Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model

Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa

NTT Secure Platform Laboratories
3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585 Japan
{saito.tsunekazu, xagawa.keita, yamakawa.takashi}@lab.ntt.co.jp

Abstract. We give a first tight security reduction for a conversion from a weakly-secure public-key encryption scheme to an IND-CCA-secure key-encapsulation mechanism scheme in the quantum random oracle model. To the best of our knowledge, previous reductions are non-tight as the security levels of the obtained schemes are degraded to at most *half or quater* of the original security level (Boneh, Dagdelen, Fischlin, Lehmann, Schafner, and Zhandry (CRYPTO 2012), Targhi and Unruh (TCC 2016-B), and Hofheinz, Hövelmanns, and Kiltz (TCC 2017)).
keywords: Tight security, chosen-ciphertext security, post-quantum cryptography, KEM.

1 Introduction

Let us consider a cryptographic primitive P based on the hardness of a problem S. As a reductionist, we prove the security of P by giving an algorithm R solving S, where R can access to an adversary A (often in the black-box way) who breaks the security of P. Let A's running time and success probability be T and ϵ, respectively. Let R's running time and success probability be T' and ϵ', respectively. The reduction is said to be *tight* if $T' \approx T$ and $\epsilon' \approx \epsilon$. The tightness gap is defined as $(T'/\epsilon')/(T/\epsilon)$, since we consider T'/ϵ' as an expected time to solve S.

The security level of cryptographic schemes strongly depends on that of underlying assumptions *and* the tightness/looseness of the security reductions. If the security reduction is tight and the underlying problem is expected to have b-bit hardness, then we can say that the security level of P is also b-bit. On the other hand, that is, if the security reduction is non-tight, then we cannot estimate the security level of P immediately: In the optimistic scenario, we hoped the existence of tighter reductions that those we have. In the "nightmare" scenario due to Menezes [Men12], the primitive is really insecure but the attacks are still hidden. Therefore, if the security reduction is loose and we are pessimistic, then we are required to set parameters large at the cost of slower performance.

Pre-Quantum Security of IND-CCA PKE/KEM: For asymmetric encryption, public-key encryption (PKE) and key-encapsulation mechanism (KEM), we already have a lot of generic conversions from weakly-secure primitives into strongly-secure PKEs in

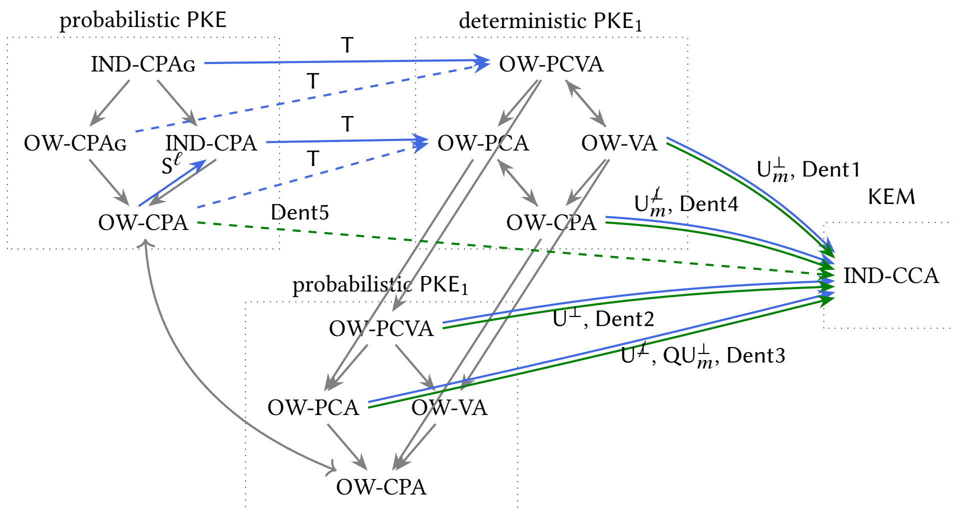

Fig. 1. Transformations in the ROM. GOAL-ATTACK$_G$ indicate that the class of PKEs which is GOAL-ATTACK-secure and $\omega(1)$-spreading [FO00,FO99]. Solid arrows indicate tight reductions, dashed arrows indicate non-tight reductions, thin arrows indicates trivial reductions, thick NGreen arrows indicates reduction in [Den03], and thick RoyalBlue arrows indicates reductions in [HHK17].

the random-oracle model (ROM); BR93 [BR93], OAEP [BR95,FOPS04], REACT [OP01], GEM [CHJ+02], FO-PKC [FO00], FO [FO99,FO13], and so on.

Dent studied five KEM variants of those conversions [Den03] and Hofheinz, Hövelmanns, and Kiltz also investigate KEM variants of the FO conversion in modular way (and in the quantum setting) [HHK17]. We summarized their results in classical setting in Figure 1, which also includes Dent's conversions. For example, we obtain a KEM variant Dent5 of the FO conversion in [Den03, Table 5] as the combination of $U^\perp \circ T$. In Figure 1, solid arrows indicate tight reductions, that is, the tightness gap is a constant. The results say that we have a tight security reduction for the proof that the KEM scheme obtained by applying $U^{\not\perp}_m \circ T \circ S^\ell$ to OW-CPA-secure PKE is IND-CCA-secure in the ROM.

Post-Quantum Security of IND-CCA PKE/KEM: Let us consider a quantum adversary, who poses a scalable quantum computer. (But, we stick classical implementation of our primitives.) Unfortunately, the security reductions in the above papers except [HHK17] considered only the classical setting; that is, they consider classical adversaries and classical random oracles. Thus, if we want to used the conversions in quantum setting, we are required to verify the security reductions or give new security reductions.

Notice that a quantum adversary can implement hash functions *quantumly* by itself. Hence, the adversary can evaluate $|x, y\rangle \mapsto |x, y \oplus H(x)\rangle$ by a quantum circuit and obtain $\sum_x |x, H(x)\rangle$ from $\sum_x |x\rangle$, the superposition of hash values. Thus, it is natural to define *quantum random oracles* and *the quanutm random oracle model (QROM)* to consider the post-quantum cryptography. (See, e.g., Boneh, Dagdelen, Fischlin, Lehmann, Schaffner, and Zhandry [BDF+11].) On strongly-secure asymmetric (or hybrid) encryption schemes, there are a few studies in the QROM:

Variant of BR93: Boneh et al. [BDF+11] showed IND-CCA security of a variant of BR93 PKE in the QROM. Their assumptions are one-time CCA-secure symmetric-key encryption and injective trapdoor functions. Their proof is non-tight.

Variant of FO: Targhi and Unruh [TU16] proposed a variant of the Fujisaki-Okamoto conversion [FO99,FO13], which we call the TU conversion and denote by TU: In the variant, they introduce another hash value $\mathsf{H}'(m)$ to a ciphertext of PKE scheme obtained by FO. They showed IND-CCA security of the obtained PKE in the QROM assuming that the underlying PKE is OW-CPA-secure and $\omega(1)$-spreading. The reduction for TU degrades the security level to approximately the *quarter* of the original security level even ignoring the number of queries, that is, the proof shows that $\epsilon_{\text{ind-cca}} \leq \mathrm{poly}(q_{\mathsf{Hash}}, q_{\mathsf{Dec}}) \cdot \epsilon_{\text{ow-cpa}}^{1/4} + \mathsf{negl}(\kappa)$. Moreover, their simulation employs Zhandry's method [Zha12] to simulate the random oracle H' with $2q$-degree random polynomial over a field. They exploited the roots of polynomial to compute candidates of δ and simulated the decryption oracle by those candidates. Thus, evaluations of H' requires $O(q^2)$ costs and simulation of the decryption oracle requires more. Hence, the reduction is non-tight from the view of time complexity.

Modular Analysis of the variant of FO: Recently Hofheinz, Hövelmanns and Kiltz [HHK17] proposed a KEM variant of TU and analyzed it in modular way and in the quantum setting. They observed that the KEM variants of the TU conversion, denoted by QFO_m^*, is decomposed into two conversions, T and QU_m^*; T converts PKE to PKE and QU_m^* converts PKE to KEM.

Variant of OAEP: Targhi and Unruh [TU16] also showed IND-CCA security of the variant of OAEP in the QROM assuming the existence of partial-domain one-way trapdoor functions. The security reductions are looser then those for TU.

Simulation of QRO: Zhandry [Zha12, Sections 3 and 6] showed that, if the number of queries is q, then the random oracle can be *perfectly simulated* by $2q$-wise independent functions in the quantum setting as the random oracle can be *perfectly simulated* by q-wise independent hash functions in the classic setting.

For the summary, see Figure 2. As far as we know, the IND-CCA-secure KEM in the QROM with the *tightest* reduction is the KEM scheme obtained by applying the $\mathsf{QU}_m^{\perp}$ or $\mathsf{QU}_m^{\not\perp}$ conversion to a OW-PCA-secure PKE scheme. The security reduction results in $\epsilon_{\text{ind-cca}} \leq 3q \cdot \epsilon_{\text{ow-pca}}^{1/2}$ and $T_{\text{ind-cca}} \approx T_{\text{ow-cpa}} + \Omega(q^2)$, where q denotes the sum of the numbers of hash and decryption queries. (See [HHK17, Thm. 4.5 and 4.6].)

We list existing IND-CCA-secure KEM/PKE schemes in the QROM in Table 1. All but NTRU Prime and RLCE employed variants of QFO_*^* and suffered from loose reductions with quartic loss. Even NTRU Prime is suffered from the loose reduction with quadratic loss. As far as we know, the existing security reductions in the QROM are loose. It is quite natural to ask that

> *Can we construct tightly-secure conversions from CPA-secure primitives to IND-CCA KEM in the QROM?*

Table 1. Existing CCA-secure KEM/PKE schemes in the QROM. pOW-CPA indicates that the underlying scheme is assumed as probabilistic and one-way against chosen-plaintext attacks; dOW-PCA indicates that the underlying scheme is assumed as deterministic and one-way against plaintext-checking attacks; QFO^X denotes $\mathsf{QU}^X \circ \mathsf{T}$ for $X \in \{\perp, \not\perp\}$.

Primitive	ref.	Name	Assumption	Conversion	Notes
KEM	[BDK+17]	Kyber	pOW-CPA	modified $\mathsf{QFO}^{\not\perp}$	involves *ek*
	[BCLvVxx]	NTRU Prime	dOW-PCA	$\mathsf{QU}_m^{\perp}$	
	[HRSS17]	NTRU HRSS	pOW-CPA	modified $\mathsf{QFO}^{\perp}$	*m* is generated form seed
	[BGG+17]	CAKE	pOW-CPA	$\mathsf{QFO}^{\perp}$	
	[Wan17]	RLCE	?	RLCEpad	a variant of OAEP+
	[Ham17]	ThreeBearsCCA	pOW-CPA	TU	ver.7, Sec.4
PKE	[CHK+16]	CHK+ PKE2	pOW-CPA	TU	
	[CKLS16]	CCALizard	pOW-CPA	TU	

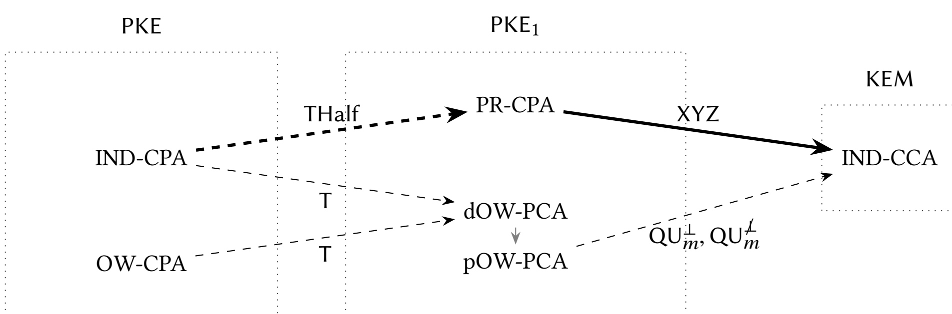

Fig. 2. Transformations in the QROM. Solid arrows indicate quantum tight reductions, dashed arrows indicate quantum non-tight reductions, thin arrows indicates existing reductions in [HHK17], and thick arrows indicates our new reductions.

1.1 Our Contributions

New Security Notion, PR-CPA: We first give a new (but seemingly folklore) security notion, PR-CPA security of *deterministic PKE*. A deterministic PKE scheme is PR-CPA if, there exist an efficient fake key-generation algorithm and a fake encryption algorithm, such that, 1) a real and fake encryption key are indistinguishable, 2) random real and fake ciphertexts on a fake key are indistinguishable, and 3) the probability that a random fake ciphertext on a fake key falls in the range of a real ciphertext on the fake key is negligible.

It is easy to find PR-CPA PKE schemes from post-quantum cryptography, say, NTRU [HPS98,SS11], the GPV TDF [GPV08], McEliece PKE [McE78], and Niederreiter PKE [Nie86]. We notice that the above schemes have similar security proofs: We can replace their keys with random keys under appropriate assumptions and random ciphertexts on the random key with completely random ciphertexts under the LPN/LWE assumptions. Moreover the random ciphertexts on the random key are exponentially sparser than the completely random ciphertexts and, thus, and random and fake ciphertexts on a fake key are overlapped only in negligible amount. See Section 3 for the detail.

PR-CPA from IND-CPA: In addition, we find that it is easy to construct PR-CPA-secure PKE scheme from any IND-CPA-secure PKE scheme whose plaintext spaces are exponentially large.

Recall that, T in [HHK17] converts probabilistic IND-CPA (or OW-CPA) PKEs into a *deterministic* OW-PCA PKE $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, where randomness in encryption is fixed as $r := \mathsf{G}(m)$, the hash value of m.

We give another transformation THalf which is essentially same as T except that THalf halves the plaintext space and employ only one of the two, while T keeps the plaintext space as the original one. The other part is used for fake ciphertexts. Unfortunately, the quantum reduction suffers from loose reduction with the quadratic loss as the reduction for T does. See Figure 2. We will prove this in Section 4.

Tight Reduction, XYZ: We propose a new conversion XYZ, which is a KEM variant of the BR93 conversion and is essentially equivalent to $\mathsf{U}_m^{\not\perp}$. We succeed to show a *tight* security reduction in the QROM by requiring an underlying PKE scheme to be PR-CPA-secure and by giving a new and simple proof, which is easily understandable (essentially without quantum knowledge): We show that

> $\mathsf{KEM} = \mathsf{XYZ}[\mathsf{PKE}_1, \mathsf{H}, \mathsf{PRF}, \mathsf{PRF}']$ is IND-CCA secure *tightly in the QROM* if PKE_1 is *deterministic* and PR-CPA, and PRF and PRF' are quantumly-secure pseudo-random functions.

Roughly speaking, our reduction results in

$$\epsilon_{\text{ind-cca}} \leq 2\epsilon_{\text{pr-cpa}} + 4\epsilon_{\text{prf}} + \mathsf{negl}(\kappa) \text{ and } T_{\text{pr-cpa}}, T_{\text{prf}} \approx T_{\text{ind-cca}} + O(q \cdot \text{poly}(\kappa)),$$

where $\epsilon_{\text{pr-cpa}}$ is the max. of the advantage for PR-CPA security, ϵ_{prf} is the max. of the advantages of PRFs. This drastically improves the previous non-tight reductions. We note that we can remove ϵ_{PRF} and $\epsilon_{\mathsf{PRF}'}$ by replacing PRF and PRF' with quantum random oracles and by invoking Zhandry's simulation method [Zha12]. However, this also replaces $O(q \cdot \text{poly}(\kappa))$ with $O(q^2 \cdot \text{poly}(\kappa))$.

See Section 5 for the details.

Implementations: We implement our conversion upon NTRU-HRSS [HRSS17] over a desktop PC and a RasPi. Assuming that NTRU-HRSS is PR-CPA, the obtained KEM is CCA secure in the QROM. See Section 6.

Open Problems: We leave interesting open problems for IND-CCA security of asymmetric encryption in the post-quantum setting:

1. Can we remove the stronger requirements for PKE_1, deterministic and pseudo-random?
2. Can we construct (almost) tightly-secure IND-CCA2 PKE/KEM *in the multi-user and multi-challenge setting* and in the QROM?

2 Preliminaries

Notation: A security parameter is denoted by κ. We use the standard O-notations, O, Θ, Ω, and ω. The abbreviations DPT and PPT stand for deterministic polynomial time and probabilistic polynomial time. A function $f(\kappa)$ is said to be *negligible* if $f(\kappa) = \kappa^{-\omega(1)}$. We denote a set of negligible functions by $\mathsf{negl}(\kappa)$. For two finite sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathrm{Map}(\mathcal{X}, \mathcal{Y})$ denotes a set of all functions whose domain is $\mathcal{X}$ and codomain is $\mathcal{Y}$.

For a distribution χ, we often write "$x \leftarrow \chi$", which indicates that we take a sample x from χ. For a finite set S, $U(S)$ denotes the uniform distribution over S. We often write "$x \leftarrow S$" instead of $x \leftarrow U(S)$.

If inp is a string, then "$\mathsf{out} \leftarrow A(\mathsf{inp})$" denotes the output of algorithm A when run on input inp. If A is deterministic, then out is a fixed value and we write "$\mathsf{out} := A(\mathsf{inp})$"; We also use the notation "$\mathsf{out} := A(\mathsf{inp}; r)$" to make the randomness r explicit.

For the Boolean statement P, $\mathsf{bool}(P)$ denote the bit that is 1 if P is true, and otherwise 0. For example, $\mathsf{bool}(b' \stackrel{?}{=} b)$ is 1 if and only if $b' = b$.

Quantum Computation: We refer to [NC00] for basic of quantum computation.

The following lemma is taken from [HHK17], a wrapper of the oneway-to-hiding (OW2H) lemma [Unr15, Lemma 6.2]. Roughly speaking, the lemma states that if any quantum adversary issuing at most q queries to H can distinguish $(x, \mathsf{H}(x))$ from (x, y), where y is chosen uniformly at random, then we can find x by measuring one of the adversary's query.

Lemma 2.1 (Algorithmic Oneway to Hiding [HHK17,Unr15]). *Let $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ be a quantum random oracle, let $\mathcal{A}$ be an adversary issuing at most q queries to H that on input $(x, y) \in \mathcal{X} \times \mathcal{Y}$ outputs either $0/1$. For all (probabilistic) algorithms F whose input space is $\mathcal{X} \times \mathcal{Y}$ and which do not make any hash queries to H, we have*

$$\begin{aligned}
&\left| \begin{aligned} &\Pr[\mathcal{A}^{\mathsf{H}}(\mathrm{inp}) \to 1 \mid x \leftarrow \mathcal{X}; \mathrm{inp} \leftarrow \mathsf{F}(x, \mathsf{H}(y))] \\ &\quad - \Pr[\mathcal{A}^{\mathsf{H}}(\mathrm{inp}) \to 1 \mid (x, y) \leftarrow \mathcal{X} \times \mathcal{Y}; \mathrm{inp} \leftarrow \mathsf{F}(x, y)] \end{aligned} \right| \\
&\leq 2q \cdot \sqrt{\Pr[\mathsf{EXT}^{\mathcal{A},\mathsf{H}}(\mathrm{inp}) \to x \mid (x, y) \leftarrow \mathcal{X} \times \mathcal{Y}; \mathrm{inp} \leftarrow \mathsf{F}(x, y)]},
\end{aligned}$$

where EXT picks $i \leftarrow \{1, \ldots, q\}$, runs $\mathcal{A}^{\mathsf{H}}(\mathrm{inp})$ until i-th query $|\hat{x}\rangle$ to H, and returns $x' := \mathsf{Measure}(|\hat{x}\rangle)$ (when $\mathcal{A}$ makes less than i queries, EXT outputs $\perp \notin \mathcal{X}$).

2.1 Key Encapsulation

The model for KEM schemes is summarized as follows:

Definition 2.1. *A KEM scheme KEM consists of the following triple of polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$*:*

- $\mathsf{Gen}(1^\kappa; r_g) \to (ek, dk)$*: a key-generation algorithm which on input 1^κ, where κ is the security parameter, outputs a pair of keys (ek, dk). ek and dk are called encapsulation key and decapsulation key, respectively.*
- $\mathsf{Encaps}(ek; r_e) \to (c, K)$*: an encapsulation algorithm which takes as input encapsulation key ek, outputs ciphertext $c \in \mathcal{C}$ and key $K \in \mathcal{K}$.*

– $\mathsf{Decaps}(dk, c) \to K/\perp$*: a decapsulation algorithm which takes as input decapsulation key dk and ciphertext c, outputs key K or a rejection symbol* $\perp \notin \mathcal{K}$.

Definition 2.2 (Correctness). *We say* $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *has* perfect correctness *if for any* (ek, dk) *generated by* Gen*, we have that*

$$\Pr[\mathsf{Decaps}(dk, c) = K : (c, K) \leftarrow \mathsf{Encaps}(ek)] = 1.$$

Security: The security of KEM schemes is defined by several notions like onewayness and indistinguishability. We recall the definition of indistinguishability under chosen-ciphertext and chosen-plaintext attacks (denoted by IND-CCA and IND-CPA) for KEM, respectively.

Definition 2.3. *A KEM scheme is* (T, ϵ)-IND-CCA secure *if the following property holds for security parameter* κ*; For any adversary* $\mathcal{A}$ *whose running time is at most* T*,*

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1] - 1 \right| \leq \epsilon.$$

We say a KEM scheme is (T, ϵ)-IND-CPA secure*, if* $\mathcal{A}$ *does not access* $\textsc{Dec}$.

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1] - 1 \right| \leq \epsilon.$$

$\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$	$\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$	$\textsc{Dec}_{c^*}(c)$
$b \leftarrow \{0,1\}$	$b \leftarrow \{0,1\}$	if $c = c^*$, return $\perp$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$K := \mathsf{Decaps}(dk, c)$
$(c^*, K_0^*) \leftarrow \mathsf{Encaps}(ek)$;	$(c^*, K_0^*) \leftarrow \mathsf{Encaps}(ek)$;	**return** K
$K_1^* \leftarrow \mathcal{K}$	$K_1^* \leftarrow \mathcal{K}$	
$b' \leftarrow \mathcal{A}(ek, c^*, K_b^*)$	$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(ek, c^*, K_b^*)$	
return $\mathsf{bool}(b' \stackrel{?}{=} b)$	**return** $\mathsf{bool}(b' \stackrel{?}{=} b)$	

Fig. 3. Games for KEM schemes

2.2 Public-Key Encryption

The model for PKE schemes is summarized as follows:

Definition 2.4. *A PKE scheme* PKE *consists of the following triple of polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and a finite message space* $\mathcal{M}$*. We assume that* $\mathcal{M}$ *is efficiently recognizable.*

– $\mathsf{Gen}(1^\kappa; r_g) \to (ek, dk)$: *a key-generation algorithm which on input* 1^κ*, where* κ *is the security parameter, outputs a pair of keys* (ek, dk)*.* ek *and* dk *are called encryption key and decryption key, respectively.*
– $\mathsf{Enc}(ek, m; r_e) \to c$: *an encryption algorithm which takes as input encryption key* ek *and message* $m \in \mathcal{M}$*, outputs ciphertext* $c \in \mathcal{C}$*.*
– $\mathsf{Dec}(dk, c) \to m/\bot$: *a decryption algorithm which takes as input decryption key* dk *and ciphertext* c*, outputs message* $m \in \mathcal{M}$ *or a rejection symbol* $\bot \notin \mathcal{M}$*.*

Definition 2.5. *We say a PKE scheme* PKE *is deterministic if* Enc *is deterministic.*

Definition 2.6 (Correctness). *We say* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* perfect correctness *if for any* (ek, dk) *generated by* Gen *and for any* $m \in \mathcal{M}$ *we have that*

$$\Pr[\mathsf{Dec}(dk, c) = m : c \leftarrow \mathsf{Enc}(ek, m)] = 1.$$

Security: The security of PKE schemes is defined by several notions like onewayness and indistinguishability. Here, we recall the definition of indistinguishability under chosen-ciphertext and chosen-plaintext attacks (denoted by IND-CCA and IND-CPA) for PKE, respectively.

Definition 2.7 (IND-CCA and IND-CPA security). *A PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* (T, ϵ)-IND-CCA secure *if the following property holds for security parameter* κ*; For any adversary* $\mathcal{A}$ *whose running time is at most* T*,*

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] - 1 \right| \leq \epsilon.$$

We say a PKE scheme is (T, ϵ)-IND-CPA secure*, if* $\mathcal{A}$ *cannot access to the decapsulation oracle* $\mathsf{Dec}_*(*)$*; that is,*

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] - 1 \right| \leq \epsilon.$$

Definition 2.8 (OW-CPA security). *A PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* (T, ϵ)-OW-CPA secure *if the following property holds for security parameter* κ*; For any adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\text{ow-cpa}}_{\mathcal{A},\mathsf{PKE}}(\kappa) := \Pr[\mathsf{Expt}^{\text{ow-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = 1] \leq \epsilon,$$

where $\mathcal{A}$ *runs in at most* T *steps.*

2.3 Pseudorandom Functions

A pseudorandom function (PRF) is a polynomial-time computable function of form $\mathsf{PRF}\colon \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$. We call the sets $\mathcal{S}$, $\mathcal{X}$, $\mathcal{Y}$ as the key space, the domain, and the codomain of PRF, respectively.

Definition 2.9. *We say* PRF *is secure, if for any (QPT) adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}_{\mathsf{PRF},\mathcal{A}}(\kappa) := \left| \Pr[\mathcal{A}^{\mathsf{PRF}(s,\cdot)}(1^\kappa) = 1] - \Pr[\mathcal{A}^{\rho(\cdot)}(1^\kappa) = 1] \right|$$

is negligible in κ*, where* $s \leftarrow \mathcal{S}$*,* $\rho \leftarrow \mathrm{Map}(\mathcal{X}, \mathcal{Y})$ *are uniformly and independently random.*

$\mathsf{Expt}^{\text{ow-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$	$\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$	$\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$	$\textsc{Dec}_a(c)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$b \leftarrow \{0,1\}$	$b \leftarrow \{0,1\}$	if $c = a$, return $\perp$
$m^* \leftarrow \mathcal{M}$	$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$m := \mathsf{Dec}(dk, c)$
$c^* \leftarrow \mathsf{Enc}(ek, m^*)$	$(m_0, m_1, st) \leftarrow \mathcal{A}_1(ek)$	$(m_0, m_1, st) \leftarrow \mathcal{A}_1^{\textsc{Dec}_\perp(\cdot)}(ek)$	**return** m
$m' \leftarrow \mathcal{A}(ek, c^*)$	$c^* \leftarrow \mathsf{Enc}(ek, m_b)$	$c^* \leftarrow \mathsf{Enc}(ek, m_b)$	
return $\mathsf{bool}(m' \stackrel{?}{=} \mathsf{Dec}(dk, c^*))$	$b' \leftarrow \mathcal{A}_2(c^*, st)$	$b' \leftarrow \mathcal{A}_2^{\textsc{Dec}_{c^*}(\cdot)}(c^*, st)$	
	return $\mathsf{bool}(b' \stackrel{?}{=} b)$	**return** $\mathsf{bool}(b' \stackrel{?}{=} b)$	

Fig. 4. Games for PKE schemes

We additionally require joint security of PRFs: Let $\mathsf{PRF}_i \colon \mathcal{S}_i \times \mathcal{X}_i \to \mathcal{Y}_i$ be PRFs for $i = 1, \ldots, k$.

Definition 2.10. *We say PRFs* $\mathsf{PRF}_1, \ldots, \mathsf{PRF}_k$ *are jointly-secure, if for any (QPT) adversary* $\mathcal{A}$*, we have*

$$\mathsf{Adv}_{\mathsf{PRF}_1+\cdots+\mathsf{PRF}_k,\mathcal{A}}(\kappa) := \left|\Pr[\mathcal{A}^{\mathsf{PRF}_1(s_1,\cdot),\ldots,\mathsf{PRF}_k(s_k,\cdot)}(1^\kappa) = 1] - \Pr[\mathcal{A}^{\rho_1(\cdot),\ldots,\rho_k(\cdot)}(1^\kappa) = 1]\right|$$

is negligible in κ*, where* $s_i \leftarrow \mathcal{S}$*,* $\rho_i \leftarrow \mathrm{Map}(\mathcal{X}_i, \mathcal{Y}_i)$ *are uniformly and independently random.*

Remark 2.1. One might wonder why we define joint security of PRFs, because it is well-known that the securities of each PRF implies joint security of PRFs in the classical setting. Recall that, in the proof of joint security, the hybrid games are introduced. We then construct reduction algorithms that simulate the hybrid games. Notice that the reduction algorithms are required to simulate the random oracles. In the classical-query setting, it is easy to simulate the random oracle *on the fly* and the simulation adds the time approximately $O(q)$ operations if we carefully design the hash table.

Meanwhile, quantum adversaries can make quantum queries to their oracles. Thus, we cannot employ the on-the-fly simulation of the random oracles. Zhandry's theorem shows that if we know the number of queries, q, then we take a random function f from $2q$-wise independent hash functions, and replace the random oracle ρ by f. To the best of our knowledge, the most efficient $2q$-wise independent hash functions requires the computational time $\Theta(q)$ operations per evaluation. This results in the additional $\Theta(q^2)$ operations to simulate the random oracle, which makes the security reduction *non-tight*.

Therefore, we adopt an option that we just assume joint security of PRFs.

3 PR-CPA security of PKE

We formally define our new security notion, PR-CPA, of deterministic PKE. We require two additional PPT algorithms $\widetilde{\mathsf{Gen}}$ and $\widetilde{\mathsf{Enc}}$: $\widetilde{\mathsf{Gen}}$ is a PPT algorithm that takes the security parameter as input and outputs a fake encryption key $\widetilde{ek}$, which is indistinguishable from a real encryption key. This means that the original encryption algorithm Enc should be able to encrypt a message even with a fake encryption key. $\widetilde{\mathsf{Enc}}$ is a PPT algorithm that takes a fake encryption key as input and outputs a random fake ciphertext, which is indistinguishable from a random real ciphertext with a fake encryption key. We further require that the probability that a random fake ciphertext with a fake encryption key falls in the range of a real ciphertext with a fake encryption key is negligible. For example, this condition is satisfied if a set of real ciphertexts is sufficiently sparser than a set of fake ciphertext or a set of real ciphertexts is disjoint with a set of fake ciphertext. The formal definition follows:

Definition 3.1. *A deterministic PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with plaintext and ciphertext spaces* $\mathcal{M}$ *and* C *is* $(T, \epsilon_{\mathrm{disj}}, \epsilon_{\text{pr-key}}, \epsilon_{\text{pr-cipher}})$-PR-CPA secure *if the following property holds for security parameter* κ*; There exist two PPT algorithms* $\widetilde{\mathsf{Gen}}$ *and* $\widetilde{\mathsf{Enc}}$ *that satisfy the followings:*

- *(Statistical Disjointness:) for any* $\widetilde{ek}$ *generated by* $\widetilde{\mathsf{Gen}}(1^\kappa)$*, the probability that a fake ciphertext is in the range of a real ciphertext generated by* $\mathsf{Enc}(\widetilde{ek}, \cdot)$ *is negligible, that is,*
$$\Pr[c \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{ek}) : c \in \mathsf{Enc}(\widetilde{ek}, \mathcal{M})] = \epsilon_{\mathrm{disj}}(\kappa).$$
- *(PR-Key Security:) for any adversary* $\mathcal{A}$*, its advantage to distinguish a real key from a fake key, denoted by* $\mathsf{Adv}^{\text{pr-key}}_{\mathcal{A},\mathsf{PKE}}(\kappa)$*, is at most* ϵ*;*
$$\mathsf{Adv}^{\text{pr-key}}_{\mathcal{A},\mathsf{PKE}}(\kappa) := \left| \begin{array}{l} \Pr\left[(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); b' \leftarrow \mathcal{A}(ek) : b' = 1\right] \\ \quad - \Pr\left[\widetilde{ek} \leftarrow \widetilde{\mathsf{Gen}}(1^\kappa); b' \leftarrow \mathcal{A}(\widetilde{ek}) : b' = 1\right] \end{array} \right| \leq \epsilon_{\text{pr-key}},$$
where $\mathcal{A}$ *runs in at most* T *steps.*
- *(PR-Ciphertexts Security:) for any adversary* $\mathcal{A}$*, its advantage to distinguish a real ciphertext from a fake ciphertext with a fake key, denoted by* $\mathsf{Adv}^{\text{pr-cipher}}_{\mathcal{A},\mathsf{PKE}}(\kappa)$*, is at most* ϵ*;*
$$\mathsf{Adv}^{\text{pr-cipher}}_{\mathcal{A},\mathsf{PKE}}(\kappa) := \left| \begin{array}{c} \Pr\left[\begin{array}{c} \widetilde{ek} \leftarrow \widetilde{\mathsf{Gen}}(1^\kappa); m^* \leftarrow \mathcal{M}; c^* := \mathsf{Enc}(\widetilde{ek}, m^*); \\ b' \leftarrow \mathcal{A}(\widetilde{ek}, c^*) : b' = 1 \end{array}\right] \\ - \Pr\left[\begin{array}{c} \widetilde{ek} \leftarrow \widetilde{\mathsf{Gen}}(1^\kappa); c^* \leftarrow \widetilde{\mathsf{Enc}}(\widetilde{ek}); \\ b' \leftarrow \mathcal{A}(\widetilde{ek}, c^*) : b' = 1 \end{array}\right] \end{array} \right| \leq \epsilon_{\text{pr-cipher}},$$
where $\mathcal{A}$ *runs in at most* T *steps.*

3.1 Examples

We found that NTRU, the GPV TDFs, the McEliece PKE, and the Niederreiter PKE are PR-CPA-secure under certain assumptions if their parameters are carefully chosen.

- (Perfect Correctness) First, we require them to be perfectly correct; this can be satisfied their noise parameter sufficiently smaller.

$\mathsf{Gen}_1(1^\kappa)$	$\mathsf{Enc}_1(ek, m)$, where $m \in \mathcal{M}_{\mathrm{even}}$	$\mathsf{Dec}_1(dk, c)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$r := \mathsf{G}(m)$	$m := \mathsf{Dec}(dk, c)$
return (ek, dk)	$c := \mathsf{Enc}(ek, m; r)$	**if** $m \notin \mathcal{M}_{\mathrm{even}}$ **return** $\perp$
	return c	**else return** m

$\widetilde{\mathsf{Gen}_1}(1^\kappa)$	$\widetilde{\mathsf{Enc}_1}(ek)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$m \leftarrow \mathcal{M}_{\mathrm{odd}},\ r \leftarrow \mathcal{R}$
return ek	$c := \mathsf{Enc}(ek, m; r)$
	return c

Fig. 5. $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1) = \mathsf{THalf}[\mathsf{PKE}, \mathsf{G}]$ with $\widetilde{\mathsf{Gen}_1}$ and $\widetilde{\mathsf{Enc}_1}$

- (PR-Keys) Second, in the proofs of semantic security (IND-CPA security) of NTRU, McEliece, and Niederreiter, we often employ the assumptions that their encryption keys are indistinguishable from random keys. Thus, the assumption just states PR-Keys security. In the case of GPV, the public key is statistically indistinguishable from random keys.
- (PR-Ciphertexts) Third, we know that after we replace their encryption keys random, then their random ciphertexts with random keys are indistinguishable from random. Hence, we just define $\widetilde{\mathsf{Enc}_1}$ as a sampler from the ambient spaces; $\mathbb{Z}_q[x]/(x^n - 1)$ for NTRU, $\mathbb{Z}_q^m$ for GPV, $\mathbb{F}^m$ for McElice, and $\mathbb{F}^n$ for Niederreiter.
- (Disjointness) Forth, we know that the ciphertext spaces are exponentially sparser than the ambient spaces in them. Thus, the disjointness easily follows.

We also show that any perfectly-correct, IND-CPA-secure PKE whose plaintext space is sufficiently large can be converted into PR-CPA-secure PKE_1 by using the random oracle G. See Section 4 for the details.

4 Conversion from IND-CPA to PR-CPA

We propose a new conversion THalf from IND-CPA-secure PKE PKE to *deterministic* PR-CPA-secure PKE PKE_1, which is a variant of T. Let $\mathcal{M}$ and $\mathcal{R}$ be the message and randomness spaces of PKE, respectively. Suppose that $\mathcal{M}$ is divided into two disjoint, sampleable spaces, $\mathcal{M} = \mathcal{M}_{\mathrm{even}} \sqcup \mathcal{M}_{\mathrm{odd}}$. (For example, $\mathcal{M}_{\mathrm{even}}$ and $\mathcal{M}_{\mathrm{odd}}$ are even and odd numbers in $\mathcal{M}$.) We set the message space of PKE_1 as $\mathcal{M}_{\mathrm{even}}$, the half of $\mathcal{M}$. Let $\mathsf{G}\colon \mathcal{M}_{\mathrm{even}} \to \mathcal{R}$ be a random oracle. We denote $\mathsf{PKE}_1 = \mathsf{THalf}[\mathsf{PKE}, \mathsf{G}] = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$. The algorithms are defined in Figure 5. We additionally require a PRF $\mathsf{PRF}\colon \mathcal{S} \times \mathcal{M}_{\mathrm{even}} \to \mathcal{R}$ for the security proof.

The proofs are very similar to those of [TU16] and [HHK17].

Theorem 4.1 (Classical Reduction). *Let* PKE *be a PKE scheme. For any PR-CPA adversary $\mathcal{A}$ against* PKE_1 *issuing at most q_{G} queries to* G*, there exist two two IND-CPA* *adversaries $\mathcal{A}_{\mathsf{PKE}}$ and $\mathcal{A}'_{\mathsf{PKE}}$ against* PKE *such that*

$$\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = 0$$
$$\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) \leq 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}'_{\mathsf{PKE}}}(\kappa) + \frac{q_{\mathsf{G}}}{\#\mathcal{M}_{\text{even}}},$$

and their running times are about that of $\mathcal{A}$.

The proof of Theorem 4.1 is in Appendix B.

Theorem 4.2 (Quantum Reduction). *Let* PKE *be a PKE scheme. For any PR-CPA quantum adversary $\mathcal{A}$ against* PKE_1 *issuing at most q_{G} queries to* G*, there exist two IND-CPA quantum adversaries $\mathcal{A}_{\mathsf{PKE}}$ and $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ against* PKE *and three quantum adversaries $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PRF}}$, $\mathcal{A}^{1}_{\mathsf{PRF}}$, and $\mathcal{A}^{2}_{\mathsf{PRF}}$ against* PRF*, such that*

$$\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = 0$$
$$\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) \leq 2q_{\mathsf{G}}\sqrt{2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PRF}}}(\kappa) + 1/\#\mathcal{M}_{\text{even}}}$$
$$+ \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^{1}_{\mathsf{PRF}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^{2}_{\mathsf{PRF}}}(\kappa)$$

and their running times are about that of $\mathcal{A}$.

The proof of Theorem 4.2 follows.

4.1 Quantum Proofs

It is obvious that $\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = 0$, since $\mathsf{Gen}_1 = \widetilde{\mathsf{Gen}_1}$. It is also obvious that the output of $\widetilde{\mathsf{Enc}_1}(ek)$ never overlaps with $\mathsf{Enc}_1(ek,\mathcal{M}_{\text{even}}) \subseteq \mathsf{Enc}(ek,\mathcal{M}_{\text{even}};\mathcal{R})$, because PKE is perfectly correct and the range of $\widetilde{\mathsf{Enc}_1}(ek)$ is $\mathsf{Enc}(ek,\mathcal{M}_{\text{odd}};\mathcal{R})$.

Table 2. Summary of Games for the Security Proof in the QROM

Game	m^*	r^*	c^*	G	justification
Game_0	$\mathcal{M}_{\text{even}}$	$\mathsf{G}(m^*)$	$\mathsf{Enc}(ek,m^*;r^*) = \mathsf{Enc}_1^{\mathsf{G}}(ek,m^*)$	$\mathsf{G}(\cdot)$	
Game_1	$\mathcal{M}_{\text{even}}$	r^*	$\mathsf{Enc}(ek,m^*;r^*)$	$\mathsf{G}(\cdot)$	IND-CPA security of PKE and the OW2H lemma
Game'_1	$\mathcal{M}_{\text{even}}$	r^*	$\mathsf{Enc}(ek,m^*;r^*)$	$\mathsf{PRF}(s,\cdot)$	PRF security of PRF
Game'_2	$\mathcal{M}_{\text{odd}}$	r^*	$\mathsf{Enc}(ek,m^*;r^*) = \widetilde{\mathsf{Enc}_1}(ek)$	$\mathsf{PRF}(s,\cdot)$	IND-CPA security of PKE
Game_2	$\mathcal{M}_{\text{odd}}$	r^*	$\mathsf{Enc}(ek,m^*;r^*) = \widetilde{\mathsf{Enc}_1}(ek)$	G	PRF security of PRF

In the rest of this section, we give a non-tight security proof for pseudorandomness of ciphertexts. What we want to show is the upper bound of

$$\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = |\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_2 = 1]|\,.$$

Game_0: We expand algorithms and obtain Game_0:

$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\text{even}}; r^* \leftarrow \mathsf{G}(m^*); c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'$.

Game_1: This game is the same as Game_0 except that the randomness of the challenge ciphertext is freshly generated:

$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\text{even}}; r^* \leftarrow \mathcal{R}; c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'$.

Applying the Algorithmic-OW2H lemma (Lemma 2.1) with $\mathcal{X} = \mathcal{M}_{\text{even}}$, $\mathcal{Y} = \mathcal{R}$, $x = m^*$, $y = r^*$, algorithms F and $\mathsf{EXT}[\mathcal{A}, \mathsf{G}]$, and game Hyb in Figure 6, we have

$$|\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_1 = 1]| \leq 2q_{\mathsf{G}}\sqrt{\Pr[\mathsf{Hyb} = 1]}.$$

Game_2: This game is the same as Game_1 except that the challenge ciphertext is generated by $\mathsf{Enc}(ek, m^*; r^*)$, where $m^* \leftarrow \mathcal{M}_{\text{odd}}$ rather than $m^* \leftarrow \mathcal{M}_{\text{even}}$:

$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\text{odd}}; r^* \leftarrow \mathcal{R}; c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'$.

In addition, for $i = 0, 1$, we define intermediate games Game'_i, in which we employ PRF $\mathsf{PRF}(s, \cdot) \colon \mathcal{M}_{\text{even}} \to \mathcal{R}$ with random key $s' \leftarrow \mathcal{S}$ instead of $\mathsf{G} \colon \mathcal{M}_{\text{even}} \to \mathcal{R}$.

It is straightforward to construct quantum reduction algorithms $\mathcal{A}^1_{\mathsf{PRF}}$, $\mathcal{A}^2_{\mathsf{PRF}}$, and satisfying

$$\left|\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}'_1 = 1]\right| \leq \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF}, \mathcal{A}^1_{\mathsf{PRF}}}(\kappa),$$
$$\left|\Pr[\mathsf{Game}'_2 = 1] - \Pr[\mathsf{Game}_2 = 1]\right| \leq \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF}, \mathcal{A}^2_{\mathsf{PRF}}}(\kappa).$$

Their running times are about that of $\mathcal{A}$.

Moreover, we have a quantum reduction algorithm $\mathcal{A}_{\mathsf{PKE}}$ satisfying

$$\left|\Pr[\mathsf{Game}'_1 = 1] - \Pr[\mathsf{Game}'_2 = 1]\right| = \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa).$$

We define $\mathcal{A}_{\mathsf{PKE}}$ as follows:

- On input ek, $\mathcal{A}_{\mathsf{PKE}}$ chooses two messages $m_0 \leftarrow \mathcal{M}_{\text{even}}$ and $m_1 \leftarrow \mathcal{M}_{\text{odd}}$ uniformly at random. It queries them to its challenge oracle and obtains $c^* \leftarrow \mathsf{Enc}(ek, m^*; r^*)$, where m^* is m_b. It invokes $\mathcal{A}$ with ek and c^*. It also chooses key of PRF as $s \leftarrow \mathcal{S}$ to simulate the oracle.
- $\mathcal{A}_{\mathsf{PKE}}$ simulates the random oracle G by computing

$$\sum_x |x\rangle\,|y\rangle \mapsto \sum_x |x\rangle\,|\mathsf{PRF}(s, x) \oplus y\rangle\,.$$

- Eventually, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}_{\mathsf{PKE}}$ outputs b' also.

It is obvious that $\mathcal{A}_{\mathsf{PKE}}$ perfectly simulates Game_{b+1} depending on the challenge bit $b \in \{0,1\}$. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}'_{\mathsf{PKE}}}(\kappa) &= |\Pr[b'=b] - 1/2| \\ &= |(1-\Pr[b'=1 \mid b=0]) + \Pr[b'=1 \mid b=1] - 1| \\ &= |1 - \Pr[\mathsf{Game}_1 = 1] + \Pr[\mathsf{Game}_2 = 1] - 1| \\ &= |\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_1 = 1]|\end{aligned}$$

as we wanted. The running time is given as

$$T_{\mathcal{A}_{\mathsf{PKE}}} \approx T_{\mathcal{A}} + O(q_{\mathsf{G}} \cdot T_{\mathsf{PRF}}).$$

Hyb*:* Finally, we upperbound $\Pr[\mathsf{Hyb} = 1]$.

Let us introduce another hybrid game Hyb', in which we replace G with $\mathsf{PRF}(s,\cdot)$. It is straightforward to construct a quantum reduction algorithm $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PRF}}$ satisfying

$$|\Pr[\mathsf{Hyb}=1] - \Pr[\mathsf{Hyb}'=1]| \le \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PRF}}}(\kappa).$$

Their running times are about that of $\mathcal{A}$.

Let $\gamma := \Pr[\mathsf{Hyb}' = 1]$. Let us construct a reduction algorithm $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ against IND-CPA security of PKE as follows:

- On input ek, $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ chooses $s \leftarrow \mathcal{S}$ and chooses two messages $m_0 \leftarrow \mathcal{M}_{\mathrm{even}}$ and $m_1 \leftarrow \mathcal{M}_{\mathrm{odd}}$ uniformly at random. It then queries m_0, m_1 to its challenge oracle and obtains $c^* \leftarrow \mathsf{Enc}(ek, m^*; r^*)$, where m^* is m_b. It invokes $\mathsf{EXT}[\mathcal{A}, \mathsf{PRF}(s,\cdot)]$ with ek and c^*.
- $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ can simulate the oracle $\mathsf{PRF}(s,\cdot)$ because it knows s.
- Eventually, $\mathsf{EXT}[\mathcal{A}, \mathsf{PRF}(s,\cdot)]$ outputs m'. $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ outputs $b' = 0$ if $m' = m_0$. Otherwise, it outputs $b' \leftarrow \{0,1\}$.

If the challenge bit b is 0, then the plaintext of c^* is correctly generated. Thus,$\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ perfectly simulates the game Hyb'. This means that we have

$$\Pr[b' = 0 \mid b = 0] = \Pr[\mathsf{Hyb}'=1] + \frac{1}{2}(1 - \Pr[\mathsf{Hyb}'=1]) = \frac{1}{2} + \frac{1}{2}\gamma.$$

On the other hand, that is, if the challenge bit b is 1, $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$ did not simulate the game correctly. Let δ denote the probability that $m' = m_0$ occurs conditioned on that the challenge bit b is 1. Since m_0 is chosen uniformly at random and $\mathsf{EXT}[\mathcal{A}, \mathsf{PRF}(s,\cdot)]$ knows nothing on m_0 from ek and c^*, we have

$$\delta \le 1/\#\mathcal{M}_{\mathrm{even}}$$

and

$$\Pr[b' = 1 \mid b = 1] = \frac{1}{2}(1 - \delta).$$

Hyb	$\mathsf{F}(m^*, r^*)$	$\mathsf{EXT}[\mathcal{A}, \mathsf{G}](\mathsf{inp})$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$i \leftarrow [q_\mathsf{H}]$
$m^* \leftarrow \mathcal{M}_{\text{even}}$	$c^* := \mathsf{Enc}(ek, m^*; r^*)$	Run $\mathcal{A}^{\mathsf{G}}(\mathsf{inp})$ until i-th query $\lvert\hat{x}\rangle$ to G
$r^* \leftarrow \mathcal{R}$	$\mathsf{inp} := (ek, c^*)$	**if** $i >$ number of queries to G, **return** $\perp$
$c^* := \mathsf{Enc}(ek, m^*; r^*)$	**return** inp	**else return** $x' := \mathsf{Measure}(\lvert\hat{x}\rangle)$
$m' \leftarrow \mathsf{EXT}[\mathcal{A}, \mathsf{G}(\cdot)](ek, c^*)$		
return $\mathsf{bool}(m' \stackrel{?}{=} m^*)$		

Fig. 6. Game Hyb and Algorithms F and EXT

Let us estimate the advantage of $\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}$. From the definition, we have

$$\begin{aligned}\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) &= \lvert 2\Pr[b' = b] - 1\rvert = \lvert \Pr[b' = 0 \mid b = 0] + \Pr[b' = 1 \mid b = 1] - 1\rvert \\ &= \left\lvert \frac{1}{2} + \frac{1}{2}\gamma + \frac{1}{2} - \frac{1}{2}\delta - 1\right\rvert = \frac{1}{2}\lvert\gamma - \delta\rvert .\end{aligned}$$

If $0 \le \gamma < \delta$, then we have the upperbound

$$\Pr[\mathsf{Hyb}' = 1] < \delta \le 1/\#\mathcal{M}_{\text{even}}.$$

On the other hand, that is, if $\gamma \ge \delta$, then we have

$$\Pr[\mathsf{Hyb}' = 1] = \gamma \le 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) + \delta \le 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) + 1/\#\mathcal{M}_{\text{even}}.$$

Thus, in the both cases, we have

$$\Pr[\mathsf{Hyb}' = 1] \le 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) + 1/\#\mathcal{M}_{\text{even}}$$

as we wanted. The running time is about that of $\mathsf{EXT}[\mathcal{A}, \mathsf{PRF}(s, \cdot)]$ and $\mathcal{A}$.

Summary: Summing up the above arguments, we obtain the bound

$$\begin{aligned}\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) &= \lvert \Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_2 = 1]\rvert \\ &\le 2q_\mathsf{G}\sqrt{2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^{\mathsf{Hyb}}_{\mathsf{PRF}}}(\kappa) + 1/\#\mathcal{M}_{\text{even}}} \\ &\quad + \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^1_{\mathsf{PRF}}}(\kappa) + \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^2_{\mathsf{PRF}}}(\kappa)\end{aligned}$$

as we wanted.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek')$	$\overline{\mathsf{Dec}}(dk, c)$, where $dk = (dk', ek', s)$
$(ek', dk') \leftarrow \mathsf{Gen}_1(1^\kappa)$	$m \leftarrow \mathcal{M}$	$m := \mathsf{Dec}_1(dk', c)$
$s \leftarrow \mathcal{S}$	$c := \mathsf{Enc}_1(ek', m)$	**if** $m = \bot$, **return** $K := \mathsf{PRF}(s, c)$
$dk \leftarrow (dk', ek', s)$	$K := \mathsf{H}(m)$	**if** $c \neq \mathsf{Enc}_1(ek', m)$, **return** $K := \mathsf{PRF}(s, c)$
return (ek', dk)	**return** (K, c)	**else return** $K := \mathsf{H}(m)$

Fig. 7. $\mathsf{KEM} := \mathsf{XYZ}[\mathsf{PKE}_1, \mathsf{PRF}, \mathsf{H}]$.

5 Conversion from PR-CPA to IND-CCA

We propose a new conversion XYZ from *deterministic* PKE $\mathsf{PKE}_1 = (\mathsf{Gen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$, whose plaintext and ciphertext spaces are denoted by $\mathcal{M}$ and $\mathcal{C}$, to $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$. We notice that this is a variant of $\mathsf{U}_m^{\not\perp}$ and a KEM variant of the BR93 conversion.

Let $\mathsf{PRF} \colon \mathcal{S} \times \mathcal{C} \to \mathcal{K}$ be a PRF and let $\mathsf{H} \colon \mathcal{M} \to \mathcal{K}$ be a random oracle. We denote $\mathsf{KEM} = \mathsf{XYZ}[\mathsf{PKE}_1, \mathsf{PRF}, \mathsf{H}]$. The algorithms are defined in Figure 7. Assuming PR-CPA security of PKE_1, we have two algorithms $\widetilde{\mathsf{Gen}}_1$ and $\widetilde{\mathsf{Enc}}_1$ that satisfy the conditions in Definition 3.1. Let $\epsilon_{\mathrm{disj}}(\kappa)$ be a disjointness probability of PKE_1 with $\widetilde{\mathsf{Gen}}_1$ and $\widetilde{\mathsf{Enc}}_1$. We additionally require a PRF $\mathsf{PRF}' \colon \mathcal{S} \times \mathcal{M} \to \mathcal{K}$.

Theorem 5.1 (Classical Reduction). *Let PKE_1 be a deterministic PKE scheme. For any IND-CCA adversary $\mathcal{B}$ against KEM, there exist PR-CPA adversaries $\mathcal{A}_{\text{pr-key}}$ and $\mathcal{A}_{\text{pr-cipher}}$ against PKE_1, three adversaries $\mathcal{A}_{\mathsf{PRF}}$ against PRF, such that*

$$\begin{aligned}\mathrm{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{B}}(\kappa) \leq{}& \mathrm{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}_{\text{pr-key}}}(\kappa) + \mathrm{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}_{\text{pr-cipher}}}(\kappa) \\ &+ \mathrm{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\kappa) + \epsilon_{\mathrm{disj}}(\kappa),\end{aligned}$$

and the running times of them are about that of $\mathcal{B}$.

The proof of Theorem 5.1 is in Appendix A.

Theorem 5.2 (Quantum Reduction). *Let PKE_1 be a deterministic PKE scheme. For any IND-CCA quantum adversary $\mathcal{B}$ against KEM, there exist PR-CPA quantum adversaries $\mathcal{A}_{\text{pr-key}}$ and $\mathcal{A}_{\text{pr-cipher}}$ against PKE_1, quantum adversary $\mathcal{A}^0_{\mathsf{PRF}'}$ against PRF', quantum adversary $\mathcal{A}_{\mathsf{PRF}'+\mathsf{PRF}}$ against PRF' and PRF, and two quantum adversaries $\mathcal{A}^3_{\mathsf{PRF}}$ and $\mathcal{A}^6_{\mathsf{PRF}}$ against PRF such that*

$$\begin{aligned}\mathrm{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{B}}(\kappa) \leq{}& \mathrm{Adv}^{\text{pr-key}}_{\mathsf{KEM},\mathcal{A}_{\text{pr-key}}}(\kappa) + \mathrm{Adv}^{\text{pr-cipher}}_{\mathsf{KEM},\mathcal{A}_{\text{pr-cipher}}}(\kappa) \\ &+ \mathrm{Adv}^{\text{prf}}_{\mathsf{PRF}',\mathcal{A}^0_{\mathsf{PRF}'}}(\kappa) + \mathrm{Adv}^{\text{prf}}_{\mathsf{PRF}'+\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}'+\mathsf{PRF}}}(\kappa) \\ &+ \mathrm{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^3_{\mathsf{PRF}}}(\kappa) + \mathrm{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}^6_{\mathsf{PRF}}}(\kappa) + \epsilon_{\mathrm{disj}}(\kappa)\end{aligned}$$

and the running times of them are about that of $\mathcal{B}$.

The proof of Theorem 5.2 follows.

Table 3. Summary of Games for the Security Proof in the QROM

Game	ek	H	c^*	K_0^*	K_1^*	valid c	invalid c	justification
Game_0	ek'	$\mathsf{H}(\cdot)$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{PRF}(s, c)$	
$\mathsf{Game}_{0.5}$	ek'	$\mathsf{PRF}'(s', \cdot)$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{PRF}(s, c)$	PRF security
Game_1	ek'	$\mathsf{H}(\cdot)$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	joint PRF security
Game_2	ek'	$\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Perfect correctness
Game_3	ek'	$\mathsf{PRF}(s, \mathsf{Enc}_1(ek, \cdot))$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{PRF}(s, c^*)$	random	$\mathsf{PRF}(s, c)$	$\mathsf{PRF}(s, c)$	PRF security
Game_4	$\widetilde{ek}'$	$\mathsf{PRF}(s, \mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\mathsf{Enc}_1(\widetilde{ek}', m^*)$	$\mathsf{PRF}(s, c^*)$	random	$\mathsf{PRF}(s, c)$	$\mathsf{PRF}(s, c)$	PR-Key
Game_5	$\widetilde{ek}'$	$\mathsf{PRF}(s, \mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}')$	$\mathsf{PRF}(s, c^*)$	random	$\mathsf{PRF}(s, c)$	$\mathsf{PRF}(s, c)$	PR-Cipher
Game_6	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}')$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	PRF security
Game_7	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}') \setminus \mathsf{Enc}_1(\widetilde{ek}', \cdot)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Statistical Argument
Game_8	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}') \setminus \mathsf{Enc}_1(\widetilde{ek}', \cdot)$	random	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Statistical Argument

5.1 Security Proof in the QROM

We use game-hopping proof. The overview of all games is given in Table 3. In what follows, q_{H} and $q_{\overline{\mathsf{Dec}}}$ are the numbers of queries to the random oracle H and the decapsulation oracle $\overline{\mathsf{Dec}}$ made by $\mathcal{A}$.

Game_0*:* This is the original game, $\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$. Thus, we have

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = |\Pr[\mathsf{Game}_0 = 1] - 1/2| .$$

$\mathsf{Game}_{0.5}$*:* This game is the same as Game_0 except that H is replaced by PRF $\mathsf{PRF}'(s', \cdot) \colon \mathcal{M} \to \mathcal{K}$ with random $s' \leftarrow \mathcal{S}$.

It is straightforward to construct quantum reduction algorithms $\mathcal{A}^0_{\mathsf{PRF}'}$ satisfying

$$|\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_{0.5} = 1]| \leq \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF}',\mathcal{A}^0_{\mathsf{PRF}'}}(\kappa).$$

The running time is about

$$T_{\mathcal{A}^0_{\mathsf{PRF}'}} \approx T_{\mathcal{A}} + T_{\mathsf{Gen}_1} + O\left(q_{\overline{\mathsf{Dec}}} \cdot (T_{\mathsf{Enc}_1} + T_{\mathsf{Dec}_1} + T_{\mathsf{PRF}})\right) .$$

Game_1*:* This game is the same as Game_0 except that the decapsulation oracle employs *another* random oracle $\mathsf{H}_q \colon \mathcal{C} \to \mathcal{K}$ instead of PRF $\mathsf{PRF}(s, \cdot)$ to generate a random key K for invalid ciphertexts.

It is straightforward to construct quantum reduction algorithms $\mathcal{A}_{\mathsf{PRF}'+\mathsf{PRF}}$ satisfying

$$|\Pr[\mathsf{Game}_{0.5} = 1] - \Pr[\mathsf{Game}_1 = 1]| \leq \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF}'+\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}'+\mathsf{PRF}}}(\kappa).$$

The running time is about

$$T_{\mathcal{A}_{\mathsf{PRF}'+\mathsf{PRF}}} \approx T_{\mathcal{A}} + T_{\mathsf{Gen}_1} + O\left(q_{\overline{\mathsf{Dec}}} \cdot (T_{\mathsf{Enc}_1} + T_{\mathsf{Dec}_1})\right) .$$

Game_2: We next define $\mathsf{H}(m) := \mathsf{H}_q(\mathsf{Enc}_1(ek, m))$, where $\mathsf{H}_q : \mathcal{C} \to \mathcal{K}$. Notice that the view $\mathsf{H}(m') = \mathsf{H}_q(c)$ for valid ciphertext c in step 5 of $\overline{\mathsf{Dec}_2}$ is equivalent to the view $\mathsf{H}(m')$ in step 5 of $\overline{\mathsf{Dec}_1}$, because $\mathsf{Enc}_1(ek, \cdot)$ is *perfectly correct* and *injective*. Thus, we have

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

We note that, now, our decapsulation oracle needs not to distinguish valid and invalid ciphertexts: It just rejects if $c = c^*$ and returns $K = \mathsf{H}_q(c)$ otherwise. Thus, in what follows, the reduction algorithm never requires a decapsulation key.

Game_3: We next modify $\mathsf{H}_q : \mathcal{C} \to \mathcal{K}$ with $\mathsf{PRF}(s, \cdot)$. It is straightforward to construct a quantum reduction algorithm $\mathcal{A}^3_{\mathsf{PRF}}$ satisfying

$$|\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_3 = 1]| \le \mathsf{Adv}^{\mathrm{prf}}_{\mathsf{PRF}, \mathcal{A}^3_{\mathsf{PRF}}}(\kappa).$$

The running time is about

$$T_{\mathcal{A}^3_{\mathsf{PRF}}} \approx T_{\mathcal{A}} + T_{\mathsf{Gen}_1} + O\left((q_{\overline{\mathsf{Dec}}} + q_{\mathsf{H}}) \cdot T_{\mathsf{Enc}_1}\right).$$

Game_4: We next replace an encryption key ek' with another one $\widetilde{ek'}$ generated by $\widetilde{\mathsf{Gen}_1}$.

Let us construct a reduction algorithm $\mathcal{A}_{\text{pr-key}}$:

- On input ek, which is ek' or $\widetilde{ek'}$, $\mathcal{A}_{\text{pr-key}}$ chooses $b \leftarrow \{0, 1\}$ and $s \leftarrow \mathcal{S}$. It chooses a message $m^* \leftarrow \mathcal{M}$ uniformly at random and computes $c^* := \mathsf{Enc}_1(ek, m^*)$ and $K^*_0 := \mathsf{PRF}(s, c^*)$. It also chooses $K^*_1 \leftarrow \mathcal{K}$ uniformly at random. It invokes the adversary $\mathcal{A}$ with ek, c^*, and K^*_b.
- $\mathcal{A}_{\text{pr-key}}$ simulates the hash oracle by computing

$$\sum_m |m\rangle\,|y\rangle \mapsto \sum_m |m\rangle\,|\mathsf{PRF}(s, \mathsf{Enc}_1(ek, m)) \oplus y\rangle\,.$$

- $\mathcal{A}_{\text{pr-key}}$ also can simulate the decapsulation oracle: On input $c \neq c^*$, it just returns $K := \mathsf{PRF}(s, c)$.
- Eventually, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}_{\text{pr-key}}$ outputs b'.

It is obvious that $\mathcal{A}_{\text{pr-key}}$ perfectly simulates Game_3 and Game_4 if ek is ek' or $\widetilde{ek'}$. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}, \mathcal{A}_{\text{pr-key}}}(\kappa) &= \left|\Pr[b' = 1 \mid ek = ek'] - \Pr[b' = 1 \mid ek = \widetilde{ek'}]\right| \\ &= |\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]|\,,\end{aligned}$$

as we wanted. The running time is given as

$$T_{\mathcal{A}_{\text{pr-key}}} \approx T_{\mathcal{A}} + O\left((q_{\overline{\mathsf{Dec}}} + q_{\mathsf{H}}) \cdot (T_{\mathsf{Enc}_1} + T_{\mathsf{PRF}})\right).$$

Game_5*:* We next replace a target ciphertext c^* generated by $\mathsf{Enc}_1(\widetilde{ek'}, m^*)$, where $m^* \leftarrow \mathcal{M}$, with another target ciphertext generated by $\widetilde{\mathsf{Enc}_1}(\widetilde{ek'})$.

Let us construct a reduction algorithm $\mathcal{A}_{\text{pr-cipher}}$ as follows:

- On input $\widetilde{ek'}$ and c^*, which is generated by $\mathsf{Enc}_1(\widetilde{ek'}, m^*)$ or $\widetilde{\mathsf{Enc}_1}(\widetilde{ek'})$, $\mathcal{A}_{\text{pr-cipher}}$ chooses $b \leftarrow \{0,1\}$ and $s \leftarrow \mathcal{S}$. It computes $K_0^* := \mathsf{PRF}(s, c^*)$. It also chooses $K_1^* \leftarrow \mathcal{K}$ uniformly at random. It invokes the adversary $\mathcal{A}$ with ek, c^*, and K_b^*.
- $\mathcal{A}_{\text{pr-cipher}}$ simulates the hash oracle by computing

$$\sum_m |m\rangle |y\rangle \mapsto \sum_m |m\rangle |\mathsf{PRF}(s, \mathsf{Enc}_1(ek, m)) \oplus y\rangle .$$

- $\mathcal{A}_{\text{pr-cipher}}$ also can simulate the decapsulation oracle: On input $c \neq c^*$, it just returns $K := \mathsf{PRF}(s, c)$.
- Eventually, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}_{\text{pr-cipher}}$ outputs b'.

It is obvious that $\mathcal{A}_{\text{pr-cipher}}$ perfectly simulates Game_4 and Game_5 depending on c^*. Therefore,

$$\begin{aligned}\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}, \mathcal{A}_{\text{pr-cipher}}}(\kappa) &= \left| \Pr[b' = 1 \mid c^* := \mathsf{Enc}_1(\widetilde{ek'}, m^*)] - \Pr[b' = 1 \mid c^* \leftarrow \widetilde{\mathsf{Enc}_1}(\widetilde{ek'})] \right| \\ &= \left| \Pr[\mathsf{Game}_4 = 1] - \Pr[\mathsf{Game}_5 = 1] \right|,\end{aligned}$$

as we wanted. The running time is given as

$$T_{\mathcal{A}_{\text{pr-key}}} \approx T_{\mathcal{A}} + O\left((q_{\overline{\mathsf{Dec}}} + q_{\mathsf{H}}) \cdot (T_{\mathsf{Enc}_1} + T_{\mathsf{PRF}})\right) .$$

Game_6*:* We replace $\mathsf{PRF}(s, \cdot)$ with $\mathsf{H}_q : \mathcal{C} \to \mathcal{K}$ again. It is straightforward to construct a quantum reduction algorithm $\mathcal{A}^6_{\mathsf{PRF}}$ satisfying

$$\left| \Pr[\mathsf{Game}_5 = 1] - \Pr[\mathsf{Game}_6 = 1] \right| \leq \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF}, \mathcal{A}^6_{\mathsf{PRF}}}(\kappa).$$

The running time is

$$T_{\mathcal{A}^6_{\mathsf{PRF}}} \approx T_{\mathcal{A}} + T_{\widetilde{\mathsf{Gen}_1}} + O\left((q_{\overline{\mathsf{Dec}}} + q_{\mathsf{H}}) \cdot T_{\mathsf{Enc}_1}\right) .$$

Game_7*:* We now turn in the statistical arguments. We employ the (inefficient) challenger that returns false if the challenge ciphertext c^* generated by $\widetilde{\mathsf{Enc}_1}(\widetilde{ek'})$. is in the range of $\mathsf{Enc}_1(\widetilde{ek'}, \mathcal{M})$. Since we have

$$\Pr[c \leftarrow \widetilde{\mathsf{Enc}_1}(\widetilde{ek}) : c \in \mathsf{Enc}(\widetilde{ek}, \mathcal{M})] = \epsilon_{\text{disj}}(\kappa)$$

from the definition of PR-CPA, this modification introduces only negligible difference. We have

$$\left| \Pr[\mathsf{Game}_6 = 1] - \Pr[\mathsf{Game}_7 = 1] \right| \leq \epsilon_{\text{disj}}(\kappa).$$

Game_8: We finally replace $K_0^* := \mathsf{H}_q(c^*)$ with $K_0^* \leftarrow \mathcal{K}$. Apparently, we have

$$\Pr[\mathsf{Game}_8 = 1] = 1/2,$$

because K_0^* and K_1^* are chosen uniformly at random and independent from c^*.

Meanwhile, we notice that the sum of the squared magnitudes of c^* over all queries made to H_q is *zero*, because c^* is outside of the range of $\mathsf{Enc}_1(\widetilde{ek}, \cdot)$ in both of Game_7 and Game_8: If the i-th query is made by the adversary, then the query is to H and cannot contain c^* because of disjointness. If the i-th query is made by the decapsulation oracle, then the query is *classical* and never equals to c^*. Since the sum is zero, $\mathcal{A}$ has no knowledge on $\mathsf{H}_q(c^*)$ and the views in Game_7 and Game_8 are equivalent. [1] Hence,

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

This completes the proof.

6 Implementation

We report the implementation results on a desktop PC and on a RasPi, based on the previous implementation of a variant of NTRU [HRSS17].

6.1 NTRU-HRSS

We review a variant of NTRU, which we call $\mathsf{NTRU}_{\mathsf{HRSS17}}$, in [HRSS17].

Let $\Phi_m(x) \in \mathbb{Z}[x]$ be the m-th cyclotomic polynomial. We have $\Phi_1 = x - 1$. If m is prime, then we have $\Phi_m = 1 + x + \cdots + x^{m-1}$. Define $S_n := \mathbb{Z}[x]/(\Phi_n)$ and $R_n := \mathbb{Z}[x]/(x^n - 1)$. For prime n, we have $x^n - 1 = \Phi_1\Phi_n$ and $R_n \simeq S_1 \times S_n$. We define $\mathrm{Lift}_p \colon S_n/(p) \to R_n$ as

$$\mathrm{Lift}_p(v) := \left[\Phi_1[v/\Phi_1]_{(p,\Phi_n)}\right]_{(x^n-1)}.$$

By definition, we have $\mathrm{Lift}_p(v) \equiv 0 \pmod{\Phi_1}$ and $\mathrm{Lift}_p(v) \equiv v \pmod{(p, \Phi_n)}$. Let $\mathfrak{p} = (p, \Phi_n)$ and $\mathfrak{q} = (q, x^n - 1)$. Let

$$\begin{aligned}
\mathcal{T} &:= \{a \in \mathbb{Z}[x] : a = [a]_{\mathfrak{p}}\} = \{a \in \mathbb{Z}[x] : a_i \in (p) \text{ and } \deg(a) < \deg(\Phi_n)\} \\
\mathcal{T}_+ &:= \{a \in \mathcal{T} : \langle xa, a\rangle \geq 0\}.
\end{aligned}$$

The definition of $\mathsf{NTRU}_{\mathsf{HRSS17}}$ is in Figure 8. Notice that all ciphertexts are equivalent to 0 modulo (q, Φ_1). which prevents a trivial distinguishing attack.

Hülsing et al. chooses $(n, p, q) = (701, 3, 8192)$: The scheme is perfectly correct and they claimed 128-bit post-quantum security of this parameter set. The implementation of $\mathsf{NTRU}_{\mathsf{HRSS17}}$ and $\mathsf{QFO}^{\perp}[\mathsf{NTRU}_{\mathsf{HRSS17}}, \mathsf{G}, \mathsf{H}, \mathsf{H}']$ is reported in [HRSS17].

[1] The reader can invoke the algorithmic OW2H lemma (Lemma 2.1) to show this equivalence. In the hybrid game, the extractor EXT will output the result c of the measure on the i-th query $|\hat{x}\rangle$ to H_q. Notice that any query $|\hat{x}\rangle$ cannot set the *non-zero* amplitude for the state $|c^*\rangle$ as we already discussed. Thus, any query cannot contain c^* and $\Pr[c = c^*]$ is zero.

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Enc}(h, m),\ m \in \mathcal{T}$	$\mathsf{Dec}(f, c)$
$g, f \leftarrow \mathcal{T}_+$	$r \leftarrow \mathcal{T}$	$m' := \left[[cf]_{\mathfrak{q}} f^{-1}\right]_{\mathfrak{p}}$
$f_q := [1/f]_{(q,\Phi_n)}$	$c := [prh + \mathrm{Lift}_p(m)]_{\mathfrak{q}}$	**return** m'
$h := [\Phi_1 g f_q]_{\mathfrak{q}}$	**return** c	
return $dk = f, ek = h$		

Fig. 8. $\mathsf{NTRU}_{\mathsf{HRSS17}}$

$\mathsf{Gen}'(1^\kappa) = \mathsf{Gen}$	$\mathsf{Enc}'(h, (m, r)),\ (m, r) \in \mathcal{T}^2$	$\mathsf{Dec}'(f, c)$
$g, f \leftarrow \mathcal{T}_+$	$c := [prh + \mathrm{Lift}_p(m)]_{\mathfrak{q}}$	$m' := \left[[cf]_{\mathfrak{q}} f^{-1}\right]_{\mathfrak{p}}$
$f_q := [1/f]_{(q,\Phi_n)}$	**return** c	$r' := \left[\left[(c - \mathrm{Lift}_p(m')) \cdot (ph)^{-1}\right]_{\mathfrak{q}}\right]_{\mathfrak{p}}$
$h := [\Phi_1 g f_q]_{\mathfrak{q}}$		**return** (m', r')
return $dk = f, ek = h$		

Fig. 9. Our Modification $\mathsf{NTRU}_{\mathsf{HRSS17}}'$

Our Modification: We want PKE_1 to be *deterministic* PKE. Hence, we consider a pair of (m, r) as a plaintext and make the decryption algorithm output (m, r) rather than m. The modification $\mathsf{NTRU}_{\mathsf{HRSS17}}'$ is summarized in Figure 9.

We also implement $\mathsf{XYZ}[\mathsf{NTRU}_{\mathsf{HRSS17}}', \mathsf{H}]$, where H is implemented by SHAKE. In order to avoid the inversion of polynomials in decapsulation, we add f^{-1} modulo $\mathfrak{p}$ to dk as [HRSS17] did. This requires extra 139 bytes. In addition, we put $(ph)^{-1}$ modulo $\mathfrak{q}$ in dk, which requires extra 1140 bytes. Thus, our decapsulation key is of length 2557 bytes.

6.2 Experimental Results

We preform the experiment with

- one core of an Intel Core i7-6700 at 3.40GHz on a desktop machine with 8GB memory and Ubuntu16.04 and
- a RasPi3 with 32-bit Rasbian.

We use gcc to compile the programs with option -O3. The experimental results are summarized in Table 4. The Basic and CCA KEM implies $\mathsf{NTRU}_{\mathsf{HRSS17}}'$ and $\mathsf{XYZ}[\mathsf{NTRU}_{\mathsf{HRSS17}}']$. The results reflect that our conversion adds only small extra amount of costs for hashing in encryption and adds about T_{Enc_1} for re-encrypting in decryption.

We note that our implementations are for reference and we did not optimize them. Further optimizations will speed up the algorithms as [HRSS17] did.

Table 4. Our Experiments: We have $|ek| = 1140$ bytes, $|dk| = 2557$ bytes, and $|c| = 1140$ bytes.

Basic KEM on a PC	(milliseconds)		Basic KEM on a RasPi3	(milliseconds)	
	Gen_1	1888		Gen_1	34048
	Enc_1	328		Enc_1	3097
	Dec_1	958		Dec_1	17717
CCA KEM on a PC	**(milliseconds)**		**CCA KEM on a RasPi3**	**(milliseconds)**	
	$\overline{\mathsf{Gen}}$	2562		$\overline{\mathsf{Gen}}$	58497
	$\overline{\mathsf{Enc}}$	333		$\overline{\mathsf{Enc}}$	3208
	$\overline{\mathsf{Dec}}$	1284		$\overline{\mathsf{Dec}}$	11843

References

BCLvVxx. Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU prime. In Nnnn Hmm, editor, *SAC 2017*, volume xxxxx of *LNCS*, pages xxx–xxx. Springer, Heidelberg, 20xx. version, 20170817:160919. See also https://eprint.iacr.org/2016/461. 4

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, 2011. 3

BDK+17. Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, and Damien Stehlé. CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. 2017. See also https://eprint.iacr.org/2017/634. 4

BGG+17. Paulo S. L. M. Barreto, Shay Gueron, Tim Gueneysu, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, and Jean-Pierre Tillich. CAKE: Code-based algorithm for key encapsulation. 2017. See also https://eprint.iacr.org/2017/757. 4

BR93. Mihir Bellare and Phillip Rogaway. Random oracle are practical: A paradigm for designing efficient protocols. In *CCS '93*, pages 62–73. ACM, 1993. 2

BR95. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT '94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, 1995. 2

CHJ+02. Jean-Sébastien Coron, Helena Handschuh, Marc Joye, Pascal Paillier, David Pointcheval, and Christophe Tymen. GEM: A Generic chosen-ciphertext secure Encryption Method. In Bart Preneel, editor, *CT-RSA 2002*, volume 2271 of *LNCS*, pages 175–184. Springer, Heidelberg, 2002. 2

CHK+16. Jung Hee Cheon, Kyoo Hyung Han, Jinsu Kim, Changmin Lee, and Yongha Son. A practical post-quantum public-key cryptosystem based on spLWE. In Seokhie Hong and Jong Hwan Park, editors, *ICISC 2016*, volume 10157 of *LNCS*, pages 51–74. Springer, Heidelberg, 2016. See also https://eprint.iacr.org/2016/1055. 4

CKLS16. Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yong Soo Song. Lizard: Cut off the tail! practical post-quantum public-key encryption from LWE and LWR. 2016. See also https://eprint.iacr.org/2016/1126. 4

Den03. Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *IMA 2003*, volume 2898 of *LNCS*, pages 133–151. Springer, Heidelberg, 2003. 2

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO '99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, 1999. 2, 3

FO00. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 83(1):24–32, 2000. A preliminary version appeared in *PKC '99*, 1999. 2

FO13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptology*, 26(1):80–101, 2013. 2, 3

FOPS04. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA-OAEP is secure under the RSA assumption. *Journal of Cryptology*, 17(2):81–104, 2004. 2

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *STOC 2008*, pages 197–206. ACM, 2008. see also https://eprint.iacr.org/2007/432. 4

Ham17. Mike Hamburg. Module-LWE: The three bears. 2017. ver. Draft 7. See also https://www.shiftleft.org/papers/threebears/. 4

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017*, volume 1xxxx of *LNCS*, pages xxx–xxx. Springer, Heidelberg, 2017. version, 20170808:094949. See also https://eprint.iacr.org/2017/604. 2, 3, 4, 5, 6, 11

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *ANTS-III*, volume 1423 of *LNCS*, pages 267–288. Springer, Heidelberg, 1998. 4

HRSS17. Andreas Hülsing, Joost Rijneveld, John Schanck, and Peter Schwabe. High-speed key encapsulation from NTRU. 2017. To appear in *CHES 2017*. See also https://eprint.iacr.org/2017/667. 4, 5, 20, 21

McE78. Robert J. McEliece. A public key cryptosystem based on algebraic coding theory. Technical report, DSN progress report, 1978. 4

Men12. Alfred Menezes. Another look at provable security. Invited Talk at EUROCRYPT 2012, 2012. 1

NC00. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000. 6

Nie86. Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15:159–166, 1986. 4

OP01. Tatsuaki Okamoto and David Pointcheval. REACT: Rapid enhanced-security asymmetric cryptosystem transform. In David Naccache, editor, *CT-RSA 2001*, volume 2020 of *LNCS*, pages 159–175. Springer, Heidelberg, 2001. 2

SS11. Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, 2011. 4

TU16. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, 2016. See also https://eprint.iacr.org/2015/1210. 3, 11

Unr15. Dominique Unruh. Revocable quantum timed-release encryption. *Journal of the ACM*, 62(6):No.49, 2015. The preliminary version appeared in *EUROCRYPT 2014*. See also https://eprint.iacr.org/2013/606. 6

Wan17. Yongge Wang. Revised quantum resistant public key encryption scheme RLCE and IND-CCA2 security for McEliece schemes. 2017. See also https://eprint.iacr.org/2017/206. 4

Zha12. Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, 2012. 3, 5

Table 5. Summary of Games for the Security Proof in the ROM

Game	ek	H	c^*	K_0^*	K_1^*	valid c	invalid c	justification
Game_0	ek'	real	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{PRF}(s, c)$	
Game_1	ek'	real	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}(m^*)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	PRF security
Game_2	ek'	$\mathsf{H}_q(\mathsf{Enc}_1(ek, \cdot))$	$\mathsf{Enc}_1(ek', m^*)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Perfect correctness
Game_3	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\mathsf{Enc}_1(\widetilde{ek}', m^*)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	–	PR-Key
Game_4	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}')$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	–	PR-Cipher
Game_5	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}') \setminus \mathsf{Enc}_1(\widetilde{ek}', \cdot)$	$\mathsf{H}_q(c^*)$	random	$\mathsf{H}_q(c)$	–	Statistical Argument
Game_6	$\widetilde{ek}'$	$\mathsf{H}_q(\mathsf{Enc}_1(\widetilde{ek}', \cdot))$	$\widetilde{\mathsf{Enc}_1}(\widetilde{ek}') \setminus \mathsf{Enc}_1(\widetilde{ek}', \cdot)$	random	random	$\mathsf{H}_q(c)$	–	Statistical Argument

A Warm Up: Proof of Theorem 5.1

The overview of all games is given in Table 5. Here, we give a sketch of proof.

- Game_0: This is the original game $\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$. We have

$$\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = |\Pr[\mathsf{Game}_0 = 1] - 1/2| .$$

- Game_1: We replace $\mathsf{PRF}(s, \cdot)$ with a random oracle $\mathsf{H}_q : C \to \mathcal{K}$. It is easy to show that there exists an adversary $\mathcal{A}_{\mathsf{PRF}}$ satisfying

$$|\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_1 = 1]| \leq \mathsf{Adv}^{\text{prf}}_{\mathsf{PRF},\mathcal{A}_{\mathsf{PRF}}}(\kappa)$$

  and its running time is about that of $\mathcal{A}$.
- Game_2: We next set $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$ as $\mathsf{H}(m) := \mathsf{H}_q(\mathsf{Enc}_1(ek', m))$ (instead of $\mathsf{H} \leftarrow \mathsf{Map}(\mathcal{M}, \mathcal{K})$). We notice that the two games are equivalent, since PKE_1 is *perfectly correct*. Thus, we have that

$$\mathsf{Game}_1 = \mathsf{Game}_2.$$

  Notice that now, we can simplify the decapsulation oracle as; output $K = \mathsf{H}_q(c)$ if $c \neq c^*$ and $\perp$ if $c = c^*$. So that, the decapsulation oracle can forget the decapsulation key in what follows.
- Game_3: We next replace ek' with $\widetilde{ek}'$ generated by $\widetilde{\mathsf{Enc}}_1$. It is straightforward to show that there exists an adversary $\mathcal{A}_{\text{pr-key}}$ satisfying

$$|\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_3 = 1]| \leq \mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}_{\text{pr-key}}}(\kappa)$$

  and its running time is about that of $\mathcal{A}$.
- Game_4: We next replace $c^* := \mathsf{Enc}_1(\widetilde{ek}', m^*)$ with $m^* \leftarrow \mathcal{M}$ with $c^* \leftarrow \widetilde{\mathsf{Enc}}_1(\widetilde{ek}')$. It is straightforward to show that there exists an adversary $\mathcal{A}_{\text{pr-cipher}}$ satisfying

$$|\Pr[\mathsf{Game}_3 = 1] - \Pr[\mathsf{Game}_4 = 1]| \leq \mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}_{\text{pr-cipher}}}(\kappa)$$

  and its running time is about that of $\mathcal{A}$.

Table 6. Summary of Games for the Security Proof in the ROM

Game	m^*	r^*	c^*	justification
Game_0	$\mathcal{M}_{\mathrm{even}}$	$\mathsf{G}(m^*)$	$\mathsf{Enc}(ek, m^*; r^*) = \mathsf{Enc}_1^{\mathsf{G}}(ek, m^*)$	
Game_1	$\mathcal{M}_{\mathrm{even}}$	r^*	$\mathsf{Enc}(ek, m^*; r^*)$	IND-CPA security of PKE
Game_2	$\mathcal{M}_{\mathrm{odd}}$	r^*	$\mathsf{Enc}(ek, m^*; r^*) = \widetilde{\mathsf{Enc}_1}(ek)$	IND-CPA security of PKE

– Game_5: We now employ statistical arguments: We replace the challenge ciphertext $c^* \leftarrow \widetilde{\mathsf{Enc}_1}(\widetilde{ek'})$ with $c^* \leftarrow \widetilde{\mathsf{Enc}_1}(\widetilde{ek'}) \setminus \mathsf{Enc}_1(\widetilde{ek'}, \mathcal{M})$;
That is, if c^* is within the range of $\mathsf{Enc}_1(\widetilde{ek'}, \mathcal{M})$, we abort the game. We have

$$|\Pr[\mathsf{Game}_4 = 1] - \Pr[\mathsf{Game}_5 = 1]| \leq \epsilon_{\mathrm{disj}}(\kappa).$$

– Game_6: We finally replace $K_0^* := \mathsf{H}_q(c^*)$ with random. Since the adversary $\mathcal{A}$ cannot ask c^* to H_q, it cannot distinguish Game_6 with Game_5. Moreover, the adversary $\mathcal{A}$ cannot distinguish two random keys in Game_6. Thus, we have

$$\Pr[\mathsf{Game}_5 = 1] = \Pr[\mathsf{Game}_6 = 1] = 1/2.$$

B Warm Up: Proof of Theorem 4.1

It is obvious that $\mathsf{Adv}^{\text{pr-key}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = 0$, since $\mathsf{Gen}_1 = \widetilde{\mathsf{Gen}_1}$. It is also obvious that the output of $\widetilde{\mathsf{Enc}_1}(ek)$ never overlaps with $\mathsf{Enc}_1(ek, \mathcal{M}_{\mathrm{even}}) \subseteq \mathsf{Enc}(ek, \mathcal{M}_{\mathrm{even}}; \mathcal{R})$, because PKE is perfectly correct and the range of $\widetilde{\mathsf{Enc}_1}(ek)$ is $\mathsf{Enc}(ek, \mathcal{M}_{\mathrm{odd}}; \mathcal{R})$.

In the rest of this section, we give a tight *classical* security proof for pseudorandomness of ciphertexts. The overview of all games is given in Table 6

What we want to show is the upper bound of

$$\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1,\mathcal{A}}(\kappa) = |\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_2 = 1]|$$

is negligible in κ.

Game_0*:* We expand algorithms and obtain Game_0:

$$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\mathrm{even}}; r^* \leftarrow \mathsf{G}(m^*); c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'.$$

Game_1*:* This game is the same as Game_0 except that the randomness of the challenge ciphertext is freshly generated:

$$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\mathrm{even}}; r^* \leftarrow \mathcal{R}; c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'.$$

In addition, we change the random oracle G as follows: on query $m \in \mathcal{M}$,

1. If (m, r) is stored in the table G, then return r
2. If $m = m^*$, then abort the game.

3. Otherwise, return $r \leftarrow \mathcal{R}$ and store (m, r) to the table G.

Let Bad denote the event that the challenger aborts the game in the simulation of G. Since the two games are equivalent until Bad occurs, we have

$$|\Pr[\mathrm{Game}_0 = 1] - \Pr[\mathrm{Game}_1 = 1]| \leq \Pr[\mathrm{Bad}].$$

Let $\gamma = \Pr[\mathrm{Bad}]$.

We can construct a reduction algorithm $\mathcal{A}_{\mathsf{PKE}}$ against IND-CPA security of PKE as follows:

- On input ek, $\mathcal{A}_{\mathsf{PKE}}$ chooses two messages $m_0 \leftarrow \mathcal{M}_{\mathrm{even}}$ and $m_1 \leftarrow \mathcal{M}_{\mathrm{odd}}$ uniformly at random. It then queries them to its challenge oracle and obtains $c^* \leftarrow \mathsf{Enc}(ek, m^*; r^*)$, where m^* is m_b. It initialize the table G and invokes $\mathcal{A}$ with ek and c^*.
- $\mathcal{A}_{\mathsf{PKE}}$ simulates the random oracle G as follows:
  1. If (m, r) is stored in the table G, then return r
  2. If $m = m_0$, then output $b' = 0$ and terminate the game.
  3. Otherwise, return $r \leftarrow \mathcal{R}$ and store (m, r) to the table G.
- Eventually, $\mathcal{A}$ outputs a bit. $\mathcal{A}_{\mathsf{PKE}}$ outputs $b' \leftarrow \{0, 1\}$.

If the challenge bit b is 0, then the plaintext of c^* is correctly generated. Thus, $\mathcal{A}_{\mathsf{PKE}}$ correctly simulates the two games until Bad occurs. This means that we have

$$\Pr[b' = 0 \mid b = 0] = \Pr[\mathrm{Bad} \mid b = 0] + \frac{1}{2}(1 - \Pr[\mathrm{Bad} \mid b = 0]) = \frac{1}{2} + \frac{1}{2}\gamma.$$

On the other hand, that is, if the challenge bit b is 1, $\mathcal{A}_{\mathsf{PKE}}$ did not simulate the game correctly. However, notice that $\mathcal{A}$ knows nothing on m_0 through ek and c^*. Thus, it is hard for $\mathcal{A}$ to make Bad occurs. Let δ denote the probability that Bad occurs conditioned on that the challenge bit b is 1. Since m_0 is chosen uniformly at random, we have

$$\delta \leq q_{\mathrm{G}}/\#\mathcal{M}_{\mathrm{even}}$$

and

$$\Pr[b' = 1 \mid b = 1] = \frac{1}{2}(1 - \Pr[\mathrm{Bad} \mid b = 1]) = \frac{1}{2} - \frac{1}{2}\delta.$$

Let us estimate the advantage of $\mathcal{A}_{\mathsf{PKE}}$. From the definition, we have

$$\begin{aligned}\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa) &= |2\Pr[b' = b] - 1| = |\Pr[b' = 0 \mid b = 0] + \Pr[b' = 1 \mid b = 1] - 1| \\ &= \left|\frac{1}{2} + \frac{1}{2}\gamma + \frac{1}{2} - \frac{1}{2}\delta - 1\right| = \frac{1}{2}\left|\gamma - \delta\right|.\end{aligned}$$

If $0 \leq \gamma < \delta$, then we have the upperbound

$$\Pr[\mathrm{Bad}] < \delta \leq q_{\mathrm{G}}/\#\mathcal{M}_{\mathrm{even}}.$$

On the other hand, that is, if $\gamma \geq \delta$, then we have

$$\Pr[\mathrm{Bad}] = \gamma \leq 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa) + \delta \leq 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa) + q_{\mathrm{G}}/\#\mathcal{M}_{\mathrm{even}}.$$

Thus, in the both cases, we have

$$\Pr[\mathrm{Bad}] \leq 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa) + q_{\mathrm{G}}/\#\mathcal{M}_{\mathrm{even}}$$

as we wanted.

Game_2*:* This game is the same as Game_1 except that the challenge ciphertext is generated by $\mathsf{Enc}(ek, m^*; r^*)$, where $m^* \leftarrow \mathcal{M}_{\mathrm{odd}}$ rather than $m^* \leftarrow \mathcal{M}_{\mathrm{even}}$:

$$ek \leftarrow \mathsf{Gen}(1^\kappa); m^* \leftarrow \mathcal{M}_{\mathrm{odd}}; r^* \leftarrow \mathcal{R}; c^* := \mathsf{Enc}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'.$$

Let us construct a reduction algorithm $\mathcal{A}'_{\mathsf{PKE}}$ against IND-CPA security of PKE as follows:

- On input ek, $\mathcal{A}'_{\mathsf{PKE}}$ chooses two messages $m_0 \leftarrow \mathcal{M}_{\mathrm{even}}$ and $m_1 \leftarrow \mathcal{M}_{\mathrm{odd}}$ uniformly at random. It then queries them to its challenge oracle and obtains $c^* \leftarrow \mathsf{Enc}(ek, m^*; r^*)$, where m^* is m_b. It initializes the table G and invokes $\mathcal{A}$ with ek and c^*.
- $\mathcal{A}'_{\mathsf{PKE}}$ simulates the random oracle G as follows:
  1. If (m, r) is stored in the table G, then return r
  2. Otherwise, return $r \leftarrow \mathcal{R}$ and store (m, r) to the table G.
- Eventually, $\mathcal{A}$ outputs a bit b'. $\mathcal{A}'_{\mathsf{PKE}}$ outputs b'.

It is obvious that $\mathcal{A}'_{\mathsf{PKE}}$ perfectly simulates Game_{b+1} depending on the challenge bit $b \in \{0, 1\}$. Therefore,

$$\begin{aligned}
\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}'_{\mathsf{PKE}}}(\kappa) &= |\Pr[b' = b] - 1/2| \\
&= |(1 - \Pr[b' = 1 \mid b = 0]) + \Pr[b' = 1 \mid b = 1] - 1| \\
&= |1 - \Pr[\mathsf{Game}_1 = 1] + \Pr[\mathsf{Game}_2 = 1] - 1| \\
&= |\Pr[\mathsf{Game}_2 = 1] - \Pr[\mathsf{Game}_1 = 1]|,
\end{aligned}$$

this results in

$$|\Pr[\mathsf{Game}_1 = 1] - \Pr[\mathsf{Game}_2 = 1]| = \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}'_{\mathsf{PKE}}}(\kappa).$$

Summary: Summing up the differences, we obtain the bound

$$\begin{aligned}
\mathsf{Adv}^{\text{pr-cipher}}_{\mathsf{PKE}_1, \mathcal{A}}(\kappa) &= |\Pr[\mathsf{Game}_0 = 1] - \Pr[\mathsf{Game}_2 = 1]| \\
&\leq 2\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}_{\mathsf{PKE}}}(\kappa) + \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}, \mathcal{A}'_{\mathsf{PKE}}}(\kappa) + \frac{q_{\mathsf{G}}}{\#\mathcal{M}_{\mathrm{even}}}
\end{aligned}$$

as we wanted.