



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 Wang¹,
and Zhi Ma^{1,4}(✉)

¹ State Key Laboratory of Mathematical Engineering and Advanced Computing,
Zhengzhou, Henan, China

hdjiang13@gmail.com, wfallmoon@163.com, ma_zhi@163.com

² TCA Laboratory, State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing, China
{zfmzhang, chenlong}@tca.iscas.ac.cn

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

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

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 $U^{\not\perp}$, U^{\perp} , $U_m^{\not\perp}$, U_m^{\perp} , $QU_m^{\not\perp}$ and QU_m^{\perp} (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) rejection¹ and Q means adding an additional hash to the ciphertext. Combing these modular transformations, they obtained several variants of FO transformation $FO^{\not\perp}$, FO^{\perp} , $FO_m^{\not\perp}$, FO_m^{\perp} , $QFO_m^{\not\perp}$ and QFO_m^{\perp} (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 $QU_m^{\not\perp}$, QU_m^{\perp} , $QFO_m^{\not\perp}$ and QFO_m^{\perp} . For these transformations, there is an additional hash in the ciphertext, which plays an important role in their reductions. The security reductions for $U^{\not\perp}$, U^{\perp} , $U_m^{\not\perp}$, U_m^{\perp} , $FO^{\not\perp}$, FO^{\perp} , $FO_m^{\not\perp}$ and FO_m^{\perp} are just presented in the ROM.

¹ 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., $\text{QFO}^{\not\perp}$, QFO^\perp , $\text{QFO}_m^{\not\perp}$ and 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 recent works [12,13].

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 $\text{QU}_m^{\not\perp}$, QU_m^\perp , $\text{QFO}_m^{\not\perp}$ and 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], and QC-MDPC [29], there

² QFO^\perp ($\text{QFO}^{\not\perp}$) is the same as QFO_m^\perp ($\text{QFO}_m^{\not\perp}$) except that $K = H(m, c)$. Its security proof can be easily obtained from the one for QFO_m^\perp ($\text{QFO}_m^{\not\perp}$) in [7].

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 [30]. Particularly, the chosen-ciphertext attacks by exploiting the gathered correctness errors [30,31] 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. [32] showed that the HILA5 KEM [33] 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 work [7], an IND-CCA-secure KEM in the ROM can be easily constructed by applying the transformation U_m^\perp (or U_m^{\leftarrow}) 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 [34], NTRU [27], the McEliece PKE [35], and the Niederreiter PKE [36]. However, the popular LWE cryptosystem and variants [37–40] 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 U_m^{\leftarrow} with stronger assumptions for under-

lying 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, FO^{\swarrow} and FO_m^{\swarrow} 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 FO_m^{\swarrow} , $\text{FO}_m^{\swarrow} = \text{TPunc} \circ \text{U}_m^{\swarrow 3}$, 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 QFO^{\swarrow} , QFO^{\perp} , QFO_m^{\swarrow} and 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 FO^{\swarrow} 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 U^{\swarrow} , U^{\perp} , U_m^{\swarrow} and 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, U^{\swarrow} from

³ 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	FO ⊥	Y	N	Y
EMBLEM and R.EMBLEM	QFO [⊥]	Y	N	Y
FrodoKEM	QFO ⊥	Y	N	Y
KINDI	QFO ⊥ _m	Y	N	Y
LAC	FO ⊥	Y	N	N
Lepton	QFO [⊥]	Y	N	Y
LIMA	FO [⊥] _m	N ^a	N	Y
Lizard	QFO ⊥	Y	N	Y
NewHope	QFO ⊥	Y	N	Y
NTRU-HRSS-KEM	QFO [⊥] _m	N	N	Y
Odd Manhattan	U [⊥] _m	N	N	N
OKCN-AKCN-CNKE	QFO ⊥	Y	N	Y
Round2	QFO ⊥	Y	N	Y
SABER	FO ⊥	Y	N	Y
ThreeBears	FO [⊥] _m	Y	N	Y
Titanium	QFO ⊥	Y	N	Y
BIG QUAKE	QFO [⊥]	N	N	Y
Classic McEliece	U ⊥	N	Y	Y
DAGS	QFO [⊥] _m	N	N	Y
HQC	QFO [⊥]	Y	N	Y
LEDAkem	U ⊥ _m	Y	Y	N
LOCKER	QFO [⊥]	Y	N	Y
QC-MDPC	QFO [⊥] _m	Y	N	Y
RQC	QFO [⊥]	N	N	Y
SIKE	FO ⊥	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 [41].

Table 2. FO transformations from standard security assumptions.

Transformation	Underlying security	Security bound	Additional hash	Perfectly correct?
$QFO_m^{\not\perp}$ and QFO_m^{\perp} [7]	OW-CPA	$q\sqrt{q^2\delta + q\sqrt{\epsilon}}$	Y	N
$FO_m^{\not\perp}$ [12]	IND-CPA	$q\sqrt{\epsilon}$	N	Y
$FO_m^{\not\perp}$ and FO_m^{\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?
QU_m^{\perp} [7]	OW-PCA	$q\sqrt{\epsilon}$	Y	N	N
$QU_m^{\not\perp}$ [7]	OW-PCA	$q\sqrt{\epsilon}$	Y	N	N
$U_m^{\not\perp}$ [12]	DS	ϵ	N	Y	Y
$U_m^{\not\perp}$ Our work	OW-qPCA	$q\sqrt{\epsilon}$	N	N	N
U^{\perp} Our work	OW-qPVCA	$q\sqrt{\epsilon}$	N	N	N
$U_m^{\not\perp}$ Our work	OW-CPA	$q\sqrt{\delta} + q\sqrt{\epsilon}$	N	Y	N
$U_m^{\not\perp}$ Our work	DS	$q\sqrt{\delta} + \epsilon$	N	Y	N
U_m^{\perp} Our work	OW-VA	$q\sqrt{\delta} + q\sqrt{\epsilon}$	N	Y	N

OW-qPCA, U^{\perp} from OW-qPVCA, $U_m^{\not\perp}$ from OW-CPA (and DS), U_m^{\perp} from OW-VA, to IND-CCA without additional hash.

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 [35], and Niederreiter [36]. 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 oracle⁴.

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 $H'(\cdot) = \text{DECAPS}(sk, \cdot)$. 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 FO^{\leftarrow} and FO_m^{\leftarrow} , reprogramming the random oracles G and H is a natural approach. In quantum setting, the

⁴ This method is also used by a concurrent and independent work [12].

one-way to hiding (OW2H) lemma [42, 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.

To tighten the security bounds, we have to decrease the times of the usage of the OW2H lemma. [7] analyzed the QROM security of QFO_m^\leftarrow (and 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 FO^\leftarrow and FO_m^\leftarrow , 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 FO^\leftarrow and FO_m^\leftarrow 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 FO_m^\leftarrow is instantiated with a Ring-LWE-based PKE scheme [39], the security of the underlying Ring-LWE problem can be reduced to the IND-CCA security of KEM [43].

In [12], Saito et al. presented a tight security reduction for U_m^{\leftarrow} by assuming a stronger underlying DPKE, which is only satisfied by Classic McEliece in Table 1. For the widely used FO^{\leftarrow} and FO_m^{\leftarrow} , 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 [7, 43], 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 [43]. 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 [44, 45]. 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 FO^{\leftarrow} and FO_m^{\leftarrow} 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 [7–9, 43]. 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 FO^{\leftarrow} and FO_m^{\leftarrow} 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 $x \stackrel{\$}{\leftarrow} X$, 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 the complete proofs of last two are presented in the full version [13]. We refer the reader to [46] for basic of quantum computation. Here, we just recall two 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 [47, 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 [48, 49]). *Let $\gamma \in [0, 1]$. Let Z be a finite set. $N_1 : Z \rightarrow \{0, 1\}$ 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*

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

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. [48, Lemma 37] and [49, Theorem 1] just consider the specific case where all p_z s are equal to γ . But in our security proof, we need to consider the case where $p_z \leq \gamma$ and p_z s are in general different from each other. Fortunately, it is not difficult to verify that the proof of [48, Lemma 37] can be extended to this generic case.

The one-way to hiding (OW2H) lemma [42, 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 $i \xleftarrow{\$} \{1, \dots, q_1\}$ 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

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

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 [42, Lemma 6.2] and we present it in the full version [13].

Lemma 4. *Let Ω_H ($\Omega_{H'}$) be the set of all functions $H : \{0, 1\}^{n_1} \times \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$ ($H' : \{0, 1\}^{n_2} \rightarrow \{0, 1\}^m$). 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_1} 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,

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

We now sketch the proof of Lemma 4. For the complete proof, please refer to the full version [13].

Proof sketch. In classical setting, it is obvious that $|\Pr[1 \leftarrow A^{H, F_0}] - \Pr[1 \leftarrow A^{H, F_1}]|$ 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, $|\Pr[1 \leftarrow A^{H, F_0}] - \Pr[1 \leftarrow A^{H, F_1}]| \leq q \frac{1}{2^{n_1}}$. 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 [42, 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 [42, Lemma 6.2], we can bound $|\Pr[1 \leftarrow A^{H, F_0}] - \Pr[1 \leftarrow A^{H, F_1}]|$ 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, $|\Pr[1 \leftarrow A^{H, F_0}] - \Pr[1 \leftarrow A^{H, F_1}]| \leq 2q \frac{1}{\sqrt{2^{n_1}}}$.

2.2 Cryptographic Primitives

Definition 1 (Public-key encryption). A public-key encryption scheme $\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ 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 $c \leftarrow \text{Enc}(pk, m)$. If necessary, we make the used randomness of encryption explicit by writing $c := \text{Enc}(pk, m; r)$, where $r \xleftarrow{\$} \mathcal{R}$ (\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 $m := \text{Dec}(sk, c)$ 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

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

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

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

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

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

Game OW-ATK	PCO(m, c)	VAL(c)
1 : $(pk, sk) \leftarrow \text{Gen}$	1 : if $m \notin \mathcal{M}$	1 : $m := \text{Dec}(sk, c)$
2 : $m^* \xleftarrow{\$} \mathcal{M}$	2 : return \perp	2 : if $m \in \mathcal{M}$
3 : $c^* \leftarrow \text{Enc}(pk, m^*)$	3 : else return	3 : return 1
4 : $m' \leftarrow \mathcal{A}^{O_{\text{ATK}}}(pk, c^*)$	4 : $\text{Dec}(sk, c) = ?m$	4 : else return 0
5 : return $m' = ?m^*$		

Fig. 1. Games OW-ATK ($\text{ATK} \in \{\text{CPA}, \text{VA}, \text{qPCA}, \text{qPVCA}\}$) 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 Sect. 4.

Definition 4 (DS-secure DPKE [12]). Let $D_{\mathcal{M}}$ denote an efficiently samplable distribution on \mathcal{M} . A DPKE scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ 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: $\text{DISJ}_{\text{PKE}, S} := \max_{pk} \Pr[c \in \text{Enc}(pk, \mathcal{M}) : c \leftarrow S(pk)]$ is negligible. (2) Ciphertext-indistinguishability: For any PPT adversary \mathcal{A} , $\text{Adv}_{\text{PKE}, D_{\mathcal{M}}, S}^{\text{DS-IND}}(\mathcal{A}) := |\Pr[\mathcal{A}(pk, c^*) \rightarrow 1 : (pk, sk) \leftarrow \text{Gen}; m^* \leftarrow D_{\mathcal{M}}; c^* := \text{Enc}(pk, m^*)] - \Pr[\mathcal{A}(pk, c^*) \rightarrow 1 : (pk, sk) \leftarrow \text{Gen}; c^* \leftarrow S(pk)]|$ 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 $K := \text{Decaps}(sk, c) \in \mathcal{K}$ or a special symbol $\perp \notin \mathcal{K}$ to indicate that c is not a valid encapsulation.

Game IND-CCA	DECAPS(sk, c)
1: $(pk, sk) \leftarrow \text{Gen}$	1: if $c = c^*$
2: $b \xleftarrow{\$} \{0, 1\}$	2: return \perp
3: $(K_0^*, c^*) \leftarrow \text{Encaps}(pk)$	3: else return
4: $K_1^* \xleftarrow{\$} \mathcal{K}$	4: $K := \text{Decaps}(sk, c)$
5: $b' \leftarrow \mathcal{A}^{\text{DECAPS}}(pk, c^*, K_b^*)$	
6: return $b' = ?b$	

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

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^{\leftarrow} and FO_m^{\leftarrow} , see Figs. 3 and 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^{\leftarrow} and FO_m^{\leftarrow} by adding an additional length-preserving hash function to the ciphertext. Here, we present two QROM security proofs for FO^{\leftarrow} and FO_m^{\leftarrow} respectively without suffering any ciphertext overhead.

Gen'	$Encaps(pk)$	$Decaps(sk', c)$
1: $(pk, sk) \leftarrow Gen$	1: $m \xleftarrow{\$} \mathcal{M}$	1: Parse $sk' = (sk, s)$
2: $s \xleftarrow{\$} \mathcal{M}$	2: $c = Enc(pk, m; G(m))$	2: $m' := Dec(sk, c)$
3: $sk' := (sk, s)$	3: $K := H(m, c)$	3: if $Enc(pk, m'; G(m')) = c$
4: return (pk, sk')	4: return (K, c)	4: return $K := H(m', c)$
		5: else return
		6: $K := H(s, c)$

Fig. 3. IND-CCA-secure $KEM-I=FO_m^{\not\leftarrow}[\text{PKE}, G, H]$

Gen'	$Encaps(pk)$	$Decaps(sk', c)$
1: $(pk, sk) \leftarrow Gen$	1: $m \xleftarrow{\$} \mathcal{M}$	1: Parse $sk' = (sk, k)$
2: $k \xleftarrow{\$} \mathcal{K}^{prf}$	2: $c = Enc(pk, m; G(m))$	2: $m' := Dec(sk, c)$
3: $sk' := (sk, k)$	3: $K := H(m)$	3: if $Enc(pk, m'; G(m')) = c$
4: return (pk, sk')	4: return (K, c)	4: return $K := H(m')$
		5: else return
		6: $K := f(k, c)$

Fig. 4. IND-CCA-secure $KEM-II=FO_m^{\not\leftarrow}[\text{PKE}, G, H, f]$

To a public-key encryption scheme $\text{PKE} = (Gen, Enc, Dec)$ with message space \mathcal{M} and randomness space \mathcal{R} , hash functions $G : \mathcal{M} \rightarrow \mathcal{R}$, $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$ and a pseudorandom function (PRF) f with key space \mathcal{K}^{prf} , we associate $KEM-I=FO_m^{\not\leftarrow}[\text{PKE}, G, H]$ and $KEM-II=FO_m^{\not\leftarrow}[\text{PKE}, G, H, f]$ ⁵ shown in Figs. 3 and 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 \xrightarrow{QROM} 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 $\text{Adv}_{KEM-I}^{\text{IND-CCA}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 4q_G \sqrt{\delta} + 2(q_G + q_H) \cdot \sqrt{\text{Adv}_{PKE}^{\text{OW-CPA}}(\mathcal{A})}$ and the running time of \mathcal{A} is about that of \mathcal{B} .*

⁵ $FO_m^{\not\leftarrow}$ here is the generic version of $FO_m^{\not\leftarrow}$ 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').

Proof. 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 . Denote Ω_G , Ω_H and $\Omega_{H'}$ as the sets of all functions $G : \mathcal{M} \rightarrow \mathcal{R}$, $H : \mathcal{M} \times \mathcal{C} \rightarrow \mathcal{K}$ and $H' : \mathcal{C} \rightarrow \mathcal{K}$, respectively. Consider the games in Figs. 5 and 9.

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

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

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

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

GAME G_2 . Note that in game G_1 , $H(m, c) = H_3(m, c)$. In game G_2 , 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

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

GAMES $G_0 - G_4$	$H(m, c)$
1 : $(pk, sk') \leftarrow \text{Gen}' ; G \xleftarrow{\$} \Omega_G$	1 : if $\text{Enc}(pk, m; G(m)) = c$ // $G_2 - G_4$
2 : $H_1, H_2 \xleftarrow{\$} \Omega_{H'} ; H_3 \xleftarrow{\$} \Omega_H$	2 : return $H_1(c)$ // $G_2 - G_4$
3 : $m^* \xleftarrow{\$} \mathcal{M}$	3 : return $H_3(m, c)$
4 : $r^* := G(m^*)$ // $G_0 - G_3$	<hr style="border: none; border-top: 1px solid black; margin: 0;"/> DECAPS ($c \neq c^*$) // $G_0 - G_2$
5 : $r^* \xleftarrow{\$} \mathcal{R}$ // G_4	1 : Parse $sk' = (sk, s)$
6 : $c^* := \text{Enc}(pk, m^*; r^*)$	2 : $m' := \text{Dec}(sk, c)$
7 : $k_0^* := H(m^*, c^*)$	3 : if $\text{Enc}(pk, m'; G(m')) = c$
8 : $k_0^* \xleftarrow{\$} \mathcal{K}$ // G_4	4 : return $K := H(m', c)$
9 : $k_1^* \xleftarrow{\$} \mathcal{K}$	5 : else return
10 : $b \xleftarrow{\$} \{0, 1\}$	6 : $K := H(s, c)$ // G_0
11 : $b' \leftarrow \mathcal{B}^{G, H, \text{DECAPS}}(pk, c^*, k_b^*)$	7 : $K := H_2(c)$ // $G_1 - G_2$
12 : return $b' = ?b$	<hr style="border: none; border-top: 1px solid black; margin: 0;"/> DECAPS ($c \neq c^*$) // $G_3 - G_4$
	1 : return $K := H_1(c)$

Fig. 5. Games G_0 - G_4 for the proof of Theorem 1

A^{H, F_i}	$\text{DECAPS } (c \neq c^*)$
1: $(pk, sk) \leftarrow \text{Gen}; G \xleftarrow{\$} \Omega_G$	1: $m' := \text{Dec}(sk, c)$
2: $m^* \xleftarrow{\$} \mathcal{M}$	2: if $\text{Enc}(pk, m'; G(m')) = c$
3: $r^* := G(m^*)$	3: return $K := H(m', c)$
4: $c^* := \text{Enc}(pk, m^*; r^*)$	4: else return
5: $k_0^* := H(m^*, c^*); k_1^* \xleftarrow{\$} \mathcal{K}$	5: $K := F_i(c)$
6: $b \xleftarrow{\$} \{0, 1\}$	
7: $b' \leftarrow \mathcal{B}^{G, H, \text{DECAPS}}(pk, c^*, k_b^*)$	
8: return $b' = ?b$	

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

$A^N(pk, sk)$	$\tilde{G}(m)$
1: Pick $2q_G$ -wise function f	1: if $N(m) = 0$
2: $b'' \leftarrow B^{\tilde{G}}(pk, sk)$	2: $\tilde{G}(m) = \text{Sample}(\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
3: return b''	3: else
	4: $\tilde{G}(m) = \text{Sample}(\mathcal{R}_{\text{bad}}(pk, sk, m); f(m))$
	5: return $\tilde{G}(m)$

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

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

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

denote the set of “bad” 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 $(pk, sk) \leftarrow \text{Gen}$.

Let G' be a random function such that $G'(m)$ is sampled from the uniform distribution in $\mathcal{R} \setminus \mathcal{R}_{\text{bad}}(pk, sk, m)$. Let

$$g'(\cdot) = \text{Enc}(pk, \cdot; G'(\cdot)).$$

Distinctly, g' is an injective function. $H_1 \circ g'$ has the same output distribution as H in G_1 . Thus, distinguishing G_2 from G_1 is equivalent to distinguishing g from g' , which is essentially the distinguishing problem between G and G' .

Let N_1 be the 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 $\Pr[N_1(m) = 0] = 1 - \delta(pk, sk, m)$. Let N_2 be a constant function that always outputs 0 for any input. 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 .

For any efficient quantum adversary $B^{\tilde{G}}(pk, sk)$, 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 $\tilde{G} = G$ if $N = N_1$ and $\tilde{G} = G'$ 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 $|\Pr[G_1^{\mathcal{B}} \Rightarrow 1 : (pk, sk)] - \Pr[G_2^{\mathcal{B}} \Rightarrow 1 : (pk, sk)]|$ can be bounded by the maximum distinguishing probability between G and G' for $B^{\tilde{G}}(pk, sk)$. 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 . 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. 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_2 and G_3 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_2 and G_3 respectively. In G_2 , 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_3 , \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, if \mathcal{B} can not find a m'' such that $g(m'') = c$, $H_1(c)$ is also a fresh random key just like $H_2(c)$ in his view. Since $m'' \neq m'$, finding such an m'' is exactly the event E that \mathcal{B} finds a plaintext m'' such that $Dec(sk, g(m'')) \neq m''$. That is, in this case, if E does not happen, the output distributions of the DECAPS oracles in G_2 and G_3 are same in \mathcal{B} 's view.

As a result, G_2 and G_3 only differ when E happens. By [7, Lemma 4.3], we know that if \mathcal{B} can find a plaintext m'' such that $Dec(sk, g(m'')) \neq m''$ with at most

q_G quantum queries to g , we can easily construct another adversary \mathcal{B}' who can find a plaintext m'' such that $N_1(m'') = 1$ with at most q_G quantum queries to N_1 . Considering that the PKE scheme is δ -correct, we can derive the upper bound of $\Pr[E]$ by utilizing Lemma 2, $\Pr[E] \leq \Pr[N_1(m'') = 1 : (pk, sk) \leftarrow \text{Gen}, m'' \leftarrow \mathcal{B}'^{N_1}] \leq 2q_G\sqrt{\delta}$. Therefore,

$$|\Pr[G_2^{\mathcal{B}} \Rightarrow 1] - \Pr[G_3^{\mathcal{B}} \Rightarrow 1]| \leq \Pr[E] \leq 2q_G\sqrt{\delta}.$$

GAME G_4 . In game G_4 , 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,

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

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_3 and G_4 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 $|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]|$. 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 $(G \times H_1^g)(x) := (G(x), H_1^g(x))$ ⁶. 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_3 and G_4 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_3 and G_4 , i.e., $(pk, sk') \leftarrow \text{Gen}', G \xleftarrow{\$} \Omega_G, H_1 \xleftarrow{\$} \Omega_{H'}, H_1^g := H_1 \circ g$. 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_3 . 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_4 . Let $B^{G \times H_1^g, H'_1}$ be an oracle algorithm that on input (pk, m^*) does the following: pick $i \xleftarrow{\$} \{1, \dots, q_G + q_H\}$, $r^* \xleftarrow{\$} \mathcal{R}$ 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_5 as in Fig. 9. Then, $\Pr[B^{G \times H_1^g, H'_1} \Rightarrow m^*] = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$.

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

$$|\Pr[G_3^{\mathcal{B}} \Rightarrow 1] - \Pr[G_4^{\mathcal{B}} \Rightarrow 1]| \leq 2(q_G + q_H)\sqrt{\Pr[G_5^{\mathcal{B}} \Rightarrow 1]}.$$

⁶ 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 [14, 50, 51] has been used to ignore part of the output of an oracle.

$A^{G \times H_1^g, H_1'}(pk, m^*, (r^*, k_0^*))$	$H(m, c)$
1 : $H_3 \stackrel{\$}{\leftarrow} \Omega_H$	1 : if $g(m) = c$
2 : $c^* := Enc(pk, m^*; r^*)$	2 : return $H_1^g(m)$
3 : $k_1^* \stackrel{\$}{\leftarrow} \mathcal{K}$	3 : else return $H_3(m, c)$
4 : $b \stackrel{\$}{\leftarrow} \{0, 1\}$	<hr style="width: 100%;"/>
5 : $b' \leftarrow \mathcal{B}^{G, H, DECAPS}(pk, c^*, k_b^*)$	DECAPS ($c \neq c^*$)
6 : return $b' =? b$	1 : return $K := H_1'(c)$

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

GAMES G_5	
1 : $i \stackrel{\$}{\leftarrow} \{1, \dots, q_G + q_H\}, (pk, sk') \leftarrow Gen', G \stackrel{\$}{\leftarrow} \Omega_G$	
2 : $H_1 \stackrel{\$}{\leftarrow} \Omega_{H'}, H_3 \stackrel{\$}{\leftarrow} \Omega_H$	
3 : $m^* \stackrel{\$}{\leftarrow} \mathcal{M}, r^* \stackrel{\$}{\leftarrow} \mathcal{R}$	
4 : $c^* := Enc(pk, m^*; r^*)$	
5 : $k_0^*, k_1^* \stackrel{\$}{\leftarrow} \mathcal{K}$	
6 : $b \stackrel{\$}{\leftarrow} \{0, 1\}$	
7 : run $\mathcal{B}^{G, H, DECAPS}(pk, c^*, k_b^*)$ until the i -th query to $G \times H_1^g$	
8 : measure the argument \hat{m}	
9 : return $\hat{m} =? m^*$	
<hr style="width: 100%;"/>	<hr style="width: 100%;"/>
$H(m, c)$	DECAPS ($c \neq c^*$)
1 : if $Enc(pk, m; G(m)) = c$	1 : return $K := H_1(c)$
2 : return $H_1(c)$	
3 : else return $H_3(m, c)$	

Fig. 9. Game G_5 for the proof of Theorem 1

Next, we construct an adversary \mathcal{A} against the OW-CPA security of the PKE scheme such that $\text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(\mathcal{A}) = \Pr[G_5^{\mathcal{B}} \Rightarrow 1]$. 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 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_5 .
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 $i \xleftarrow{\$} \{1, \dots, q_G + q_H\}$, measure the argument \hat{m} of i -th query to $G \times H_1^q$ and output \hat{m} .

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

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

□

Theorem 2 (PKE OW-CPA $\xrightarrow{\text{QROM}}$ 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 $\text{Adv}_{\text{KEM-II}}^{\text{IND-CCA}}(\mathcal{B}) \leq \text{Adv}_{\text{PRF}}(\mathcal{A}') + 4q_G \sqrt{\delta} + 2(q_H + q_G) \cdot \sqrt{\text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(\mathcal{A})}$ 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 the full version [13].

4 Modular Analysis of FO Transformation in the QROM

In [7], Hofheinz et al. introduced seven modular transformations T , $U^{\not\perp}$, U^\perp , $U_m^{\not\perp}$, U_m^\perp , $QU_m^{\not\perp}$ and QU_m^\perp . But, they just presented QROM security reductions for the transformations T , $QU_m^{\not\perp}$ and QU_m^\perp . Different from the transformations $U^{\not\perp}$, U^\perp , $U_m^{\not\perp}$ and U_m^\perp , the transformations $QU_m^{\not\perp}$ and QU_m^\perp have an additional length-preserving hash in the ciphertext, thus they can follow the proof technique in [14, 52] 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 $U^{\not\perp}$, U^\perp , $U_m^{\not\perp}$ and U_m^\perp are missing in [7]. In [12], Saito et al. presented a tight QROM security reduction for $U_m^{\not\perp}$ with stronger assumptions for underlying DPKE scheme, DS-security and perfect correctness.

In this section, we revisit the transformations $U^{\not\perp}$, U^\perp , $U_m^{\not\perp}$ and 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 Sect. 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 $U^{\not\perp}$ (U^\perp) from the OW-qPCA (OW-qPVCA) security of PKE to the IND-CCA security of KEM is given in Sect. 4.2 (4.3). In Sect. 4.4, we show that $U_m^{\not\perp}$ (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 $\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ with message space \mathcal{M} and randomness space R , and a hash function $G : \mathcal{M} \rightarrow \mathcal{R}$, we associate $\text{PKE}' = T[\text{PKE}, G]$. The algorithms of $\text{PKE}' = (\text{Gen}, \text{Enc}', \text{Dec}')$ are defined in Fig. 10.

Theorem 3 (PKE OW-CPA $\stackrel{\text{QROM}}{\Rightarrow}$ 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 $\text{Adv}_{\text{PKE}'}^{\text{OW-qPCA}}(\mathcal{B}) \leq 2q_G \cdot \sqrt{\delta} + (1 + 2q_G) \cdot \sqrt{\text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(\mathcal{A})}$ 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 that $\text{Enc}(pk, m; G(m)) = ?c$ 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 exists 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 $\text{Dec}(sk, \text{Enc}(pk, m; G(m))) = m$, and otherwise $N_1(m) = 1$. Thus, using Lemma 2, we can obtain that $|\Pr[G_0^{\mathcal{B}} \Rightarrow 1] - \Pr[G_1^{\mathcal{B}} \Rightarrow 1]| \leq 2q_G \cdot \sqrt{\delta}$. Then, following the security proof of [7, Theorem 4.4], we can easily prove Theorem 3.

<u>$\text{Enc}'(pk, m)$</u>	<u>$\text{Dec}'(sk, c)$</u>
1 : $c = \text{Enc}(pk, m; G(m))$	1 : $m' := \text{Dec}(sk, c)$
2 : return c	2 : if $\text{Enc}(pk, m'; G(m')) = c$
	3 : return m'
	4 : else return \perp

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

4.2 $\mathcal{U}^{\mathcal{A}}$: from OW-qPCA to IND-CCA in the QROM

To a public-key encryption $\text{PKE}' = (\text{Gen}', \text{Enc}', \text{Dec}')$ and a hash function H , we associate $\text{KEM-III} = \mathcal{U}^{\mathcal{A}}[\text{PKE}', H]$. The algorithms of $\text{KEM-III} = (\text{Gen}, \text{Encaps}, \text{Decaps})$ are defined in Fig. 11.

<i>Gen</i>	<i>Encaps(pk)</i>	<i>Decaps(sk', c)</i>
1: $(pk, sk) \leftarrow Gen'$	1: $m \xleftarrow{\$} \mathcal{M}$	1: Parse $sk' = (sk, s)$
2: $s \xleftarrow{\$} \mathcal{M}$	2: $c \leftarrow Enc'(pk, m)$	2: $m' := Dec'(sk, c)$
3: $sk' := (sk, s)$	3: $K := H(m, c)$	3: if $m' = \perp$
4: return (pk, sk')	4: return (K, c)	4: return $K := H(s, c)$
		5: else return
		6: $K := H(m', c)$

Fig. 11. IND-CCA-secure KEM-III = $U^{\mathcal{A}}[\text{PKE}', H]$

Theorem 4 (PKE' OW-qPCA $\stackrel{QROM}{\Rightarrow}$ 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 $\text{Adv}_{\text{KEM-III}}^{\text{IND-CCA}}(\mathcal{B}) \leq 2q_H \frac{1}{\sqrt{|\mathcal{M}|}} + 2q_H \cdot \sqrt{\text{Adv}_{\text{PKE}'}^{\text{OW-qPCA}}(\mathcal{A})}$ 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 the full version [13].

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 plaintext-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 plaintext-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 Theorems 1 and 2, the δ term is removed with the OW-qPCA security of underlying PKE.

Gen	$Encaps(pk)$	$Decaps^\perp(sk, c)$
1 : $(pk, sk) \leftarrow Gen'$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : $m' := Dec'(sk, c)$
2 : return (pk, sk)	2 : $c \leftarrow Enc'(pk, m)$	2 : if $m' = \perp$
	3 : $K := H(m, c)$	3 : return \perp
	4 : return (K, c)	4 : else return
		5 : $K := H(m', c)$

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 $\text{PKE}' = (Gen', Enc', Dec')$ and a hash function H , we associate $\text{KEM-IV} = U^\perp[\text{PKE}', H]$. We remark that U^\perp is essentially the transformation [6, Table 2], a KEM variant of the REACT/GEM transformations [53, 54]. The algorithms of $\text{KEM-IV} = (Gen, Encaps, Decaps^\perp)$ are defined in Fig. 12.

Theorem 5 (PKE' OW-qPVCA $\stackrel{QROM}{\Rightarrow}$ 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 $\text{Adv}_{\text{KEM-IV}}^{\text{IND-CCA}}(\mathcal{B}) \leq 2q_H \cdot \sqrt{\text{Adv}_{\text{PKE}'}^{\text{OW-qPVCA}}(\mathcal{A})}$ 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_m^\neq/U_m^\perp : from OW-CPA/OW-VA to IND-CCA for Deterministic Encryption in the QROM

The transformation $U_m^\neq(U_m^\perp)$ is a variant of $U^\neq(U^\perp)$ that derives the KEM key as $K = H(m)$, instead of $K = H(m, c)$. To a deterministic public-key

encryption scheme $\text{PKE}' = (\text{Gen}', \text{Enc}', \text{Dec}')$ with message space \mathcal{M} , a hash function $H : \mathcal{M} \rightarrow \mathcal{K}$, and a pseudorandom function f with key space \mathcal{K}^{prf} , we associate $\text{KEM-V} = \text{U}_m^{\neq}[\text{PKE}', H, f]$ and $\text{KEM-VI} = \text{U}_m^{\perp}[\text{PKE}', H]$ shown in Figs. 13 and 14, respectively.

<i>Gen</i>	<i>Encaps(pk)</i>	<i>Decaps(sk', c)</i>
1 : $(pk, sk) \leftarrow \text{Gen}'$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : Parse $sk' = (sk, k)$
2 : $k \xleftarrow{\$} \mathcal{K}^{\text{prf}}$	2 : $c := \text{Enc}'(pk, m)$	2 : $m' := \text{Dec}'(sk, c)$
3 : $sk' := (sk, k)$	3 : $K := H(m)$	3 : if $\text{Enc}'(pk, m') = c$
4 : return (pk, sk')	4 : return (K, c)	4 : return $K := H(m')$
		5 : else return
		6 : $K := f(k, c)$

Fig. 13. IND-CCA-secure $\text{KEM-V} = \text{U}_m^{\neq}[\text{PKE}', H, f]$

<i>Gen</i>	<i>Encaps(pk)</i>	<i>Decaps(sk, c)</i>
1 : $(pk, sk) \leftarrow \text{Gen}'$	1 : $m \xleftarrow{\$} \mathcal{M}$	1 : $m' := \text{Dec}(sk, c)$
2 : return (pk, sk)	2 : $c := \text{Enc}'(pk, m)$	2 : if $\text{Enc}'(pk, m') = c$
	3 : $K := H(m)$	3 : return $K := H(m')$
	4 : return (K, c)	4 : else return \perp

Fig. 14. IND-CCA-secure $\text{KEM-VI} = \text{U}_m^{\perp}[\text{PKE}', H]$

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, combining 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 $\stackrel{QROM}{\Rightarrow}$ 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 oracle⁷, 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'*

⁷ For the deterministic scheme PKE', given public key pk , quantum adversary \mathcal{B} can execute the encryption algorithm Enc' in a quantum computer.

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

Theorem 7 (PKE' OW-VA $\stackrel{QROM}{\Rightarrow}$ 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 $\text{Adv}_{\text{KEM-VI}}^{\text{IND-CCA}}(\mathcal{B}) \leq 2q_E\sqrt{\delta} + 2q_H \cdot \sqrt{\text{Adv}_{\text{PKE}'}^{\text{OW-VA}}(\mathcal{A})}$ 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 J. Comput.* **33**(1), 167–226 (2003)
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.) *ACISP 2008*. LNCS, vol. 5107, pp. 69–83. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70500-0_6
3. Fujioka, A., Suzuki, K., Xagawa, K., Yoneyama, K.: Strongly secure authenticated key exchange from factoring, codes, and lattices. *Des. Codes Crypt.* **76**(3), 469–504 (2015)
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.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) *CRYPTO 1991*. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_35
6. Dent, A.W.: A designer's guide to KEMs. In: Paterson, K.G. (ed.) *Cryptography and Coding 2003*. LNCS, vol. 2898, pp. 133–151. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40974-8_12
7. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) *TCC 2017*. LNCS, vol. 10677, pp. 341–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70500-2_12

8. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_34
9. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptol.* **26**(1), 1–22 (2013)
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, pp. 62–73. ACM (1993)
11. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_3
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.) EUROCRYPT 2018. LNCS, vol. 10822, pp. 520–551. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78372-7_17
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.) TCC 2016-B. LNCS, vol. 9986, pp. 192–216. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_8
15. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995). <https://doi.org/10.1007/BFb0053428>
16. Fujisaki, E., Okamoto, T., Pointcheval, D., Stern, J.: RSA-OAEP is secure under the RSA assumption. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 260–274. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_16
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, pp. 212–219. ACM (1996)
18. Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P.: High-speed key encapsulation from NTRU. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 232–252. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66787-4_12
19. Hamburg, M.: Module-LWE: the three bears. Technical report. <https://www.shiftright.org/papers/threebears/>
20. Ding, J.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptology ePrint Archive 2012/688 (2012)
21. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
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, pp. 553–570 (2015)
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, pp. 327–343. USENIX Association (2016)

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, pp. 1006–1018. ACM (2016)
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 (2018, to appear)
27. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS-III 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054868>
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.) SAC 2017. LNCS, vol. 10719, pp. 235–260. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72565-9_12
29. 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), pp. 2069–2073. IEEE (2013)
30. Howgrave-Graham, N., et al.: The impact of decryption failures on the security of NTRU encryption. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 226–246. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_14
31. 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.) ASIACRYPT 2016. LNCS, vol. 10031, pp. 789–815. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_29
32. Bernstein, D.J., Groot Bruinderink, L., 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.) AFRICACRYPT 2018. LNCS, vol. 10831, pp. 203–216. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89339-6_12
33. Saarinen, M.-J.O.: HILA5: on reliability, reconciliation, and error correction for ring-LWE encryption. In: Adams, C., Camenisch, J. (eds.) SAC 2017. LNCS, vol. 10719, pp. 192–212. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-72565-9_10
34. 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, pp. 197–206. ACM (2008)
35. McEliece, R.J.: A public-key cryptosystem based on algebraic. DSN progress report 42-44, pp. 114–116 (1978)
36. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. *Probl. Control Inf. Theory* **15**(2), 159–166 (1986)
37. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. *J. ACM (JACM)* **56**(6), 34 (2009)
38. Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19074-2_21

39. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
40. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 35–54. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_3
41. Google: PQC-forum. LIMA (2018). <https://groups.google.com/a/list.nist.gov/forum/#!topic/pqc-forum/6khIivE2KE0>
42. Unruh, D.: Revocable quantum timed-release encryption. *J. ACM* **62**(6), 49:1–49:76 (2015)
43. 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., Sneekenes, E. (eds.) ESORICS 2017. LNCS, vol. 10492, pp. 29–46. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66402-6_4
44. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum private queries. *Phys. Rev. Lett.* **100**(23), 230502 (2008)
45. De Martini, F., Giovannetti, V., Lloyd, S., Maccone, L., Nagali, E., Sansoni, L., Sciarrino, F.: Experimental quantum private queries with linear optics. *Phys. Rev. A* **80**(1), 010302 (2009)
46. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information*, no 2. Cambridge University Press, Cambridge (2000)
47. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_44
48. 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, pp. 474–483. IEEE (2014)
49. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.-M., Chung, K.-M., Persiano, G., Yang, B.-Y. (eds.) PKC 2016. LNCS, vol. 9614, pp. 387–416. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49384-7_15
50. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_21
51. Zhandry, M.: A note on the quantum collision and set equality problems. *Quant. Inf. Comput.* **15**(7–8), 557–567 (2015)
52. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 755–784. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_25
53. Okamoto, T., Pointcheval, D.: REACT: rapid enhanced-security asymmetric cryptosystem transform. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 159–174. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45353-9_13
54. 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.) CT-RSA 2002. LNCS, vol. 2271, pp. 263–276. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45760-7_18