'$,$H$,$f$]

```
Gen
    $(pk, sk) \leftarrow Gen'$
    return $(pk, sk)$

Encaps(pk)
    $m \xleftarrow{\$} \mathcal{M}$
    $c := Enc'(pk, m)$
    $K := H(m)$
    return $(K, c)$

Decaps(sk, c)
    $m' := Dec(sk, c)$
    if $Enc'(pk, m') = c$
        return $K := H(m')$
    else return $\perp$
```

Fig. 14: IND-CCA-secure KEM-VI=$\mathrm{U}^{\perp}_{m}$[PKE$'$,H]

We note that for a deterministic PKE scheme the OW-PCA security is equivalent to the OW-CPA security as we can simulate the Pco oracle via re-encryption during the proof. Thus, combing the proofs of Theorem 2, Theorem 4, Theorem 5 and [12, Theorem 4.1], we can easily obtain the following two theorems.

Theorem 6 (PKE$'$ OW-CPA $\overset{QROM}{\Rightarrow}$ KEM-V IND-CCA). *If* PKE$'$ *is* δ-*correct and deterministic, for any IND-CCA $\mathcal{B}$ against KEM-V, issuing at most q_E quantum queries to the encryption oracle*[11]*, at most q_D (classical) queries to the decapsulation oracle* Decaps *and at most q_H quantum queries to the random oracle H, there exist a quantum OW-CPA adversary $\mathcal{A}$ against* PKE$'$*, an adversary $\mathcal{A}'$ against the security of* PRF *with at most q_D classical queries and an adversary $\mathcal{C}$ against the $U_{\mathcal{M}}$-DS security with a simulator S of* PKE$'$ *($U_{\mathcal{M}}$ is the uniform distribution in $\mathcal{M}$) such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-V}}(\mathcal{B}) \leq \mathtt{Adv}_{\text{PRF}}(\mathcal{A}') +$

[11] For the deterministic scheme PKE$'$, given public key pk, quantum adversary $\mathcal{B}$ can execute the encryption algorithm Enc' in a quantum computer.

$4q_E\sqrt{\delta} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW-CPA}}_{\text{PKE}'}(\mathcal{A})}$ *and* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-V}}(\mathcal{B}) \leq \mathtt{Adv}_{\text{PRF}}(\mathcal{A}') + 4q_E\sqrt{\delta} + \mathtt{Adv}^{\text{DS-IND}}_{\text{PKE}',U_{\mathcal{M}},S}(\mathcal{C}) + \textsc{Disj}_{\text{PKE}',S}$*, and the running time of* $\mathcal{A}$ *(*$\mathcal{C}$*) is about that of* $\mathcal{B}$*.*

Theorem 7 (PKE′ OW-VA $\overset{QROM}{\Rightarrow}$ KEM-VI IND-CCA). *If* PKE′ *is* δ-correct *and deterministic, for any IND-CCA* $\mathcal{B}$ *against KEM-VI, issuing at most* q_E *quantum queries to the encryption oracle, at most* q_D *(classical) queries to the decapsulation oracle* Decaps *and at most* q_H *quantum queries to the random oracle* H*, there exists a quantum OW-VA adversary* $\mathcal{A}$ *against* PKE′ *who makes at most* q_D *queries to the* Val *oracle such that* $\mathtt{Adv}^{\text{IND-CCA}}_{\text{KEM-VI}}(\mathcal{B}) \leq 2q_E\sqrt{\delta} + 2q_H \cdot \sqrt{\mathtt{Adv}^{\text{OW-VA}}_{\text{PKE}'}(\mathcal{A})}$ *and the running time of* $\mathcal{A}$ *is about that of* $\mathcal{B}$*.*

Acknowledgements. We would like to thank anonymous reviews of Crypto 2018, Keita Xagawa, Takashi Yamakawa, Jiang Zhang, and Edoardo Persichetti for their helpful comments and suggestions. This work is supported by the National Key Research and Development Program of China (No. 2017YFB0802000), the National Natural Science Foundation of China (No. U1536205, 61472446, 61701539, 61501514), and the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing (No. 2016A01).

References

1. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing **33**(1) (2003) 167–226
2. Boyd, C., Cliff, Y., Gonzalez Nieto, J., Paterson, K.G.: Efficient one-round key exchange in the standard model. In Mu, Y., Susilo, W., Seberry, J., eds.: Information Security and Privacy, 13th Australasian Conference – ACISP 2008. Volume 5107 of LNCS., Springer-Verlag (2008) 69–83
3. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. Designs, Codes and Cryptography **76**(3) (2015) 469–504
4. NIST: National institute for standards and technology. Post quantum crypto project (2017) `https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions`.
5. Rackoff, C., Simon, D.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Feigenbaum, J., ed.: Advances in Cryptology – CRYPTO 1991. Volume 576 of LNCS., Springer (1992) 433–444
6. Dent, A.W.: A designer's guide to KEMs. In Paterson, K.G., ed.: Cryptography and Coding: 9th IMA International Conference. Volume 2898 of LNCS., Springer-Verlag (2003) 133–151
7. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In Kalai, Y., Reyzin, L., eds.: Theory of Cryptography - 15th International Conference – TCC 2017. Volume 10677 of LNCS., Springer (2017) 341–371

8. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In Wiener, M.J., ed.: Advances in Cryptology – CRYPTO 1999. Volume 99 of LNCS., Springer (1999) 537–554
9. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of cryptology **26**(1) (2013) 1–22
10. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V., eds.: Proceedings of the 1st ACM Conference on Computer and Communications Security – CCS 1993, ACM (1993) 62–73
11. Boneh, D., Dagdelen, O., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In Lee, D.H., Wang, X., eds.: Advances in Cryptology – ASIACRYPT 2011. Volume 7073 of LNCS., Springer (2011) 41–69
12. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Nielsen, J.B., Rijmen, V., eds.: Advances in Cryptology – EUROCRYPT 2018. Volume 10822 of LNCS. (2018) 520–551
13. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. Technical report, Cryptology ePrint Archive, Report 2017/1096, 2017. https://eprint. iacr. org/2017/1096
14. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Hirt, M., Smith, A.D., eds.: Theory of Cryptography Conference – TCC 2016-B. Volume 9986 of LNCS., Springer (2016) 192–216
15. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In Santis, A.D., ed.: Advances in Cryptology – EUROCRYPT 1994. Volume 950 of LNCS., Springer (1994) 92–111
16. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. In Kilian, J., ed.: Advances in Cryptology – CRYPTO 2001. Volume 2139 of LNCS., Springer (2001) 260–274
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. In Miller, G.L., ed.: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing – STOC 1996, ACM (1996) 212–219
18. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: High-speed key encapsulation from NTRU. In Fischer, W., Homma, N., eds.: Cryptographic Hardware and Embedded Systems – CHES 2017. Volume 10529 of LNCS., Springer-Verlag (2017) 232–252
19. Hamburg, M.: Module-LWE: The three bears. Technical report, https://www.shiftleft.org/papers/threebears/
20. Ding, J.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive **2012** (2012) 688
21. Peikert, C.: Lattice cryptography for the internet. In Mosca, M., ed.: International Workshop on Post-Quantum Cryptography – PQCrypto 2014. Volume 8772 of LNCS., Springer (2014) 197–219
22. Bos, J.W., Costello, C., Naehrig, M., Stebila, D.: Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In: 2015 IEEE Symposium on Security and Privacy – SP 2015. (2015) 553–570
23. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange - a new hope. In Holz, T., Savage, S., eds.: 25th USENIX Security Symposium – USENIX Security 2016, USENIX Association (2016) 327–343
24. Bos, J.W., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key

exchange from LWE. In Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S., eds.: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security – CCS 2016, ACM (2016) 1006–1018

25. Cheon, J.H., Kim, D., Lee, J., Song, Y.S.: Lizard: Cut off the tail! practical post-quantum public-key encryption from LWE and LWR. Technical report, Cryptology ePrint Archive, Report 2016/1126, 2016. http://eprint.iacr.org/2016/1126
26. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Stehlé, D.: Crystals–kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy – EuroSP 2018 (to appear). (2018)
27. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In Buhler, J., ed.: Algorithmic Number Theory, Third International Symposium, ANTS-III. Volume 1423 of LNCS., Springer (1998) 267–288
28. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU Prime: reducing attack surface at low cost. In Adams, C., Camenisch, J., eds.: Selected Areas in Cryptography – SAC 2017. Volume 10719 of LNCS., Springer (2017) 235–260
29. Barreto, P.S., Gueron, S., Gueneysu, T., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P.: CAKE: Code-based algorithm for key encapsulation. In O'Neill, M., ed.: Cryptography and Coding - 16th IMA International Conference – IMACC 2017. Volume 10655 of LNCS., Springer (2017) 207–226
30. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.: MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In: Proceedings of the 2013 IEEE International Symposium on Information Theory (ISIT), IEEE (2013) 2069–2073
31. Howgrave-Graham, N., Nguyen, P.Q., Pointcheval, D., Proos, J., Silverman, J.H., Singer, A., Whyte, W.: The impact of decryption failures on the security of NTRU encryption. In Boneh, D., ed.: Advances in Cryptology – CRYPTO 2003. Volume 2729 of LNCS., Springer (2003) 226–246
32. Guo, Q., Johansson, T., Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In Cheon, J.H., Takagi, T., eds.: Advances in Cryptology – ASIACRYPT 2016. Volume 10031 of LNCS., Springer (2016) 789–815
33. Bernstein, D.J., Bruinderink, L.G., Lange, T., Panny, L.: HILA5 pindakaas: On the CCA security of lattice-based encryption with error correction. In Joux, A., Nitaj, A., Rachidi, T., eds.: Progress in Cryptology – AFRICACRYPT 2018. Volume 10831 of LNCS., Springer (2018) 203–216
34. Saarinen, M.J.O.: HILA5: On reliability, reconciliation, and error correction for Ring-LWE encryption. In: Selected Areas in Cryptography – SAC 2017. Volume 10719 of LNCS., Springer (2017) 192–212
35. Persichetti, E.: Secure and anonymous hybrid encryption from coding theory. In Gaborit, P., ed.: Post-Quantum Cryptography - 5th International Workshop – PQCrypto 2013. Volume 7932 of LNCS., Springer (2013) 174–187
36. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In Dwork, C., ed.: Proceedings of the 40th Annual ACM Symposium on Theory of Computing – STOC 2008, ACM (2008) 197–206
37. Mceliece, R.J.: A public-key cryptosystem based on algebraic. DSN progress report **42-44** (1978) 114–116
38. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory **15**(2) (1986) 159–166
39. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM (JACM) **56**(6) (2009) 34

40. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-Based encryption. In Kiayias, A., ed.: The Cryptographers' Track at the RSA Conference Topics in Cryptology – CT-RSA 2011. Volume 6558 of LNCS., Springer (2011) 319–339
41. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In Gilbert, H., ed.: Advances in Cryptology – EUROCRYPT 2010. Volume 6110 of LNCS., Springer (2010) 1–23
42. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for Ring-LWE cryptography. In Johansson, T., Nguyen, P.Q., eds.: Advances in Cryptology – EUROCRYPT 2013. Volume 7881 of LNCS., Springer (2013) 35–54
43. Google: pqc-forum. LIMA (2018) `https://groups.google.com/a/list.nist.gov/forum/#!topic/pqc-forum/6khIivE2KE0`.
44. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM **62**(6) (2015) 49:1–49:76
45. Albrecht, M.R., Orsini, E., Paterson, K.G., Peer, G., Smart, N.P.: Tightly secure Ring-LWE based key encapsulation with short ciphertexts. In Foley, S.N., Gollmann, D., Snekkenes, E., eds.: 22nd European Symposium on Research in Computer Security – ESORICS 2017. Volume 10492 of LNCS., Springer (2017) 29–46
46. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum private queries. Physical review letters **100**(23) (2008) 230502
47. De Martini, F., Giovannetti, V., Lloyd, S., Maccone, L., Nagali, E., Sansoni, L., Sciarrino, F.: Experimental quantum private queries with linear optics. Physical Review A **80**(1) (2009) 010302
48. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In Safavi-Naini, R., Canetti, R., eds.: Advances in Cryptology – CRYPTO 2012. Volume 7417 of LNCS., Springer (2012) 758–775
49. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: 55th IEEE Annual Symposium on Foundations of Computer Science – FOCS 2014, IEEE (2014) 474–483
50. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In Cheng, C., Chung, K., Persiano, G., Yang, B., eds.: Public-Key Cryptography–PKC 2016. Volume 9614 of LNCS., Springer (2016) 387–416
51. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In Canetti, R., Garay, J.A., eds.: Advances in Cryptology – CRYPTO 2013. Volume 8043 of LNCS., Springer (2013) 361–379
52. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Information & Computation **15**(7-8) (2015) 557–567
53. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In Oswald, E., Fischlin, M., eds.: Advances in Cryptology – EUROCRYPT 2015. Volume 9057 of LNCS., Springer (2015) 755–784
54. Okamoto, T., Pointcheval, D.: REACT: Rapid enhanced-security asymmetric cryptosystem transform. In Naccache, D., ed.: Topics in Cryptology – CT-RSA 2001. Volume 2020 of LNCS., Springer (2001) 159–174
55. Jean-Sébastien, C., Handschuh, H., Joye, M., Paillier, P., Pointcheval, D., Tymen, C.: GEM: A generic chosen-ciphertext secure encryption method. In Preneel, B., ed.: Topics in Cryptology – CT-RSA 2002. Volume 2271 of LNCS., Springer (2002) 263–276
56. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Number 2. Cambridge University Press (2000)
57. Unruh, D.: Quantum position verification in the random oracle model. In Garay, J.A., Gennaro, R., eds.: Advances in Cryptology – CRYPTO 2014. Volume 8617 of LNCS., Springer (2014) 1–18

A Quantum Computation

We give a short introduction to quantum computation. For a more thorough discussion, please see [56].

A quantum system A is a complex Hilbert space $\mathcal{H}$ with an inner product $\langle\cdot|\cdot\rangle$. The state of a quantum system is given by a vector $|\Psi\rangle$ of unit norm ($\langle\Psi|\Psi\rangle = 1$). Given quantum systems A and B over spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, we define the joint or composite quantum system through the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$. The product state of $|\varphi_A\rangle \in \mathcal{H}_A$ and $|\varphi_B\rangle \in \mathcal{H}_B$ is denoted by $|\varphi_A\rangle \otimes |\varphi_B\rangle$ or simply $|\varphi_A\rangle|\varphi_B\rangle$. A n-qubit system lives in the joint quantum system of n two-dimensional Hilbert spaces. The standard orthonormal computational basis $B = \{|x\rangle\}$ for such a system is given by $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$ for $x = x_1 \cdots x_n$. Any (classical) bit string x is encoded into a quantum state by $|x\rangle$. Denote $TD(|\Psi\rangle, |\varphi\rangle)$ as the trace distance between quantum states $|\Psi\rangle$ and $|\varphi\rangle$.

Quantum measurement. Given a state $|\varphi\rangle$, we can measure $|\varphi\rangle$ in the basis B, obtaining the value x with probability $|\langle x|\varphi\rangle|^2$. Thus, to each $|\varphi\rangle$, we associate a distribution D_φ where $D_\varphi(x) = |\langle x|\varphi\rangle|^2$. The normalization constant and the fact that B is an orthonormal basis ensure that D_φ is exactly a valid distribution. After measurement, the system is in state $|x\rangle$.

Quantum algorithm. A quantum algorithm A over a Hilbert space $\mathcal{H}$ with a standard orthonormal basis B is specified by unitary transformation U. The input to A is the initial state $|x_0\rangle$. Then U is applied to the system, and the final state is obtained $|\varphi\rangle = U|x_0\rangle$. At last, A's output is obtained by performing a measurement on $|\varphi\rangle$.

Quantum algorithm usually operates on a product space $S \otimes K \otimes V$, where S represents the work space, K the input space, and V the output space. Given a function $H : K \to V$, define the standard orthonormal basis B as the set $|s, k, v\rangle$ for $s \in S$, $k \in K$, and $v \in V$. Define the unitary transformation O_H over the Hilbert space spanned by B as the transformation that takes $|s, k, v\rangle$ into $|s, k, v \oplus H(k)\rangle$. O_H is unitary, its own inverse, and Hermitian.

A quantum algorithm A making q quantum queries to H is then specified by a sequence of unitary transformations $\mathrm{U}_0, \ldots, U_q$. The evaluation of A then consists of alternately applying U_i and O_H to the initial state $\mathrm{U}_0|x_0\rangle$. The final state of the algorithm is

$$U_q O_H \ldots U_1 O_H U_0 |x_0\rangle.$$

We say that a quantum algorithm is efficient if q is a polynomial, and all the U_is are composed of a polynomial number of universal basis gates (the Hadamard, CNOT, and phase shift gates are commonly used).

B Proof of Lemma 3

Proof. Assume that A uses three quantum systems S, K and V for its state, oracle input and oracle output, where K has two subsystems $K = K_1 \otimes K_2$ and V

has two subsystems $V = V_1 \otimes V_2$. Let $x_i, y_i \in \{0, 1\}$ ($i \in \{1, 2, \dots, q\}, q = q_1 + q_2$) such that $\sum x_i = q_1, \sum y_i = q_2, x_i + y_i = 1$. Then an execution of A leads to the final state

$$|\Psi_q\rangle := \prod_{i=1}^{q} (U_2^i O_2^{y_i} U_1^i O_1^{x_i}) |\Psi_0\rangle,$$

where $|\Psi_0\rangle$ is the initial state, U_1 and U_2 are A's state transition operations, O_1 and O_2 are the oracle queries such that $O_1|s, k_1, k_2, v_1, v_2\rangle := |s, k_1, k_2, v_1 \oplus \mathcal{O}_1(k_1), v_2\rangle, O_2|s, k_1, k_2, v_1, v_2\rangle := |s, k_1, k_2, v_1, v_2 \oplus \mathcal{O}_2(k_2)\rangle$. A's output is produced by applying a measurement M to A's final state. Then,

$$\Pr[E_1] = \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha b,$$

where α is the probability of each particular pair $(\mathcal{O}_1, \mathcal{O}_2, inp, x)y$ and $b = \Pr[M$ outputs 1 on state $|\Psi_q\rangle$.

Reprogram $\mathcal{O}_1$ at x. Denote O_1' as the function that $O_1'(x) = y$ and $O_1' = O_1$ everywhere else. Then, the final state becomes

$$|\Psi_q'\rangle := \prod_{i=1}^{q} (U_2^i O_2^{y_i} U_1^i {O_1'}^{x_i}) |\Psi_0\rangle.$$

Thus,

$$\Pr[E_2] = \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha b',$$

where $b' = \Pr[M$ outputs 1 on state$|\Psi_q'\rangle$.

According to [56, Theorem 9.1], we know that

$$|\Pr[E_1] - \Pr[E_2]| \leq \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha \, |b - b'| \leq \sum_{(\mathcal{O}_1, \mathcal{O}_2, inp, x)y} \alpha D_q, \tag{1}$$

where $D_q := TD(|\Psi_q\rangle, |\Psi_q'\rangle)$ is the trace distance between quantum states $|\Psi_q\rangle$ and $|\Psi_q'\rangle$.

Note the fact that the difference between $|\Psi_q\rangle$ and $|\Psi_q'\rangle$ just comes from the difference between O_1 and O_1'. Thus, the formulas of $|\Psi_q\rangle$ and $|\Psi_q'\rangle$ can be simplified by $|\Psi_q\rangle := \prod\limits_{i=1}^{q_1} (U_i O_1) U_0 |\Psi_0\rangle$ and $|\Psi_q'\rangle := \prod\limits_{i=1}^{q_1} (U_i O_1') U_0 |\Psi_0\rangle$, where U_i is the product of the transformations between the i-th O_1 and $(i+1)$-th O_1. Specifically, $\mathrm{U}_0 = \prod_{l < j_1} (U_2^l O_2^{y_l} U_1^l {O_l}^{x_l})$, $\mathrm{U}_i = \prod_{j_i \leq l < j_{i+1}} (U_2^l O_2^{y_l} U_1^l {O_l}^{x_l}) \times {O_l}^{x_{j_i}}$ $(1 \leq i < q_1)$ and $\mathrm{U}_{q_1} = \prod_{l > j_{q_1}} (U_2^l O_2^{y_l} U_1^l {O_l}^{x_l}) \times U_2^{j_{q_1}} O_2^{y_{j_{q_1}}} U_1^{j_{q_1}}$ ($j_i \in \{i : x_i = 1\}$, $j_1 < j_2 \ldots < j_{q_1}$).

Define $|\Phi_i\rangle := \prod\limits_{j=1}^{i} (U_j O_1) U_0 |\Psi_0\rangle$ and $|\Phi_i'\rangle := \prod\limits_{j=1}^{i} (U_j O_1') U_0 |\Psi_0\rangle$ ($i \in \{1, \dots, q_1\}$). Then, $|\Phi_{q_1}\rangle {=} |\Psi_q\rangle$, $|\Phi_{q_1}'\rangle {=} |\Psi_q'\rangle$ and $D_q = TD(|\Psi_q\rangle, |\Psi_q'\rangle) = TD(|\Phi_{q_1}\rangle, |\Phi_{q_1}'\rangle)$.

Describe B as follows: $B^{\mathcal{O}_1,\mathcal{O}_2}(inp,x)$ picks $i \xleftarrow{\$} \{1,\ldots,q_1\}$ and $y \xleftarrow{\$} \{0,1\}^m$, measures the quantum system K_1 of the state $|\Phi'_{i-1}\rangle$, and outputs the result. Thus,

$$P_B := \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \frac{\alpha}{q_1} \left\|Q_x|\Phi'_{i-1}\rangle\right\|^2,$$

where Q_x is the projector projecting K_1 onto $|x\rangle$ (i.e., $Q_x = I\otimes|x\rangle\langle x|\otimes I\otimes I$).

In fact, we can view K_2 and V_2 as the subsystems of the auxiliary quantum system S (that is, $\mathcal{O}_2$ is redundant). Then, according to the proof of the OW2H lemma in [44, Lemma 6.2], we can directly obtain $|\Pr[E_1] - \Pr[E_2]| \leq 2q_1\sqrt{P_B}$. But, for completeness, we also preset the complete proof here.

Let $D_i := TD(|\Phi_i\rangle, |\Phi'_i\rangle)$. $D_0 = TD(U_0|\Phi_0\rangle, U_0|\Phi_0\rangle)$ and

$$\begin{aligned} D_i &= TD(U_iO_1|\Phi_{i-1}\rangle, U_iO'_1|\Phi'_{i-1}\rangle) \\ &\leq TD(U_iO_1|\Phi_{i-1}\rangle, U_iO_1|\Phi'_{i-1}\rangle) + TD(U_iO_1|\Phi'_{i-1}\rangle, U_iO'_1\Phi'_{i-1}\rangle) \\ &\leq D_{i-1} + TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle). \end{aligned}$$

Hence,

$$D_q \leq \sum_{i=1}^{q} TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle). \tag{2}$$

Let $V_y|s,k_1,k_2,v_1,v_2\rangle := |s,k_1,k_2,v_1\oplus y,v_2\rangle$. Then $O'_1 = O_1(1-Q_x)+V_yQ_x$. By using [57, Lemma 12], we can get that

$$\begin{aligned} &TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle) \\ &= TD(O_1(1-Q_x)|\Phi'_{i-1}\rangle + O_1Q_x|\Phi'_{i-1}\rangle, O_1(1-Q_x)|\Phi'_{i-1}\rangle + V_yQ_x|\Phi'_{i-1}\rangle) \\ &\leq 2\left\|O_1Q_x|\Phi'_{i-1}\rangle\right\| = 2\left\|Q_x|\Phi'_{i-1}\rangle\right\|. \end{aligned} \tag{3}$$

Combing the equations (1, 2, 3), we obtain that

$$\begin{aligned} |\Pr[E_1] - \Pr[E_2]| &\leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)y} \alpha D_q \leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \alpha TD(O_1|\Phi'_{i-1}\rangle, O'_1|\Phi'_{i-1}\rangle) \\ &\leq \sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \alpha 2\left\|Q_x|\Phi'_{i-1}\rangle\right\| \\ &\overset{(*)}{\leq} 2q_1\sqrt{\sum_{(\mathcal{O}_1,\mathcal{O}_2,inp,x)yi} \frac{\alpha}{q_1}\left\|Q_x|\Phi'_{i-1}\rangle\right\|^2} = 2q_1\sqrt{P_B}, \end{aligned}$$

where (*) uses Jensen's inequality. □

C Proof of Lemma 4

Proof. Assume that A uses three quantum systems S, K and V for its state, oracle input and oracle output, where K has two subsystems $K = K_1 \otimes K_2$. K_1, K_2 and V have n_1, n_2 and m qubits respectively. Then an execution of

A leads to the final state $(UO_H)^q|\Psi_{xH'}\rangle$, where $|\Psi_{xH'}\rangle$ is the initial state, $O_H \mid s, k_1 \otimes k_2, v\rangle := |s, k_1 \otimes k_2, v \oplus H(k_1, k_2)\rangle$, and U is A's state transition operation. We assume that all the transition operations U_i are identical and equal to U (the proof in the general case is essentially identical). A's output is produced by applying a measurement M to A's final state.

Define $|\Psi^i_{HxH'}\rangle := (UO_H)^i|\Psi_{xH'}\rangle$. Then, we can obtain

$$\Pr[E_1] = \sum_{HxH'} \alpha b_{HxH'},$$

where $b_{HxH'} = \Pr[M\ outputs\ 1\ on\ state\ |\Psi^q_{HxH'}\rangle]$, $\alpha = 2^{-m2^{(n_1+n_2)}-n_1-m2^{n_2}}$.

Reprogram H at $(x, \cdot)$. Denote $H_{xH'}$ as the function that $H_{xH'}(x, \cdot) = H'(\cdot)$ and $H_{xH'} = H$ everywhere else. Thus,

$$\Pr[E_2] = \sum_{HxH'} \alpha b_{H_{xH'}xH'}.$$

According to [56, Theorem 9.1], we know that

$$|\Pr[E_1] - \Pr[E_2]| \leq \sum_{HxH'} \alpha \left| b_{HxH'} - b_{H_{xH'}xH'} \right| \leq \sum_{HxH'} \alpha D_q, \tag{4}$$

where $D_i := TD(|\Psi^i_{HxH'}\rangle, |\Psi^i_{H_{xH'}xH'}\rangle)$ is the trace distance between quantum states $|\Psi^i_{HxH'}\rangle$ and $|\Psi^i_{H_{xH'}xH'}\rangle$.

Note that $D_0 = TD(|\Psi_{xH'}\rangle, |\Psi_{xH'}\rangle) = 0$ and

$$\begin{aligned}
D_i &= TD(UO_H|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{H_{xH'}xH'}\rangle) \\
&\leq TD(UO_H|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle) + TD(UO_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle, UO_{H_{xH'}}|\Psi^{i-1}_{H_{xH'}xH'}\rangle) \\
&\leq D_{i-1} + TD(O_H|\Psi^{i-1}_{HxH'}\rangle, O_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle).
\end{aligned}$$

Hence,

$$D_q \leq \sum_{i=1}^{q} TD(O_H|\Psi^{i-1}_{HxH'}\rangle, O_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle) \tag{5}$$

Let $O_{H'}|a, k_1 \otimes k_2, v\rangle := |a, k_1 \otimes k_2, v \oplus H'(k_2)\rangle$. Q_x is the projector projecting K_1 onto $|x\rangle$ (i.e., $Q_x = I \otimes |x\rangle\langle x| \otimes I \otimes I$). Then, $O_{H_{xH'}} = O_H(1 - Q_x) + O_{H'}Q_x$. By using [57, Lemma 12], we can get that

$$\begin{aligned}
&TD(O_H|\Psi^{i-1}_{HxH'}\rangle, O_{H_{xH'}}|\Psi^{i-1}_{HxH'}\rangle) \\
&= TD(O_H(1 - Q_x)|\Psi^{i-1}_{HxH'}\rangle + O_H Q_x|\Psi^{i-1}_{HxH'}\rangle, O_H(1 - Q_x)|\Psi^{i-1}_{HxH'}\rangle + O_{H'}Q_x|\Psi^{i-1}_{HxH'}\rangle) \\
&\leq 2\left\|O_H Q_x|\Psi^{i-1}_{HxH'}\rangle\right\| = 2\left\|Q_x|\Psi^{i-1}_{HxH'}\rangle\right\|.
\end{aligned} \tag{6}$$

Combing the equations (4, 5, 6), we obtain that

$$|\Pr[E_1] - \Pr[E_2]| \leq \sum_{HxH'i} 2\alpha \left\|Q_x|\Psi^{i-1}_{HxH'}\rangle\right\| \overset{(*)}{\leq} 2q\sqrt{\sum_{HxH'i} \frac{\alpha}{q} \left\|Q_x|\Psi^{i-1}_{HxH'}\rangle\right\|^2},$$

where (*) uses Jensen's inequality.

Define algorithm B as follows: pick $i \xleftarrow{\$} \{1, \dots, q\}$, measure the quantum system K_1 of A's i-th query state $|\Psi^{i-1}_{HxH'}\rangle$, obtain $\hat{x}$ and output $\hat{x} =?x$. Thus, $\Pr[B \Rightarrow 1]$ is exactly $\sum_{HxH'i} \frac{\alpha}{q} \left\| Q_x |\Psi^{i-1}_{HxH'}\rangle \right\|^2$. Because x is chosen uniformly at random and independent from A's view, $\Pr[B \Rightarrow 1] = \frac{1}{2^{n_1}}$. Therefore,

$$|\Pr[E_1] - \Pr[E_2]| \leq 2q \frac{1}{\sqrt{2^{n_1}}}.$$

□

D Proof of Theorem 2

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-II, issuing at most q_D classical queries to Decaps, at most q_G queries to G and at most q_H queries to H. Consider the sequence of games given in Fig. 15 and Fig. 17. Let $\Omega_{H''}$ be the set of all functions $H'' : \mathcal{M} \to \mathcal{K}$ and we follow the same notations Ω_G, $\Omega_{G'}$, Ω_H and $\Omega_{H'}$ in the proof of Theorem 1.

```
GAMES $G_0 - G_6$
$(pk, sk') \leftarrow Gen'; G \xleftarrow{\$} \Omega_G$
$G' \xleftarrow{\$} \Omega_{G'}; G := G'$    $//G_2 - G_4$
$H_1 \xleftarrow{\$} \Omega_{H''}, H_2, H_3 \xleftarrow{\$} \Omega_{H'}$
$m^* \xleftarrow{\$} \mathcal{M}$
$r^* := G(m^*)$
$r^* \xleftarrow{\$} \mathcal{R}$    $//G_6$
$c^* := Enc(pk, m^*; r^*)$
$k_0^* := H(m^*)$
$k_0^* \xleftarrow{\$} \mathcal{K}$    $//G_6$
$k_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0, 1\}$
$b' \leftarrow \mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
return $b' =?b$
```

```
$H(m)$
return $H_1(m)$    $//G_0 - G_2$
$g(\cdot) := Enc(pk, \cdot; G(\cdot))//G_3 - G_6$
return $H_2(g(m))$    $//G_3 - G_6$
```

```
Decaps $(c \neq c^*)$    $//G_0 - G_3$
Parse $sk' = (sk, k)$
$m' := Dec(sk, c)$
if $Enc(pk, m'; G(m')) = c$
    $K := H(m')$
else return
    return $K := f(k, c)$    $//G_0$
    return $K := H_3(c)$    $//G_1 - G_3$
```

```
Decaps $(c \neq c^*)$    $//G_4 - G_6$
return $K := H_2(c)$
```

Fig. 15: Games $G_0 - G_6$ for the proof of Theorem 2

GAME G_0. Game G_0 is exactly the IND-CCA game,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathtt{Adv}_{\text{KEM-II}}^{\text{IND-CCA}}(\mathcal{B}).$$

GAME G_1. In game G_1, the DECAPS oracle is changed that the pseudorandom function f is replaced by a random function H_3. Thus, the private key k, contained in the secret key sk', is never used in G_1. Because $\mathcal{B}$'s queries to DECAPS are just classical, $\mathcal{B}$ can make classical queries to f at most q_D times. $\mathcal{B}$'s views in G_0 and G_1 are same unless there exists some adversary $\mathcal{A}'$ who can distinguish f from the random function H_3 with at most q_D classical queries. Then,

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \le \mathtt{Adv}_{\text{PRF}}(\mathcal{A}').$$

GAME G_2. In game G_2, we replace G by G' that uniformly samples from "good" randomness at random, i.e., $G' \xleftarrow{\$} \Omega_{G'}$. Essentially, the distinguishing problem between G_1 and G_2 is equivalent to the distinguishing problem between G and G'. Using the same method in the proof of Theorem 1, we obtain

$$\left|\Pr[G_1^{\mathcal{B}} \Rightarrow 1] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1]\right| \le 2q_G\sqrt{\delta}.$$

GAME G_3. In game G_3, H_1 is substituted with $H_2 \circ g$ $(g(\cdot) := Enc(pk, \cdot; G(\cdot)))$. Since the G in this game only samples "good" randomness, the function g is injective. Thus, $H_2 \circ g$ is a perfect random function. Therefore, G_2 and G_3 are statistically indistinguishable and we can obtain

$$\Pr[G_2^{\mathcal{B}} \Rightarrow 1] = \Pr[G_3^{\mathcal{B}} \Rightarrow 1].$$

GAME G_4. In game G_4, the DECAPS oracle is changed that it makes no use of the secret key sk' any more. When $\mathcal{B}$ queries the DECAPS oracle on c $(c \neq c^*)$, $K := H_2(c)$ is returned as the response. With the same analysis method in the proof of Theorem 1, we have

$$\Pr[G_3^{\mathcal{B}} \Rightarrow 1] = \Pr[G_4^{\mathcal{B}} \Rightarrow 1].$$

GAME G_5. In game G_5, we replace G' by G, that is, G in this game is reset to be an ideal random oracle. Then, similar to the case of G_1 and G_2, the distinguishing problem between G_4 and G_5 also can be converted to the distinguishing problem between G and G'. Thus, as the same case in the proof of Theorem 1, we have

$$\left|\Pr[G_4^{\mathcal{B}} \Rightarrow 1] - \Pr[G_5^{\mathcal{B}} \Rightarrow 1]\right| \le 2q_G\sqrt{\delta}.$$

GAME G_6. In game G_6, r^* and k_0^* are chosen uniformly at random from $\mathcal{R}$ and $\mathcal{K}$, respectively. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_6^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

```
$A^{G\times H_2^g, H_2'}(pk, m^*, (r^*, k_0^*))$
    $c^* := Enc(pk, m^*; r^*)$
    $k_1^* \xleftarrow{\$} \mathcal{K}$
    $b \xleftarrow{\$} \{0, 1\}$
    $b' \leftarrow \mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
    return $b' =?b$

$H(m)$
    return $H_2^g(m)$

$\textsc{Decaps}\ (c \neq c^*)$
    return $K := H_2'(c)$
```

Fig. 16: $A^{G\times H_2^g, H_2'}$ for the proof of Theorem 2.

```
GAMES $G_7$
    $i \xleftarrow{\$} \{1, \dots, q_G + q_H\}$
    $(pk, sk') \leftarrow Gen'$
    $G \xleftarrow{\$} \Omega_G; H_2 \xleftarrow{\$} \Omega_H'$
    $m^* \xleftarrow{\$} \mathcal{M}; r^* \xleftarrow{\$} \mathcal{R}$
    $c^* := Enc(pk, m^*; r^*)$
    $k^* \xleftarrow{\$} \mathcal{K}$
    run $\mathcal{B}^{G,H,\textsc{Decaps}}(pk, c^*, k^*)$
    until the $i-$th query to $G \times H$
    measure the argument $\hat{m}$
    return $\hat{m} =?m^*$

$H(m)$
    $g(\cdot) := Enc(pk, \cdot; G(\cdot))$
    return $H_2(g(m))$

$\textsc{Decaps}\ (c \neq c^*)$
    return $K := H_2(c)$
```

Fig. 17: Game G_7 for the proof of Theorem 2

Let $(G\times H_2^g)(m) = (G(m), H_2 \circ g(m))$. The number of total queries to $G\times H_2^g$ is at most $q_G + q_H$. Let H_2' be the function that $H_2'(g(m^*)) = \perp$ and $H_2' = H_2$ everywhere else.

Let $A^{G\times H_2^g, H_2'}$ be an oracle algorithm on input $(pk, m^*, (r^*, k_0^*))$ in Fig. 16. Sample G, H_2, H_2^g and pk in the same way as G_5 and G_6, i.e., $(pk, sk') \leftarrow Gen'$,$G \xleftarrow{\$} \Omega_G$, $H_2 \xleftarrow{\$} \Omega_{H'}$ and $H_2^g := H_2 \circ g$. Let $m^* \xleftarrow{\$} \mathcal{M}$.

Then, if $r^* := G(m^*)$ and $k_0^* := H_2^g(m^*)$, $A^{G\times H_2^g, H_2'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_5. And, if $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{G\times H_2^g, H_2'}$ on input $(pk, m^*, (r^*, k_0^*))$ perfectly simulates G_6.

Let $B^{G\times H_2^g, H_2'}$ be an oracle algorithm that on input (pk, m^*) does the following: pick $i \xleftarrow{\$} \{1, \dots, q_G + q_H\}$, $r^* \xleftarrow{\$} \mathcal{R}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{G\times H_2^g, H_2'}(pk, m^*, (r^*, k_0^*))$ until the i-th query to $G \times H_2^g$, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{G\times H_2^g, H_2'}$ makes less than i queries, output $\perp$). Define game G_7 as in Fig. 17. Then, $\Pr[B^{G\times H_2^g, H_2'} \Rightarrow m^*] = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = G \times H_2^g$, $\mathcal{O}_2 = H_2'$, $inp = pk$, $x = m^*$ and $y = (r^*, k_0^*)$, we have

$$\left|\Pr[G_5^{\mathcal{B}} \Rightarrow 1] - \Pr[G_6^{\mathcal{B}} \Rightarrow 1]\right| \leq 2(q_G + q_H)\sqrt{\Pr[G_7^{\mathcal{B}} \Rightarrow 1]}.$$

Then, we construct an adversary $\mathcal{A}$ against the OW-CPA security of PKE such that $\mathtt{Adv}_{\mathrm{PKE}}^{\text{OW-CPA}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input $(1^\lambda,\, pk,\, c)$ does the following:

1. Run the adversary $\mathcal{B}$ in game G_7.
2. Use a $2q_G$-wise independent function and a $2q_H$-wise independent function to simulate random oracles G and H_2 respectively. The random oracle H is simulated by $H_2 \circ g$. Use $G \times H$ to answer $\mathcal{B}$'s queries to both G and H.
3. Answer the decapsulation queries by using the Decaps oracle as in Fig. 17.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with $(c,\, k^*)$.
5. Select $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}$, measure the argument $\hat{m}$ of the i-th query to $G \times H$ and output $\hat{m}$.

It is obvious that $\mathtt{Adv}_{\mathrm{PKE}}^{\text{OW-CPA}}(\mathcal{A}) = \Pr[G_7^{\mathcal{B}} \Rightarrow 1]$. Combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}_{\text{KEM-II}}^{\text{IND-CCA}}(\mathcal{B}) \leq \mathtt{Adv}_{\mathrm{PRF}}(\mathcal{A}') + 4q_G \cdot \sqrt{\delta} + 2(q_H + q_G) \cdot \sqrt{\mathtt{Adv}_{\mathrm{PKE}}^{\text{OW-CPA}}(\mathcal{A})}.$$

□

E Proof of Theorem 4

Proof. Let $\mathcal{B}$ be an adversary against the IND-CCA security of KEM-III, issuing at most q_D queries to Decaps and at most q_H queries to H. We follow the same notations Ω_H and $\Omega_{H'}$ in the proof of Theorem 1. Consider the games in Fig. 18 and Fig. 20.

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

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \frac{1}{2}\right| = \mathtt{Adv}_{\text{KEM-III}}^{\text{IND-CCA}}(\mathcal{B}).$$

Game G_1. In game G_1, the Decaps oracle is changed that $H_2(c)$ is returned instead of $H(s, c)$ for an invalid encapsulation c. Considering that $\mathcal{B}$'s view is independent from (the uniform secret) s, we can use Lemma 4 to obtain

$$\left|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_H \cdot \frac{1}{\sqrt{\mathcal{M}}}.$$

```
GAMES $G_0 - G_4$
$(pk, sk') \leftarrow Gen'; G \xleftarrow{\$} \Omega_G$
$H_1, H_2 \xleftarrow{\$} \Omega_{H'}; H_3 \xleftarrow{\$} \Omega_H$
$m^* \xleftarrow{\$} \mathcal{M}$
$c^* \leftarrow Enc(pk, m^*)$
$k_0^* := H(m^*, c^*)$
$k_0^* \xleftarrow{\$} \mathcal{K}$            $//G_4$
$k_1^* \xleftarrow{\$} \mathcal{K}$
$b \xleftarrow{\$} \{0, 1\}$
$b' \leftarrow B^{G,H,\textsc{Decaps}}(pk, c^*, k_b^*)$
return $b' =?b$
```

```
$H(m, c)$
if $\textsc{Pco}(m, c) = 1$   $//G_2 - G_4$
    return $H_1(c)$   $//G_2 - G_4$
return $H_3(m, c)$
```

```
$\textsc{Decaps}$ $(c \neq c^*)$ $//G_0 - G_2$
Parse $sk' = (sk, s)$
$m' := Dec'(sk, c)$
if $m' \neq \perp$ return $K := H(m', c)$
else return
    $K := H(s, c)$   $//G_0$
    $K := H_2(c)$   $//G_1 - G_2$
```

```
$\textsc{Decaps}$ $(c \neq c^*)$   $//G_3 - G_4$
return $K := H_1(c)$
```

Fig. 18: Games G_0-G_4 for the proof of Theorem 4

Game G_2. In game G_2, H is changes that $H_1(c)$ is returned instead of $H_3(m, c)$ when $(m,\ c)$ satisfies $\textsc{Pco}(m, c) = 1$ (i.e., $Dec'(sk, c) = m$). Note that it is impossible that $\textsc{Pco}(m_1, c) = \textsc{Pco}(m_2, c) = 1$ for $m_1 \neq m_2$ because Dec' is a deterministic algorithm. Further, as H_1 is a random function independent of H_3, H in game G_2 is also a uniform random function like the one in game G_1. Thus,

$$\Pr[G_1^{\mathcal{B}} \Rightarrow 1] = \Pr[G_2^{\mathcal{B}} \Rightarrow 1].$$

Game G_3. In game G_3, the Decaps oracle is changed that it makes no use of the secret key sk' any more. When $\mathcal{B}$ queries the Decaps oracle on c ($c \neq c^*$), $K := H_1(c)$ is returned as the response. In order to show that the output distributions of Decaps are identical in G_2 and G_3, we consider the following cases for a fixed ciphertext c and $m' := Dec'(sk, c)$.

Case 1: $m' \neq \perp$. Note that $H(m', c) = H_1(c)$ on account of $\textsc{Pco}(m', c) = 1$. Therefore, the two Decaps oracles in games G_2 and G_3 return the same value.

Case 2: $m' = \perp$. Random values $H_2(c)$ and $H_1(c)$ in $\mathcal{K}$ are returned in G_2 and G_3, respectively. In G_2, H_2 is a random function independent of G and H. In G_3, $\mathcal{B}$'s queries to H can only help him get access to H_1 at c such that $Dec'(sk, c) = \hat{m}$ for some $\hat{m} \neq \perp$. Therefore, $\mathcal{B}$ never sees $H_1(c)$ by querying H. Hence, in $\mathcal{B}$'s view, $H_1(c)$ is totally uniform at random like $H_2(c)$. As a result, the Decaps oracle in G_3 has the same output distribution as the one in G_2.

```
$A^{H,H'_1}(pk, (m^*, c^*), k_0^*)$
    $k_1^* \xleftarrow{\$} \mathcal{K}$
    $b \xleftarrow{\$} \{0,1\}$
    $b' \leftarrow \mathcal{B}^{H,\text{Decaps}}(pk, c^*, k_b^*)$
    return $b' =?b$

Decaps $(c \neq c^*)$
    return $K := H'_1(c)$
```

Fig. 19: A^{H,H'_1} for the proof of Theorem 4.

```
GAMES $G_5$
    $i \xleftarrow{\$} \{1, \ldots, q_G + q_H\}, (pk, sk') \leftarrow Gen'$
    $H_1 \xleftarrow{\$} \Omega_{H'}, H_3 \xleftarrow{\$} \Omega_H$
    $m^* \xleftarrow{\$} \mathcal{M}$
    $c^* \leftarrow Enc(pk, m^*)$
    $k^* \xleftarrow{\$} \mathcal{K}$
    run $\mathcal{B}^{G,H,\text{Decaps}}(pk, c^*, k^*)$ until the $i-$th query to $H$
    measure the argument $\hat{m} \| \hat{c}$
    return $\hat{m} =?m^* \wedge \hat{c} =?c^*$

$H(m, c)$
    if $\text{Pco}(m, c) = 1$
        return $H_1(c)$
    else return $H_3(m, c)$

Decaps $(c \neq c^*)$
    return $K := H_1(c)$
```

Fig. 20: Game G_5 for the proof of Theorem 4

We have shown that $\mathcal{B}$'s views are identical in both games and

$$\Pr[G_2^{\mathcal{B}} \Rightarrow 1] = \Pr[G_3^{\mathcal{B}} \Rightarrow 1].$$

Game G_4. In game G_4, k_0^* is chosen uniformly at random from $\mathcal{K}$. In this game, bit b is independent from $\mathcal{B}$'s view. Hence,

$$\Pr[G_4^{\mathcal{B}} \Rightarrow 1] = \frac{1}{2}.$$

Let A^{H,H'_1} be an oracle algorithm on input $(pk, (m^*, c^*), k_0^*)$ as in Fig. 19. Let $(pk, sk') \leftarrow Gen'$, $H_1 \xleftarrow{\$} \Omega_{H'}$, $H_3 \xleftarrow{\$} \Omega_H$, $m^* \xleftarrow{\$} \mathcal{M}$, $c^* \leftarrow Enc(pk, m^*)$ and H is simulated as the one in G_3 and G_4. Let H'_1 be the function with $H'_1(c^*) =\perp$ and $H'_1 = H_1$ everywhere else. Then, if $k_0^* := H(m^*, c^*)$, A^{H,H'_1} perfectly simulates

G_3. And, if $k_0^* \xleftarrow{\$} \mathcal{K}$, $A^{H,H_1'}$ perfectly simulates G_4. Let $B^{H,H_1'}$ be an oracle algorithm that on input $(pk, (m^*, c^*))$ does the following: pick $i \xleftarrow{\$} \{1, \ldots, q_H\}$ and $k_0^* \xleftarrow{\$} \mathcal{K}$, run $A^{H,H_1'}(pk, (m^*, c^*), k_0^*)$ until the i-th query to H, measure the argument of the query in the computational basis, output the measurement outcome (when $A^{H,H_1'}$ makes less than i queries, output $\perp$). Define game G_5 as in Fig. 20. Then, $\Pr[B^{H,H_1'} \Rightarrow (m^*, c^*)] = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$.

Applying Lemma 3 with $\mathcal{O}_1 = H$, $\mathcal{O}_2 = H_1'$, $inp = pk$, $x = (m^*, c^*)$ and $y = k_0^*$, we have

$$\left|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]\right| \leq 2q_H \sqrt{\Pr[G_5^{\mathcal{B}} \Rightarrow 1]}.$$

Then, we construct an adversary $\mathcal{A}$ against the OW-qPCA security of the PKE′ scheme such that $\mathtt{Adv}_{\mathrm{PKE}'}^{\mathrm{OW-qPCA}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. The adversary $\mathcal{A}$ on input $(1^\lambda, pk, c)$ does the following:

1. Run the adversary $\mathcal{B}$ in game G_5.
2. Use two different $2q_H$-wise independent functions to simulate the random oracles H_1 and H_3 respectively. The random oracle H is simulated in the same way as the one in game G_5.
3. Answer the decapsulation queries by using the DECAPS oracle in Fig. 20.
4. Select $k^* \xleftarrow{\$} \mathcal{K}$ and respond to $\mathcal{B}$'s challenge query with (c, k^*).
5. Select $i \xleftarrow{\$} \{1, \ldots, q_H\}$, measure the argument $\hat{m}\|\hat{c}$ of the i-th query to H and output $\hat{m}$.

According to Lemma 1, $\mathtt{Adv}_{\mathrm{PKE}'}^{\mathrm{OW-}q\mathrm{PCA}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. Finally, combing this with the bounds derived above, we can conclude that

$$\mathtt{Adv}_{\text{KEM-III}}^{\text{IND-CCA}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{\mathcal{M}}} + 2q_H \cdot \sqrt{\mathtt{Adv}_{\mathrm{PKE}'}^{\mathrm{OW-}q\mathrm{PCA}}(\mathcal{A})}.$$

□