Anonymity of NIST PQC Round-3 KEMs

Keita Xagawa[1]

NTT Social Informatics Laboratories, keita.xagawa.zv@hco.ntt.co.jp

Abstract. This paper investigates *anonymity* of all NIST PQC Round 3 KEMs: Classic McEliece, Kyber, NTRU, Saber, BIKE, FrodoKEM, HQC, NTRU Prime (Streamlined NTRU Prime and NTRU LPRime), and SIKE. We show the following results:

- NTRU is anonymous in the quantum random oracle model (QROM) if the underlying deterministic PKE is strongly disjoint-simulatable. NTRU is collision-free in the QROM. A hybrid PKE scheme constructed from NTRU as KEM and appropriate DEM is anonymous and robust. Similar results hold for BIKE, FrodoKEM, HQC, NTRU LPRime, and SIKE.
- Classic McEliece is anonymous in the QROM if the underlying PKE is strongly disjoint-simulatable and a hybrid PKE scheme constructed from it as KEM and appropriate DEM is anonymous.
- Grubbs, Maram, and Paterson pointed out that Kyber and Saber has a gap in the current IND-CCA security proof in the QROM (Cryptography ePrint Archive 2021/708). We found that Streamlined NTRU Prime has another technical obstacle for the IND-CCA security proof in the QROM.

Those answer the open problem to investigate the anonymity and robustness of NIST PQC Round 3 KEMs posed by Grubbs, Maram, and Paterson (Cryptography ePrint Archive 2021/708).

We use strong disjoint-simulatability of the underlying PKE of KEM and strong pseudorandomness and smoothness of KEMs, which will be of independent interest.

Keywords: anonymity, robustness, post-quantum cryptography, NIST PQC standardization, KEM, PKE

Table of Contents

1 Introduction

Public-Key Encryption (PKE) allows us to send a message confidentially to a receiver if the receiver's public key is available. However, a ciphertext may reveal the receiver's public key. Roughly speaking, PKE is *anonymous* [BBDP01] if a ciphertext hides the receiver's information. Anonymous primitive is often used in the context of privacy-enhancing technologies.

If we use anonymous PKE, then a ciphertext indicates (computationally) no information of a receiver. Thus, once the receiver receives a ciphertext, it should decrypt it and check the message. In this situation, a ciphertext maybe has two (or more) recipients. Intuitively speaking, PKE is *robust* [ABN10] if only the intended receiver can obtain a meaningful plaintext from a ciphertext.

Both anonymity and robustness are important and useful properties beyond the standard IND-CCA security. Anonymous PKE is an important building primitive for anonymous credential systems [CL01], auction protocols [Sak00], (weakly) anonymous AKE [BCGNP09, FSXY13, FSXY15, SSW20], and so on. Robust PKE has an application for searchable encryption [ABC+05] and auction [Sak00].

Previous works: Mohassel [Moh10] studied anonymity and robustness of a special KEM/DEM framework, where KEM is implemented by PKE with random plaintext. He observed that even if anonymous KEM and DEM sometimes fail to lead to an anonyous hybrid PKE.

Grubbs, Maram, and Paterson [GMP21a] discussed anonymity and robustness of post-quantum KEM schemes and KEM/DEM framework in the quantum random oracle model (QROM). They also studied anonymity and robustness of the hybrid PKE based on KEM with implicit rejection. On the variants of the Fujisaki-Okamoto transformation [FO99, FO13], they showed that anonymity and collision-freeness of KEMs obtained by $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\not\perp\prime\prime}$[1] and they lead to anonymous, robust hybrid PKEs from appropriate assumptions. They also show anonymity and robustness of KEM obtained by $\mathsf{HFO}^{\perp\prime}$[2] and it lead to anonymous, robust hybrid PKE form appropriate assumptions. They then examined NIST PQC Standardization finalists (Classic McEliece [ABC+20], Kyber [SAB+20], NTRU [CDH+20], and Saber [DKR+20]). They showed the following results:

- Classic McEliece: They found that Classic McEliece is not collision-free. Since their anonymity proof in [GMP21a, Theorem 5] strongly depends on the collision-freeness of the underlying PKE, we cannot apply their anonymity proof to Classic McEliece. They also show that the hybrid PKE fails to achieve robustness since Classic McEliece is not collision-free.
- Kyber: They found that Kyber's anonymity (and even IND-CCA security) has two technical obstacles ('pre-key' and 'nested random oracles') in the QROM.
- NTRU: NTRU's anonymity has another technical obstacle: A key is computed as $\mathsf{H}(\mu)$ instead of $\mathsf{H}(\mu, c)$, where μ is a plaintext and c is a ciphertext. The robustness of the hybrid PKE with NTRU is unclear.
- Saber: They insisted they show Saber's anonymity and IND-CCA security and the robustness of the hybrid PKE with Saber in the QROM. Unfortunately, Saber in [DKR+20] also uses both 'pre-key' and 'nested random oracles' as Kyber and their proofs cannot be applied to Saber. See their slides [GMP21b].

Grubbs et al. left several open problems: One of them is the anonymity and robustness of NTRU; the other important one is the anonymity of Classic McEliece.

Summarizing above, unfortunately, we do not know whether all four finalists are anonymous or not, although the much effort of Grubbs et al. and their clean and modular framework.

1.1 Our Contribution

Anonymity through pseudorandomness and smoothness: Our starting point is strong pseudorandomness instead of anonymity: We say PKE/KEM/DEM is strongly pseudorandom if its ciphertext is indistinguishable from a random string chosen by a simulator on input the security parameter. [3] It is easy to show strong pseudorandomness implies anonymity.

Using this notion, we attempt to follow the IND-CCA security proof of the KEM/DEM framework [CS02], that is, we try to show that the hybrid PKE from strongly pseudorandom KEM/DEM is also strongly pseudorandom, which implies that the hybrid PKE is anonymous. If we directly try to prove the ANON-CCA security of the hybrid PKE, then we will need to simulate *two* decryption oracles. Considering pseudorandomness allows us to treat a *single* key and oracle and simplifies the security proof. Unfortunately, we face another obstacle in the security proof when we consider pseudorandomness.

To resolve the obstacle, we define *sparseness* of KEM with explicit rejection and *smoothness* of KEM with implicit rejection: We say KEM with explicit rejection is sparse if a ciphertext c chosen by a simulator is decapsulated into $\perp$ with overwhelming probability. We say KEM with implicit rejection is smooth if, given a ciphertext c chosen by a simulator, any efficient adversary cannot distinguish a random key from a decapsulated key. This definition imitates the smoothness of hash proof system [CS02]. Those notions help us to prove the pseudorandomness of the hybrid PKE. We believe that sparseness and smoothness will play important role in another place.

[1] A variant of $\mathsf{FO}^{\not\perp}$ using 'pre-key' technique. They wrote "a variant of the $\mathsf{FO}^{\not\perp}$ transform" in their paper.
[2] They modify 'key-confirmation hash' $c_1 = \mathsf{F}(\mu)$ of $\mathsf{HFO}^{\perp}$ with $c_1 = \mathsf{F}(\mu, c_0)$, where $c_0 = \mathsf{Enc}(ek, \mu)$.
[3] if the simulator can depend on an encryption key, then we just say pseudorandom.

Pseudorandomness, smoothness, and collision-freeness of the FO variants: In order to treat the case for Classic McEliece and NTRU, in which the underlying PKE is deterministic, we treat SXY [SXY18], variants of U [HHK17], and variants of HU [JZM19]. Modifying the security proofs of them, we show that the obtained KEM is strongly pseudorandom and smooth if the underlying PKE is strongly disjoint-simulatable [SXY18]. We also show that the obtained KEM is collision-free if the underlying deterministic PKE (DPKE) is collision-free. We finally note that our reductions enjoy *tightness*.
Grubbs et al. [GMP21a] discussed a barrier to show anonymity of NTRU, which stems from the design choice $K = \mathsf{H}(\mu)$ instead of $K = \mathsf{H}(\mu, c)$. The former choice makes their simulation difficult. In addition, their proof technique requires the underlying PKE to be collision-free. Since the underlying PKE of Classic McEliece lacks collision freeness, they left the proof of anonymity of Classic McEliece as an open problem. Both barriers stem from the fact that we need to simulate *two* decapsulation oracles in the proof of ANON-CCA-security. We avoid those technical barriers by using a stronger notion, SPR-CCA security; in the proof of SPR-CCA-security, we only need to simulate a *single* decapsulation oracle.

Application to NIST PQC Round-3 KEMs: By using the above techniques, we solve open problems posed by Grubbs et al. and extend the study of finalists and alternative candidates of NIST PQC Round 3 KEMs.
We found the following properties (we omit the detail of the assumptions):
- Classic McEliece is anonymous, but not collision-free. The hybrid PKE is anonymous.
- NTRU is anonymous and collision-free. The hybrid PKE is anonymous and robust. Similar results hold for BIKE, HQC (HQC-128 and HQC-196)[4], NTRU LPRime, and SIKE.
- FrodoKEM uses $\mathsf{FO}^{\not\perp\prime}$. We can use the results of Grubbs et al.and FrodoKEM is anonymous and collision-free. The hybrid PKE is anonymous and robust.
- Grubbs et al. reported that Kyber and Saber have similar obstacles for anonymity (and IND-CCA security) [GMP21a, GMP21b]: The one is a 'pre-key' problem: the encapsulation algorithm first computes $(k, r) \leftarrow \mathsf{G}(\mu, \mathsf{Hash}(ek))$ from a message μ and computes $K = \mathsf{H}(k, \mathsf{Hash}(c))$ where c is a ciphertext. The other is a 'nested random oracles' problem: the key is computed as $K = \mathsf{H}(k, \mathsf{Hash}(c))$ instead of $\mathsf{H}(k, c)$.
- We found that Streamlined NTRU Prime has another technical obstacle for anonymity: the key and key-confirmation value is computed as $K = \mathsf{H}(k, c_0, c_1)$ and $h = \mathsf{F}(k, \mathsf{Hash}(ek))$, where $k = \mathsf{H}_3(\mu)$. While this is not a big problem for the IND-CCA security in the ROM, we fail to show the IND-CCA security in the QROM. See section R for the discussion.

See the summary in Table 1.

Remark 1.1. Bernstein [Ber21] suggests to use the domain extension of quantum random oracles in [Zha19, Section 5], which is shown *quantumly indifferentiable*. Let $C^{\mathsf{H}_1,\mathsf{H}_2}(x, y) = \mathsf{H}_1(\mathsf{H}_2(x), y)$. Roughly speaking, we say $C^{\mathsf{H}_1,\mathsf{H}_2}$ is indifferentiable if any efficient adversary cannot distinguish oracles $\mathsf{H}_1, \mathsf{H}_2, C^{\mathsf{H}_1,\mathsf{H}_2}$ from $\mathsf{Sim}^{\mathsf{H}}, \mathsf{H}$, where Sim queries to H and simulates H_1 and H_2.
We did not check the detail. This would solve the problems of Kyber, Saber, and Streamlined NTRU Prime.

Open Problems: We leave showing anonymity and the IND-CCA security of Kyber, Saber, and Streamlined NTRU Prime in the QROM as an important open problem as Gurbbs et al. posed. It might be interesting to investigate anonymity and robustness of them in the ROM.

Organization: section 2 reviews quantum random oracle models, definitions of primitives, and results of Grubbs et al. [GMP21a]. section 3 shows that strong pseudorandomness implies anonymity. **??** studies strong pseudorandomness of KEM/DEM framework. section 5 studies SXY's security properties. section 6 examines anonymity and robustness of NTRU.
For contents of appendices, see table of contents.

2 Preliminaries

Notations: A security parameter is denoted by κ. We use the standard O-notations. DPT, PPT, and QPT stand for deterministic polynomial time, probabilistic polynomial time, and quantum polynomial time, respectively. 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 a distribution χ, we often write "$x \leftarrow \chi$," which indicates that we take a sample x according to χ. For a finite set S, $U(S)$ denotes the uniform distribution over S. We often write "$x \leftarrow S$" instead of "$x \leftarrow U(S)$." For a set S and

[4] HQC-256 is not anonymous

Table 1. Summary of NIST PQC Round 3 KEM Candidates (finalists and alternate candidates) and the hybrid PKEs using them. In the first row, IND = Indistinguishability, SPR = Strong Pseudorandomness, ANO = Anonymity, CF = Collision Freeness, and ROB = Robustness under chosen-ciphertext attacks in the QROM. Y = Yes, N = No, ? = Unknown.

		KEM					PKE		
Name	Trans.	IND	SPR	ANO	CF	ROB	ANO	ROB	
Classic McEliece [ABC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	Y	Y	Y	N	N	Y	N	section L
Kyber [SAB+20]	$\mathsf{FO}^{\not\perp\prime}$	?	?	?	?	N	?	?	section M
NTRU [CDH+20]	SXY	Y	Y	Y	Y	N	Y	Y	section 6
Saber [DKR+20]	$\mathsf{FO}^{\not\perp\prime}$	?	?	?	?	N	?	?	section N
BIKE [ABB+20]	$\mathsf{FO}^{\not\perp}$	Y	Y	Y	Y	N	Y	Y	section O
FrodoKEM [NAB+20]	$\mathsf{FO}^{\not\perp\prime\prime}$	Y	Y	Y	Y	N	Y	Y	section P
HQC-128/192 [AAB+20]	$\mathsf{HFO}^{\perp}$	Y	Y	Y	Y	Y	Y	Y	section Q
HQC-256 [AAB+20]	$\mathsf{HFO}^{\perp}$	Y	N	N	Y	Y	N	Y	section Q
Streamlined NTRU Prime [BBC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	?	?	?	?	N	?	?	section R
NTRU LPRime [BBC+20]	$\mathsf{HFO}^{\not\perp,\mathsf{prf}}$	Y	Y	Y	Y	N	Y	Y	section S
SIKE [JAC+20]	$\mathsf{FO}^{\not\perp}$	Y	Y	Y	Y	N	Y	Y	section T

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

For a statement P (e.g., $r \in [0,1]$), we define $\mathrm{boole}(P) = 1$ if P is satisfied and 0 otherwise.

For two finite sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{F}(\mathcal{X},\mathcal{Y})$ denotes a set of all mapping from $\mathcal{X}$ to $\mathcal{Y}$.

Lemma 2.1 (Generic distinguishing problem with bounded probabilities [HKSU20, Lemma 2.9], adapted). *Let $\mathcal{X}$ be a finite set. Let $\delta \in [0,1]$. Let $\mathsf{F}\colon \mathcal{X} \to \{0,1\}$ be the following function: for each $x \in \mathcal{X}$, $\mathsf{F}(x) = 1$ with probability $\delta_x \le \delta$ and $\mathsf{F}(x) = 0$ else. Let $\mathsf{Z}\colon \mathcal{X} \to \{0,1\}$ be the zero function, that is, $\mathsf{Z}(x) = 0$ for all x. If an unbounded time quantum adversary $\mathcal{A}$ makes a query to F or Z at most Q times, then we have*

$$\left|\Pr[\mathcal{A}^{\mathsf{F}(\cdot)}() \to 1] - \Pr[\mathcal{A}^{\mathsf{Z}(\cdot)}() \to 1]\right| \le 8(Q+1)^2\delta.$$

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

Quantum Random Oracle Model: Roughly speaking, the quantum random oracle model (QROM) is an idealized model where a hash function is modeled as a publicly and quantumly accessible random oracle. In this paper, we model a quantum oracle O as a mapping $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus O(x)\rangle$, where $x \in \{0,1\}^n$, $y \in \{0,1\}^m$, and $O\colon \{0,1\}^n \to \{0,1\}^m$. See [BDF+11] for a more detailed description of the model.

Lemma 2.2 (QRO is PRF). *Let ℓ be a positive integer. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathsf{H}_{\mathsf{prf}}\colon \mathcal{M} \times \mathcal{X} \to \mathcal{Y}$ and $\mathsf{H}_q\colon \mathcal{X} \to \mathcal{Y}$ be two independent random oracles. If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H at most Q times, then we have*

$$\left|\Pr[s \leftarrow \mathcal{M} : \mathcal{A}^{\mathsf{H}_{\mathsf{prf}}(\cdot,\cdot),\mathsf{H}_{\mathsf{prf}}(s,\cdot)}() \to 1] - \Pr[\mathcal{A}^{\mathsf{H}_{\mathsf{prf}}(\cdot,\cdot),\mathsf{H}_q(\cdot)}() \to 1]\right| \le 2Q/\sqrt{|\mathcal{M}|},$$

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

See [SXY18] and [JZC+18] for the proof.

Lemma 2.3 (QRO is collision-resistant [Zha15, Theorem 3.1]). *There is a universal constant C such that the following holds: Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathsf{H}\colon \mathcal{X} \to \mathcal{Y}$ be a random oracle. If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H at most Q times, then we have*

$$\Pr_{\mathsf{H},\mathcal{A}}[(x,x') \leftarrow \mathcal{A}^{\mathsf{H}(\cdot)} : x \ne x' \wedge \mathsf{H}(x) = \mathsf{H}(x')] \le C(Q+1)^3/|\mathcal{Y}|,$$

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

Remark 2.1. We implicitly assume that $|\mathcal{X}| = \Omega(|\mathcal{Y}|)$, because of the birthday bound.

Lemma 2.4 (QRO is claw-free). *There is a universal constant C such that the following holds: Let $\mathcal{X}_0$ and $\mathcal{X}_1$ and $\mathcal{Y}$ be finite sets. Let $N_0 = |\mathcal{X}_0|$ and $N_1 = |\mathcal{X}_1|$. Without loss of generality, we assume $N_0 \leq N_1$. Let $\mathsf{H}_0 \colon \mathcal{X}_0 \to \mathcal{Y}$ and $\mathsf{H}_1 \colon \mathcal{X}_1 \to \mathcal{Y}$ be two random oracles.*
If an unbounded time quantum adversary $\mathcal{A}$ makes a query to H_0 and H_1 at most Q_0 and Q_1 times, then we have

$$\Pr[(x_0, x_1) \leftarrow \mathcal{A}^{\mathsf{H}_0(\cdot), \mathsf{H}_1(\cdot)} : \mathsf{H}_0(x_0) = \mathsf{H}_1(x_1)] \leq C(Q_0 + Q_1 + 1)^3 / |\mathcal{Y}|,$$

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

The following proof is due to Hosoyamada [Hos21].

Proof. Let us reduce the problem to collision-finding problem as follows: We assume that $\mathcal{X}_0$ and $\mathcal{X}_1$ are enumerable. Given $\mathsf{H} \colon [N_0 + N_1] \to \mathcal{Y}$, we define $\mathsf{H}_0 \colon \mathcal{X}_0 \to \mathcal{Y}$ and $\mathsf{H}_1 \colon \mathcal{X}_1 \to \mathcal{Y}$ by $\mathsf{H}_0(x) = \mathsf{H}(\mathsf{index}_0(x))$ and $\mathsf{H}_1(x) = \mathsf{H}(\mathsf{index}_1(x) + N_0)$, where $\mathsf{index}_i \colon \mathcal{X}_i \to [N_i]$ is an index function which returns the index of x in $\mathcal{X}_i$. H_0 and H_1 are random since H is a randomly chosen. If $\mathcal{A}$ finds the claw (x_0, x_1) for H_0 and H_1 with Q_0 and Q_1 queries, then we can find a collision $(\mathsf{index}_0(x_0), \mathsf{index}_1(x_1) + N_0)$ for H with $Q_0 + Q_1$ queries. Using Lemma 2.4, we obtain the bound as we wanted. □

2.1 Public-Key Encryption (PKE)

The model for PKE schemes is summarized as follows:

Definition 2.1. *A PKE scheme* PKE *consists of the following triple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

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

We review δ-correctness in Hofheinz, Hövelmanns, and Kiltz [HHK17].

Definition 2.2 (δ-Correctness). *Let $\delta = \delta(\kappa)$. We say* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* δ-correct *if*

$$\mathrm{Exp}_{(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)} \left[\max_{\mu \in \mathcal{M}} \Pr[c \leftarrow \mathsf{Enc}(ek, \mu) : \mathsf{Dec}(dk, c) \neq \mu] \right] \leq \delta.$$

In particular, we say that PKE is perfectly correct *if $\delta = 0$.*

We also define a key pair's accuracy.

Definition 2.3 (Accuracy [XY19]). *We say that a key pair (ek, dk) is* accurate *if for any $\mu \in \mathcal{M}$,*

$$\Pr_{c \leftarrow \mathsf{Enc}(ek, \mu)}[\mathsf{Dec}(dk, c) = \mu] = 1.$$

If a key pair is not accurate, then we call it inaccurate.

Security Notions: We review onewayness under chosen-plaintext attacks (OW-CPA), indistinguishability under chosen-plaintext attacks (IND-CPA), indistinguishability under chosen-ciphertext attacks (IND-CCA) [RS92, BDPR98], pseudorandom under chosen-ciphertext attacks (PR-CCA), and its strong version (SPR-CCA) for PKE. We define PR-CCA with simulator $\mathcal{S}$ as a generalization of IND\$-CCA-security in [vH04, Hop05]. We also review anonymity (ANON-CCA) [BBDP01], robustness (WROB-CCA and SROB-CCA) [Moh10], and collision-freeness (WCFR-CCA and SCFR-CCA) [Moh10]. We additionally define extended collision-freeness (XCFR), in which any efficient adversary cannot find a colliding ciphertext even if the adversary is given two decryption keys.

Definition 2.4 (Security notions for PKE). *Let* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme. Let* $\mathcal{D}_{\mathcal{M}}$ *be a distribution over the message space* $\mathcal{M}$.

For any $\mathcal{A}$ *and* goal-atk $\in$ {ind-cca, pr-cca, anon-cca}*, we define its* goal-atk *advantage against* PKE *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{S}],\mathcal{A}}(\kappa) := \left|2\Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{S}],\mathcal{A}}(\kappa) = 1] - 1\right|,$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{S}],\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*

For any $\mathcal{A}$ *and* goal-atk $\in$ {ow-cca, srob-cca, scfr-cca, wrob-cca, wcfr-cca, xcfr}*, we define its* goal-atk *advantage against* PKE *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{D}_{\mathcal{M}}],\mathcal{A}}(\kappa) := \Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{D}_{\mathcal{M}}],\mathcal{A}}(\kappa) = 1],$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{D}_{\mathcal{M}}],\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*

For GOAL-ATK $\in$ {OW-CCA, IND-CCA, PR-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA, XCFR}*, we say that* PKE *is* GOAL-ATK*-secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{D}_{\mathcal{M}},\mathcal{S}],\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary* $\mathcal{A}$*. We also say that* PKE *is* SPR-CCA*-secure if it is* PR-CCA*-secure and its simulator ignores ek. We also say that* PKE *is* GOAL-CPA*-secure if it is* GOAL-CCA*-secure even without the decryption oracle.*

Disjoint simulatability: We review disjoint simulatability defined in [SXY18].

Definition 2.5 (Disjoint simulatability [SXY18]). *Let* $\mathcal{D}_{\mathcal{M}}$ *denote an efficiently sampleable distribution on a set* $\mathcal{M}$*. A deterministic PKE scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with plaintext and ciphertext spaces* $\mathcal{M}$ *and* C *is* $\mathcal{D}_{\mathcal{M}}$-disjoint-simulatable *if there exists a PPT algorithm* $\mathcal{S}$ *that satisfies the followings:*

- *(Statistical disjointness:)*

$$\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) := \max_{(ek,dk)\in\mathsf{Gen}(1^\kappa;\mathcal{R}_{\mathsf{Gen}})} \Pr[c \leftarrow \mathcal{S}(1^\kappa, ek) : c \in \mathsf{Enc}(ek, \mathcal{M})]$$

  is negligible.
- *(Ciphertext-indistinguishability:) For any QPT adversary* $\mathcal{A}$*,*

$$\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}}(\kappa) := \left|\begin{array}{c}\Pr[(ek,dk) \leftarrow \mathsf{Gen}(1^\kappa), \mu^* \leftarrow \mathcal{D}_{\mathcal{M}}, c^* := \mathsf{Enc}(ek,\mu^*) : \mathcal{A}(ek,c^*) \to 1] \\ -\Pr[(ek,dk) \leftarrow \mathsf{Gen}(1^\kappa), c^* \leftarrow \mathcal{S}(1^\kappa, ek) : \mathcal{A}(ek,c^*) \to 1]\end{array}\right|$$

Liu and Wang gave a slightly modified version of statistical disjointness in [LW21]. As they noted, their definition below is enough to show the security proof.

$$\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) := \Pr[(ek,dk) \in \mathsf{Gen}(1^\kappa), c \leftarrow \mathcal{S}(1^\kappa, ek) : c \in \mathsf{Enc}(ek,\mathcal{M})]$$

Definition 2.6 (strong disjoint-simulatability). *We call* PKE *has* strong disjoint-simulatability *if* $\mathcal{S}$ *ignores ek.*

Remark 2.2. We note that a deterministic PKE scheme produced by TPunc [SXY18] or Punc [HKSU20] is not *strongly* disjoint-simulatable, because their simulator will output a random ciphertext $\mathsf{Enc}(ek, \hat{\mu})$ of a special plaintext $\hat{\mu}$, which depends on *ek*.

2.2 Key Encapsulation Mechanism (KEM)

The model for KEM schemes is summarized as follows:

Definition 2.7. *A KEM scheme* KEM *consists of the following triple of polynomial-time algorithms* $(\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$:

- $\overline{\mathsf{Gen}}(1^\kappa) \to (ek, dk)$*: a key-generation algorithm that on input* 1^κ*, where* κ *is the security parameter, outputs a pair of keys* (ek, dk)*. ek and dk are called the encapsulation key and decapsulation key, respectively.*
- $\overline{\mathsf{Enc}}(ek) \to (c, K)$*: an encapsulation algorithm that takes as input encapsulation key ek and outputs ciphertext* $c \in C$ *and key* $K \in \mathcal{K}$.
- $\overline{\mathsf{Dec}}(dk, c) \to K/\bot$*: a decapsulation algorithm that takes as input decapsulation key dk and ciphertext c and outputs key K or a rejection symbol* $\bot \notin \mathcal{K}$.

Definition 2.8 (δ-Correctness). *Let* $\delta = \delta(\kappa)$*. We say that* $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ *is* δ-correct *if*

$$\Pr[(ek,dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa), (c,K) \leftarrow \overline{\mathsf{Enc}}(ek) : \overline{\mathsf{Dec}}(dk,c) \neq K] \leq \delta(\kappa).$$

In particular, we say that KEM *is* perfectly correct *if* $\delta = 0$.

$\underline{\mathsf{Expt}^{\text{ow-cca}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{A}}(\kappa)}$

$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$\mu^* \leftarrow \mathcal{D}_{\mathcal{M}}$
$c^* \leftarrow \mathsf{Enc}(ek, \mu^*)$
$\mu' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}}(ek, c^*)$
return $\mathsf{boole}(\mu' \stackrel{?}{=} \mathsf{Dec}(dk, c^*))$

$\underline{\textsc{Dec}_a(c)}$

if $c = a$, return $\perp$
$\mu := \mathsf{Dec}(dk, c)$
return μ

$\underline{\textsc{Dec}_a(\mathsf{id}, c)}$

if $c = a$, return $\perp$
$\mu := \mathsf{Dec}(dk_{id}, c)$
return μ

$\underline{\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0, 1\}$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$(\mu_0, \mu_1, state) \leftarrow \mathcal{A}_1^{\textsc{Dec}_\perp(\cdot)}(ek)$
$c^* \leftarrow \mathsf{Enc}(ek, \mu_b)$
$b' \leftarrow \mathcal{A}_2^{\textsc{Dec}_{c^*}(\cdot)}(c^*, state)$
return $\mathsf{boole}(b = b')$

$\underline{\mathsf{Expt}^{\text{pr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0, 1\}$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$(\mu, state) \leftarrow \mathcal{A}_1^{\textsc{Dec}_\perp(\cdot)}(ek)$
$c_0^* \leftarrow \mathsf{Enc}(ek, \mu)$
$c_1^* \leftarrow \mathcal{S}(1^\kappa, ek)$
$b' \leftarrow \mathcal{A}_2^{\textsc{Dec}_{c_b^*}(\cdot)}(c_b^*, state)$
return $\mathsf{boole}(b = b')$

$\underline{\mathsf{Expt}^{\text{anon-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0, 1\}$
$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$(\mu, state) \leftarrow \mathcal{A}_1^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$c^* \leftarrow \mathsf{Enc}(ek_b, \mu)$
$b' \leftarrow \mathcal{A}_2^{\textsc{Dec}_{c^*}(\cdot,\cdot)}(c^*, state)$
return $\mathsf{boole}(b = b')$

$\underline{\mathsf{Expt}^{\text{wcfr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$(\mu, b) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$c \leftarrow \mathsf{Enc}(ek_b, \mu)$
$\mu' \leftarrow \mathsf{Dec}(dk_{1-b}, c)$
return $\mathsf{boole}(\mu = \mu' \neq \perp)$

$\underline{\mathsf{Expt}^{\text{scfr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$\mu_0 \leftarrow \mathsf{Dec}(dk_0, c)$
$\mu_1 \leftarrow \mathsf{Dec}(dk_1, c)$
return $\mathsf{boole}(\mu_0 = \mu_1 \neq \perp)$

$\underline{\mathsf{Expt}^{\text{xcfr}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$c \leftarrow \mathcal{A}(ek_0, dk_0, ek_1, dk_1)$
$\mu_0 \leftarrow \mathsf{Dec}(dk_0, c)$
$\mu_1 \leftarrow \mathsf{Dec}(dk_1, c)$
return $\mathsf{boole}(\mu_0 = \mu_1 \neq \perp)$

$\underline{\mathsf{Expt}^{\text{wrob-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$(\mu, b) \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$c \leftarrow \mathsf{Enc}(ek_b, \mu)$
$\mu' \leftarrow \mathsf{Dec}(dk_{1-b}, c)$
return $\mathsf{boole}(\mu' \neq \perp)$

$\underline{\mathsf{Expt}^{\text{srob-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)}$

$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$\mu_0 \leftarrow \mathsf{Dec}(dk_0, c)$
$\mu_1 \leftarrow \mathsf{Dec}(dk_1, c)$
return $\mathsf{boole}(\mu_0 \neq \perp \wedge \mu_1 \neq \perp)$

Fig. 1. Games for PKE schemes

Security: We review indistinguishability under chosen-plaintext attacks (IND-CPA), indistinguishability under chosen-ciphertext attacks (IND-CCA) [RS92, BDPR98], pseudorandomness under chosen-ciphertext attacks (PR-CCA), and its strong version (SPR-CCA) for KEM. We define PRCCA with simulator $\mathcal{S}$ as a generalization of IND$-CCA-security in [vH04, Hop05]. We also review anonymity (ANON-CCA), robustness (WROB-CCA and SROB-CCA), and collision-freeness (WCFR-CCA and SCFR-CCA) [GMP21a].
We also define *smoothness* under chosen-ciphertext attacks (denoted by SMT-CCA) by following smoothness of hash proof system [CS02]: Roughly speaking, we say a KEM scheme is SMT-CCA-secure if, given a random ciphertext c^* chosen by the simulator, any efficient adversary cannot distinguish random key K_0^* and decapsulated key $K_1^* = \mathsf{Dec}(dk, c^*)$.

Definition 2.9 (Security notions for KEM). *Let* $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ *be a KEM scheme.*
For any $\mathcal{A}$ *and* goal-atk $\in$ {ind-cca, pr-cca, smt-cca, anon-cca, srob-cca, scfr-cca}*, we define its* goal-atk *advantage against* KEM *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\kappa) := \left|2\Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\kappa) = 1] - 1\right|,$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*
For any $\mathcal{A}$ *and* goal-atk $\in$ {srob-cca, scfr-cca, wrob-cca, wcfr-cca}*, we define its* goal-atk *advantage against* KEM *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{KEM},\mathcal{A}}(\kappa) := \Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1],$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*
For GOAL-ATK $\in$ {IND-CCA, PR-CCA, SMT-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA}*, we say that* KEM *is* GOAL-ATK*-secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{KEM}[,\mathcal{S}],\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary* $\mathcal{A}$*. We say that* KEM *is* SPR-CCA*-secure (or* SSMT-CCA*-secure) if it is* PR-CCA*-secure (or* SMT-CCA*-secure) and its simulator ignores ek, respectively. We say that* KEM *is* WANON-CCA*-secure if it is* ANON-CCA*-secure where we modify the input* (ek_0, ek_1, c^*, K^*) *into* (ek_0, ek_1, c^*)*. We also say that* KEM *is* GOAL-CPA*-secure if it is* GOAL-CCA*-secure even without the decapsulation oracle.*

We additionally define ϵ-sparseness.

Definition 2.10. *Let* $\mathcal{S}$ *be a simulator for the* PR-CCA *security. We say that* KEM *is* ϵ-sparse *if*

$$\Pr[(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa), c^* \leftarrow \mathcal{S}(1^\kappa, ek) : \overline{\mathsf{Dec}}(dk, c) \neq \bot] \leq \epsilon.$$

2.3 Data Encapsulation

The model for DEM schemes is summarized as follows:

Definition 2.11. *A DEM scheme* DEM *consists of the following triple of polynomial-time algorithms* (E, D) *with key space* $\mathcal{K}$ *and message space* $\mathcal{M}$*:*

- $\mathsf{E}(K, \mu) \to d$*: an encapsulation algorithm that takes as input key* K *and data* μ *and outputs ciphertext* d*.*
- $\mathsf{D}(K, d) \to m/\bot$*: a decapsulation algorithm that takes as input key* K *and ciphertext* d *and outputs data* μ *or a rejection symbol* $\bot \notin \mathcal{M}$*.*

Definition 2.12 (Correctness). *We say* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *has* perfect correctness *if for any* $K \in \mathcal{K}$ *and any* $\mu \in \mathcal{M}$*, we have*

$$\Pr[\mathsf{D}(K, d) = \mu : d \leftarrow \mathsf{E}(K, \mu)] = 1.$$

We review indistinguishability under chosen-ciphertext attacks (IND-CCA), pseudorandomness under chosen-ciphertext attacks (PR-CCA), and pseudorandomness under one-time chosen-ciphertext attacks (PR-otCCA). Robustness of DEM (FROB and XROB) are taken from Farshim, Orlandi, and Roşi [FOR17].

$\underline{\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$

$(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(c^*, K_0^*) \leftarrow \overline{\mathsf{Enc}}(ek);$

$K_1^* \leftarrow \mathcal{K}$

$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(ek, c^*, K_b^*)$

return boole$(b = b')$

$\underline{\textsc{Dec}_a(c)}$

if $c = a$, return $\perp$

$K := \overline{\mathsf{Dec}}(dk, c)$

return K

$\underline{\textsc{Dec}_a(\mathsf{id}, c)}$

if $c = a$, return $\perp$

$K := \overline{\mathsf{Dec}}(dk_{\mathsf{id}}, c)$

return K

$\underline{\mathsf{Expt}^{\text{pr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$

$(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(c_0^*, K_0^*) \leftarrow \overline{\mathsf{Enc}}(ek);$

$(c_1^*, K_1^*) \leftarrow \mathcal{S}(1^\kappa, ek) \times \mathcal{K}$

$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}(\cdot)}(ek, c_b^*, K_b^*)$

return boole$(b = b')$

$\underline{\mathsf{Expt}^{\text{smt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$

$(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(c^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa, ek) \times \mathcal{K}$

$K_1^* \leftarrow \overline{\mathsf{Dec}}(dk, c^*)$

$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(ek, c^*, K_b^*)$

return boole$(b = b')$

$\underline{\mathsf{Expt}^{\text{anon-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$b \leftarrow \{0,1\}$

$(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(c^*, K^*) \leftarrow \overline{\mathsf{Enc}}(ek);$

$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot,\cdot)}(ek_0, ek_1, c^*, K^*)$

return boole$(b = b')$

$\underline{\mathsf{Expt}^{\text{wcfr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$b \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$

$(c, K_b) \leftarrow \overline{\mathsf{Dec}}(ek_b)$

$K_{1-b} \leftarrow \overline{\mathsf{Dec}}(dk_{1-b}, c)$

return boole$(K_0 = K_1 \neq \perp)$

$\underline{\mathsf{Expt}^{\text{scfr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$

$K_0 \leftarrow \overline{\mathsf{Dec}}(dk_0, c)$

$K_1 \leftarrow \overline{\mathsf{Dec}}(dk_1, c)$

return boole$(K_0 = K_1 \neq \perp)$

$\underline{\mathsf{Expt}^{\text{srob-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$

$K_0 \leftarrow \overline{\mathsf{Dec}}(dk_0, c)$

$K_1 \leftarrow \overline{\mathsf{Dec}}(dk_1, c)$

return boole$(K_0 \neq \perp \wedge K_1 \neq \perp)$

$\underline{\mathsf{Expt}^{\text{wrob-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)}$

$(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$

$b \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$

$(c, K_b) \leftarrow \overline{\mathsf{Dec}}(ek_b)$

$K_{1-b} \leftarrow \overline{\mathsf{Dec}}(dk_{1-b}, c)$

return boole$(K_{1-b} \neq \perp)$

Fig. 2. Games for KEM schemes

Definition 2.13 (Security notions for DEM). *Let* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *be a DEM scheme whose key space is* $\mathcal{K}$. *For* $\mu \in \mathcal{M}$, *let* $C_{|\mu|}$ *be a ciphertext space defined by the length of message* μ.
For any $\mathcal{A}$ *and* goal-atk $\in$ {ind-cca, pr-cca, pr-otcca}, *we define its* goal-atk *advantage against* DEM *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa) := \left| 2\Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa) = 1] - 1 \right|,$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*
For any $\mathcal{A}$ *and* goal-atk $\in$ {frob, xrob}, *we define its* goal-atk *advantage against* DEM *as follows:*

$$\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa) := \Pr[\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa) = 1],$$

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*

For GOAL-ATK $\in$ {IND-CCA, PR-CCA, PR-OTCCA, FROB, XROB}, *we say that* DEM *is* GOAL-ATK-*secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary* $\mathcal{A}$.

2.4 Review of Grubbs, Maram, and Paterson [GMP21a]

Grubbs et al. studied KEM's anonymity and hybrid PKE's anonymity and robustness, which is an extension of Mohassel [Moh10]. We use $\mathsf{KEM}^{\perp}$ and $\mathsf{KEM}^{\not\perp}$ to indicated KEM with explicit rejection and implicit rejection. For KEM with eplicit rejection, they showed the following theorem which generalizes Mohassel's theorem [Moh10]:

Theorem 2.1 ([GMP21a, Theorem 1]). *Let* $\mathsf{PKE}_{\mathsf{hy}} = \mathsf{Hyb}[\mathsf{KEM}^{\perp}, \mathsf{DEM}]$, *a hybrid PKE scheme obtained by composing* KEM *and* DEM. *(See Figure 4.)*

1. *If* $\mathsf{KEM}^{\perp}$ *is* wANON-CPA-*secure,* IND-CCA-*secure,* WROB-CCA-*secure, and* δ-*correct and* DEM *is* INT-CTXT-*secure, then* $\mathsf{PKE}_{\mathsf{hy}}$ *is* ANON-CCA-*secure.*
2. *If* $\mathsf{KEM}^{\perp}$ *is* SROB-CCA-*secure (and* WROB-CCA-*secure), then* $\mathsf{PKE}_{\mathsf{hy}}$ *is* SROB-CCA-*secure (and* WROB-CCA-*secure), respectively.*

Grubbs et al. [GMP21a] then treat KEM with implicit rejection, which is used in all NIST PQC Round 3 KEM candidates except HQC. Roughly speaking, they showed that the following two theorems on robustness and anonymity of hybrid PKE from KEM with implicit rejection:

Theorem 2.2 (Robustness of $\mathsf{PKE}_{\mathsf{hy}}$ [GMP21a, Theorem 2]). *Let* $\mathsf{PKE}_{\mathsf{hy}} = \mathsf{Hyb}[\mathsf{KEM}^{\not\perp}, \mathsf{DEM}]$. *If* $\mathsf{KEM}^{\not\perp}$ *is* SCFR-CCA-*secure (and* WCFR-CCA-*secure) and* DEM *is* FROB-*secure (and* XROB-*secure), then* $\mathsf{PKE}_{\mathsf{hy}}$ *is* SROB-CCA-*secure (and* WROB-CCA-*secure), respectively.*

Theorem 2.3 (Anonymity of $\mathsf{PKE}_{\mathsf{hy}}$ using $\mathsf{FO}^{\not\perp}$ [GMP21a, Theorem 7]). *Let* $\mathsf{PKE}_{\mathsf{hy}} = \mathsf{Hyb}[\mathsf{KEM}^{\not\perp}, \mathsf{DEM}]$. *If* PKE *is* δ-*correct, and* γ-*spreading,* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ *is* WCFR-CPA-*secure,* $\mathsf{KEM}^{\not\perp} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *is* ANON-CCA-*secure and* IND-CCA-*secure,* DEM *is* INT-CTXT-*secure, then* $\mathsf{PKE}_{\mathsf{hy}}$ *is* ANON-CCA-*secure.*

They also showed that the following theorem:

Theorem 2.4 (Anonymity of $\mathsf{KEM}^{\not\perp}$ using $\mathsf{FO}^{\not\perp}$ [GMP21a, Theorem 5]). *If* PKE *is* wANON-CPA-*secure,* OW-CPA-*secure, and* δ-*correct, and* $\mathsf{PKE}_1 = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ *is* SCFR-CPA-*secure, then a KEM scheme* $\mathsf{KEM} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ *is* ANON-CCA-*secure.*

In their security proof, they need to simulate *both* decapsulation oracles without secrets when they reduce from wANON-CPA-security. Jiang et al. [JZC+18] used the simulation trick that replaces $\mathsf{H}(\mu, c)$ with $\mathsf{H}_q(\mathsf{Enc}(ek, \mu))$ if $c = \mathsf{Enc}(ek, \mu)$ and $\mathsf{H}'_q(m, c)$ else, which helps the simulation of the decapsulation oracle without secrets. Grubbs et al. extended this trick to simulate *two* decapsulation oracles by replacing $\mathsf{H}(\mu, c)$ with $\mathsf{H}_{q,i}(\mathsf{Enc}(ek_i, \mu))$ if $c = \mathsf{Enc}(ek_i, \mu)$ and $\mathsf{H}'_q(\mu, c)$ else. Notice that this extended simulation heavily depends on the fact that H takes μ and c. If the random oracle takes μ only, their trick fails the simulation.

$$
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
b \leftarrow \{0,1\}\\
K \leftarrow \mathcal{K}\\
(\mu_0, \mu_1, \mathit{state}) \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{\perp}(\cdot)}(1^\kappa)\\
d^* \leftarrow \mathsf{E}(K, \mu_b)\\
b' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{d^*}(\cdot)}(d^*, \mathit{state})\\
b_l \leftarrow \mathsf{boole}(|\mu_0| = |\mu_1|)\\
\textbf{return } \mathsf{boole}(b = b' \wedge b_l)
\end{array}
\qquad
\begin{array}{l}
\underline{\textsc{Enc}(\mu)}\\
d \leftarrow \mathsf{E}(K, \mu)\\
\textbf{return } d\\
\\
\underline{\textsc{Dec}_a(d)}\\
\text{if } d = a, \text{ return } \perp\\
\mu \leftarrow \mathsf{D}(K, d)\\
\textbf{return } \mu
\end{array}
$$

$$
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{int-ctxt}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
K \leftarrow \mathcal{K}\\
w \leftarrow \perp\\
L \leftarrow \emptyset\\
\mathcal{A}^{\textsc{Enc2}(\cdot),\textsc{Dec2}(\cdot)}(1^\kappa)\\
\textbf{return } w
\end{array}
\qquad
\begin{array}{l}
\underline{\textsc{Enc2}(\mu)}\\
d \leftarrow \mathsf{E}(K, \mu)\\
L \leftarrow L \cup \{d\}\\
\textbf{return } d
\end{array}
\qquad
\begin{array}{l}
\underline{\textsc{Dec2}(d)}\\
\mu \leftarrow \mathsf{D}(K, d)\\
\text{if } \mu \neq \perp \text{ and } d \notin L, \text{ set } w = \top\\
\textbf{return } \mu
\end{array}
$$

$$
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{pr-cca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
b \leftarrow \{0,1\}\\
K \leftarrow \mathcal{K}\\
(\mu, \mathit{state}) \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{\perp}(\cdot)}(1^\kappa)\\
d_0^* \leftarrow \mathsf{E}(K, \mu)\\
d_1^* \leftarrow U(C_{|\mu|})\\
b' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{d_b^*}(\cdot)}(d_b^*, \mathit{state})\\
\textbf{return } \mathsf{boole}(b = b')
\end{array}
\qquad
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{pr-otcca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
b \leftarrow \{0,1\}\\
K \leftarrow \mathcal{K}\\
(\mu, \mathit{state}) \leftarrow \mathcal{A}(1^\kappa)\\
d_0^* \leftarrow \mathsf{E}(K, \mu)\\
d_1^* \leftarrow U(C_{|\mu|})\\
b' \leftarrow \mathcal{A}^{\textsc{Dec}_{d_b^*}(\cdot)}(d_b^*, \mathit{state})\\
\textbf{return } \mathsf{boole}(b = b')
\end{array}
$$

$$
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{frob}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
(d, k_0, k_1) \leftarrow \mathcal{A}(1^\kappa)\\
\mu_0 \leftarrow \mathsf{D}(k_0, d)\\
\mu_1 \leftarrow \mathsf{D}(k_1, d)\\
b \leftarrow \mathsf{boole}(\mu_0 \neq \perp \wedge \mu_1 \neq \perp)\\
b_k \leftarrow \mathsf{boole}(k_0 \neq k_1)\\
\textbf{return } \mathsf{boole}(b \wedge b_k)
\end{array}
\qquad
\begin{array}{l}
\underline{\mathsf{Expt}^{\text{xrob}}_{\mathsf{DEM},\mathcal{A}}(\kappa)}\\
(\mu_0, k_0, R_0, k_1, d_1) \leftarrow \mathcal{A}(1^\kappa)\\
d_0 \leftarrow \mathsf{E}(k_0, \mu_0; R_0)\\
\mu_1 \leftarrow \mathsf{D}(k_1, d_1)\\
b \leftarrow \mathsf{boole}(\mu_0 \neq \perp \wedge \mu_1 \neq \perp)\\
b_k \leftarrow \mathsf{boole}(k_0 \neq k_1)\\
b_c \leftarrow \mathsf{boole}(d_0 = d_1 \neq \perp)\\
\textbf{return } \mathsf{boole}(b \wedge b_k \wedge b_c)
\end{array}
$$

Fig. 3. Games for DEM schemes

3 Strong Pseudorandomness Implies Anonymity

We observe that strong pseudorandomness of PKE/KEM immediately implies anonymity of PKE/KEM, which may be folklore. For completeness, we include the proof for PKE in subsection B.1.

Theorem 3.1. *If* PKE *is* SPR-CCA*-secure, then it is* ANON-CCA*-secure. If* KEM *is* SPR-CCA*-secure, then it is* ANON-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ against the ANON-CCA *security of* PKE/KEM*, there exist $\mathcal{A}_{10}$ and $\mathcal{A}_{11}$ against the* SPR-CCA *security of* PKE/KEM *such that*

$$\mathsf{Adv}^{\text{anon-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}_{10}}(\kappa) + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}_{11}}(\kappa),$$
$$\mathsf{Adv}^{\text{anon-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}_{10}}(\kappa) + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}_{11}}(\kappa).$$

4 Strong Pseudorandomness of Hybrid PKE

The hybrid PKE $\mathsf{PKE}_{\mathsf{hy}} = (\mathsf{Gen}_{\mathsf{hy}}, \mathsf{Enc}_{\mathsf{hy}}, \mathsf{Dec}_{\mathsf{hy}})$ constructed from $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ and $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ is summarized as in Figure 4

$\mathsf{Gen}_{\mathsf{hy}}(1^\kappa)$	$\mathsf{Enc}_{\mathsf{hy}}(ek, \mu)$	$\mathsf{Dec}_{\mathsf{hy}}(dk, ct = (c, d))$
$(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$	$(c, K) \leftarrow \overline{\mathsf{Enc}}(ek)$	$K' \leftarrow \overline{\mathsf{Dec}}(dk, c)$
return (ek, dk)	$d \leftarrow \mathsf{E}(K, \mu)$	**if** $K' = \perp$ **then return** $\perp$
	return $ct := (c, d)$	$\mu' \leftarrow \mathsf{D}(K', d)$
		if $\mu' = \perp$ **then return** $\perp$
		return μ'

Fig. 4. $\mathsf{PKE}_{\mathsf{hy}} = \mathsf{Hyb}[\mathsf{KEM}, \mathsf{DEM}]$

We show the following two theorems on SPR-CCA security of a hybrid PKE:

Theorem 4.1. *Let $\mathsf{PKE}_{\mathsf{hy}} = (\mathsf{Gen}_{\mathsf{hy}}, \mathsf{Enc}_{\mathsf{hy}}, \mathsf{Dec}_{\mathsf{hy}})$ be a hybrid encryption scheme obtained by composing a KEM scheme $\mathsf{KEM}^{\perp} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ and a DEM scheme $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ that share key space $\mathcal{K}$. If $\mathsf{KEM}^{\perp}$ is* SPR-CCA*-secure, δ-correct with negligible δ, and ϵ-sparse and* DEM *is* PR-OTCCA*-secure and* INT-CTXT*-secure, then $\mathsf{PKE}_{\mathsf{hy}}$ is* SPR-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ against the SPR-CCA *security of $\mathsf{PKE}_{\mathsf{hy}}$, there exist $\mathcal{A}_{23}$ against the* SPR-CCA *security of $\mathsf{KEM}^{\perp}$, $\mathcal{A}_{34}$ against the* SPR-OTCCA *security of* DEM*, and $\mathcal{A}_{45}$ against the* INT-CTXT *security of* DEM *such that*

$$\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\perp},\mathcal{S},\mathcal{A}_{23}}(\kappa) + \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM},\mathcal{A}_{34}}(\kappa) + \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathcal{A}_{45}}(\kappa) + \epsilon + \delta.$$

Theorem 4.2. *Let $\mathsf{PKE}_{\mathsf{hy}} = (\mathsf{Gen}_{\mathsf{hy}}, \mathsf{Enc}_{\mathsf{hy}}, \mathsf{Dec}_{\mathsf{hy}})$ be a hybrid encryption scheme obtained by composing a KEM scheme $\mathsf{KEM}^{\not\perp} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ and a DEM scheme $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ that share key space $\mathcal{K}$. If* KEM *is* SPR-CCA*-secure,* SSMT-CCA*-secure, and δ-correct with negligible δ and* DEM *is* PR-OTCCA*-secure, then $\mathsf{PKE}_{\mathsf{hy}}$ is* SPR-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ against the SPR-CCA *security of $\mathsf{PKE}_{\mathsf{hy}}$, there exist $\mathcal{A}_{23}$ against the* SPR-CCA *security of $\mathsf{KEM}^{\not\perp}$, $\mathcal{A}_{34}$ against the* SPR-OTCCA *security of* DEM*, and $\mathcal{A}_{45}$ against the* SSMT-CCA *security of $\mathsf{KEM}^{\not\perp}$ such that*

$$\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\not\perp},\mathcal{S},\mathcal{A}_{23}}(\kappa) + \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM},\mathcal{A}_{34}}(\kappa) + \mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM}^{\not\perp},\mathcal{S},\mathcal{A}_{45}}(\kappa) + \delta.$$

Table 2. Summary of Games for the Proof of Theorem 4.1

Game	c^* and K^*	d^*	Decryption oracle	justification
Game_0	$\overline{\mathsf{Enc}}(ek)$	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$	
Game_1	$\overline{\mathsf{Enc}}(ek)$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$	conceptual change
Game_2	$\overline{\mathsf{Enc}}(ek)$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	δ-correctness of $\mathsf{KEM}^{\perp}$
Game_3	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-CCA security of $\mathsf{KEM}^{\perp}$
Game_4	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-otCCA security of DEM
Game_5	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$; use $\perp^*$ if $c = c^*$	INT-CTXT security of DEM
Game_6	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	ϵ-sparseness of $\mathsf{KEM}^{\perp}$
Game_7	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	conceptual change

4.1 Proof of Theorem 4.1

Let us consider Game_i for $i = 0, \dots, 6$. We summarize the games in Table 3. Let S_i denote the event that the adversary outputs $b' = 1$ in Game_i.
Let $\mathcal{S}$ be the simulator for the SPR-CCA security of $\mathsf{KEM}^{\not\perp}$. We define $\mathcal{S}_{\mathsf{hy}}(1^\kappa, |\mu^*|) := \mathcal{S}(1^\kappa) \times U(C_{|\mu^*|})$ be the simulator for the SPR-CCA security of $\mathsf{PKE}_{\mathsf{hy}}$.
The security proof is similar to the security proof of the IND-CCA security of KEM/DEM [CS03] for $\mathsf{Game}_0, \dots, \mathsf{Game}_4$. We need to take care of pseudorandom ciphertexts when moving from Game_4 to Game_7 and require the INT-CTXT security of DEM and the ϵ-sparseness of $\mathsf{KEM}^{\perp}$.

Game_0: This is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}}, \mathcal{S}_{\mathsf{hy}}, \mathcal{A}}(\kappa)$ with $b = 0$. Given μ^*, the target ciphertext is computed as follows:

$$(c^*, K^*) \leftarrow \overline{\mathsf{Enc}}(ek); d^* \leftarrow \mathsf{E}(K^*, \mu^*);\ \text{return } ct^* = (c^*, d^*).$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}}, \mathcal{S}_{\mathsf{hy}}, \mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1: In this game, c_0^* and K_0^* are generated before invoking $\mathcal{A}$ with ek. This change is just conceptual and we have

$$\Pr[S_0] = \Pr[S_1].$$

Game_2: In this game, the decryption oracle uses K^* if $c = c^*$ instead of $K = \overline{\mathsf{Dec}}(dk, c^*)$. Game_1 and Game_2 differ if correctly generated ciphertext c^* with K^* is decapsulated into different $K \neq K^*$ or $\perp$, which violates the correctness and occurs with probability at most δ. Hence, the difference of Game_1 and Game_2 is bounded by δ and we have

$$|\Pr[S_1] - \Pr[S_2]| \leq \delta.$$

This is corresponding to the event $\mathsf{BadKeyPair}$ in [CS03].

Game_3: In this game, the challenger uses random (c^*, K^*) and uses K^* in DEM. The challenge ciphertext is generated as follows:

$$(c^*, K^*) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{K}); d^+ \leftarrow \mathsf{E}(K^*, \mu^*);\ \text{return } ct^* = (c^*, d^+).$$

The difference is bounded by SPR-CCA security of $\mathsf{KEM}^{\not\perp}$: There is an adversary $\mathcal{A}_{23}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\not\perp}, \mathcal{S}, \mathcal{A}_{23}}(\kappa).$$

We omit the detail of $\mathcal{A}_{23}$, since it is straightforward.

Game_4: In this game, the challenger uses random d^*. The challenge ciphertext is generated as follows:

$$(c^*, K^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}; d^* \leftarrow U(C_{|\mu^*|});\ \text{return } ct^* = (c^*, d^*).$$

The difference is bounded by SPR-otCCA security of DEM: There is an adversary $\mathcal{A}_{34}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM}, \mathcal{A}_{34}}(\kappa).$$

We omit the detail of $\mathcal{A}_{34}$ since it is straightforward.

Table 3. Summary of Games for the Proof of Theorem 4.2

Game	c^* and K^*	d^*	Decryption oracle	justification
Game_0	$\overline{\mathsf{Enc}}(ek)$	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$	
Game_1	$\overline{\mathsf{Enc}}(ek)$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$	conceptual change
Game_2	$\overline{\mathsf{Enc}}(ek)$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	δ-correctness of $\mathsf{KEM}^{\not\perp}$
Game_3	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$\mathsf{E}(K^*, \mu^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-CCA security of $\mathsf{KEM}^{\not\perp}$
Game_4	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-oTCCA security of DEM
Game_5	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$ at the beginning	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	SSMT-CCA security of $\mathsf{KEM}^{\not\perp}$
Game_6	$\mathcal{S}(1^\kappa) \times U(\mathcal{K})$	$U(C_{\lvert\mu^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	conceptual change

Game_5*:* We replace the decryption oracle. If given $ct = (c^*, d)$, the decryption oracle always return $\perp$.
Let Forge be an event that the adversary queries $d \neq d^*$ decrypted into some $\mu \neq \perp$ under K^*. Game_4 and Game_5 are equivalent until the event Forge occurs in Game_4. There is an adversary $\mathcal{A}_{45}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_4] - \Pr[S_5]| \leq \Pr[\mathsf{Forge}] \leq \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathcal{A}_{45}}(\kappa).$$

We omit the detail of $\mathcal{A}_{45}$ since it is straightforward.

Game_6*:* We replace the decryption oracle in Game_5 with the original one.
Let D be an event that a randomly chosen $c^* \leftarrow \mathcal{S}(1^\kappa)$ is decapsulated into a key $K \neq \perp$. Game_5 and Game_6 are equivalent unless the event D occurs. Since $\mathsf{KEM}^{\perp}$ is ϵ-sparse, we have

$$|\Pr[S_5] - \Pr[S_6]| \leq \Pr[\mathsf{D}] \leq \epsilon.$$

Game_7*:* We change the timing of the generation of (c^*, K^*) as the original. This change is just conceptual and we have

$$\Pr[S_6] = \Pr[S_7].$$

Notice that this is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa)$ with $b = 1$, thus, we have

$$\Pr[S_7] = \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain the bound in the statement as follows:

$$\begin{aligned}\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}}\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_7]| \leq \sum_i |\Pr[S_i] - \Pr[S_{i+1}]| \\ &\leq \delta + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\perp},\mathcal{S},\mathcal{A}_{23}}(\kappa) + \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM},\mathcal{A}_{34}}(\kappa) + \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathcal{A}_{45}}(\kappa) + \delta + \epsilon.\end{aligned}$$

□

4.2 Proof of Theorem 4.2

Let us consider Game_i for $i = 0, \dots, 6$. We summarize the games in Table 3. Let S_i denote the event that the adversary outputs $b' = 1$ in Game_i.
Let $\mathcal{S}$ be the simulator for the SPR-CCA security of $\mathsf{KEM}^{\not\perp}$. We define $\mathcal{S}_{\mathsf{hy}}(1^\kappa, |\mu^*|) := \mathcal{S}(1^\kappa) \times U(C_{|\mu^*|})$ be the simulator for the SPR-CCA security of $\mathsf{PKE}_{\mathsf{hy}}$.
The security proof is similar to the security proof of the IND-CCA security of KEM/DEM [CS03] for $\mathsf{Game}_0, \dots, \mathsf{Game}_4$. We need to take care of pseudorandom ciphertexts when moving from Game_4 to Game_5 and require the SSMT-CCA security of $\mathsf{KEM}^{\not\perp}$.

Game_0*:* This is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa)$ with $b = 0$. Given μ^*, the target ciphertext is computed as follows:

$$(c^*, K^*) \leftarrow \overline{\mathsf{Enc}}(ek); d^* \leftarrow \mathsf{E}(K^*, \mu^*);\ \text{return } ct^* = (c^*, d^*).$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}},\mathcal{S}_{\mathsf{hy}},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1*:* In this game, c_0^* and K_0^* are generated before invoking $\mathcal{A}$ with ek. This change is just conceptual and we have

$$\Pr[S_0] = \Pr[S_1].$$

Game_2*:* In this game, the decryption oracle uses K^* if $c = c^*$ instead of $K = \overline{\mathsf{Dec}}(dk, c^*)$. Game_1 and Game_2 differ if correctly generated ciphertext c^* with K^* is decapsulated into different $K \neq K^*$ or $\perp$, which violates the correctness and occurs with probability at most δ. Hence, the difference of Game_1 and Game_2 is bounded by δ and we have

$$|\Pr[S_1] - \Pr[S_2]| \leq \delta.$$

This is corresponding to the event BadKeyPair in [CS03].

Game_3*:* In this game, the challenger uses random (c^*, K^*) and uses K^* in DEM. The challenge ciphertext is generated as follows:

$$(c^*, K^*) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{K}); d^+ \leftarrow \mathsf{E}(K^*, \mu^*);\ \text{return } ct^* = (c^*, d^+).$$

The difference is bounded by SPR-CCA security of $\mathsf{KEM}^{\not\perp}$: There is an adversary $\mathcal{A}_{23}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\not\perp}, \mathcal{S}, \mathcal{A}_{23}}(\kappa).$$

We omit the detail of $\mathcal{A}_{23}$, since it is straightforward.

Game_4*:* In this game, the challenger uses random d^*. The challenge ciphertext is generated as follows:

$$(c^*, K^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}; d^* \leftarrow U(C_{|\mu^*|});\ \text{return } ct^* = (c^*, d^*).$$

The difference is bounded by SPR-OTCCA security of DEM: There is an adversary $\mathcal{A}_{34}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM}, \mathcal{A}_{34}}(\kappa).$$

We omit the detail of $\mathcal{A}_{34}$ since it is straightforward.

Game_5*:* We replace the decryption oracle. If given $ct = (c^*, d)$, the decryption oracle uses $K = \overline{\mathsf{Dec}}(dk, c^*)$ instead of K^*.
The difference is bounded by SSMT-CCA security of $\mathsf{KEM}^{\not\perp}$: There is an adversary $\mathcal{A}_{45}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM}^{\not\perp}, \mathcal{S}, \mathcal{A}_{45}}(\kappa).$$

We omit the detail of $\mathcal{A}_{45}$ since it is straightforward.

Game_6*:* We change the timing of the generation of (c^*, K^*). This change is just conceptual and we have

$$\Pr[S_5] = \Pr[S_6].$$

Notice that this is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}}, \mathcal{S}_{\mathsf{hy}}, \mathcal{A}}(\kappa)$ with $b = 1$, thus, we have

$$\Pr[S_6] = \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}}, \mathcal{S}_{\mathsf{hy}}, \mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain the bound in the statement as follows:

$$\begin{aligned}
\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE}_{\mathsf{hy}}, \mathcal{S}_{\mathsf{hy}} \mathcal{A}}(\kappa) = |\Pr[S_0] - \Pr[S_6]| &\leq \sum_i |\Pr[S_i] - \Pr[S_{i+1}]| \\
&\leq \delta + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM}^{\not\perp}, \mathcal{S}, \mathcal{A}_{23}}(\kappa) + \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM}, \mathcal{A}_{34}}(\kappa) + \mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM}^{\not\perp}, \mathcal{S}, \mathcal{A}_{45}}(\kappa) + \delta.
\end{aligned}$$

□

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{D}_\mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c)$
$s \leftarrow \{0,1\}^\ell$	$c := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \bot$ or $c \neq \mathsf{Enc}(ek, \mu')$
$\overline{dk} := (dk, ek, s)$	$K := \mathsf{H}(\mu)$	**then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c)$
return $(ek, \overline{dk})$	**return** (c, K)	**else return** $K := \mathsf{H}(\mu')$

Fig. 5. $\mathsf{KEM} = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$

5 Properties of SXY

Let us review SXY [SXY18] as known as $\mathsf{U}_m^{\not\perp}$ with explicit re-encryption check [HHK17].
Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic PKE scheme. Let $\mathcal{M}$, C, and $\mathcal{K}$ be a plaintext, ciphertext, and key space of PKE, respectively. Let $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$ and $\mathsf{H}_{\mathsf{prf}}\colon \{0,1\}^\ell \times C \to \mathcal{K}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$ is defined as in Figure 5.

5.1 SPR-CCA Security

We first show KEM is strongly pseudorandom if the underlying PKE is strongly disjoint-simulatable.

Theorem 5.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. *Suppose that a ciphertext space* C *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable and* δ*-correct with negligible* δ*, then* $\mathsf{KEM} = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$ *is* SPR-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ *against the* SPR-CCA *security of* KEM *issuing at most* q_{DEC} *queries to the decapsulation oracle and* q_{G}, q_{H}, *and* $q_{\mathsf{H}_{\mathsf{prf}}}$ *queries to* G, H, *and* $\mathsf{H}_{\mathsf{prf}}$*, respectively, there exist* $\mathcal{A}_{34}$ *against ciphertext-indistinguishability of* PKE *such that*

$$\mathrm{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) \leq \mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_\mathcal{M},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathrm{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4\delta$$
$$+ 16(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta + 16(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2\delta + 4(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

Theorem 5.2. *Suppose that a ciphertext space* C *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable and* δ*-correct with negligible* δ*, then* $\mathsf{KEM} = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$ *is* SPR-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ *against the* SPR-CCA *security of* KEM *issuing at most* q_{DEC} *queries to the decapsulation oracle and* q_{G}, q_{H}, *and* $q_{\mathsf{H}_{\mathsf{prf}}}$ *queries to* G, H, *and* $\mathsf{H}_{\mathsf{prf}}$*, respectively, there exist* $\mathcal{A}_{34}$ *against ciphertext-indistinguishability of* PKE *such that*

$$\mathrm{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A},\mathcal{S}}(\kappa) \leq \mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_\mathcal{M},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathrm{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2} + 4\delta.$$

We here prove Theorem 5.1 because the proof of Theorem 5.2 is a special case of Theorem 5.1.

Proof of Theorem 5.1: We use the game-hopping proof. We consider Game_i for $i = 0, \dots, 8$. We summarize the games in Table 4. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i. We extend the security proof for SXY in [LW21], which extends the security proof for SXY [SXY18, XY19] to the case that the underlying PKE is derandomized by $\mathsf{KC} \circ \mathsf{T}$.

Game_0*:* This game is the original game $\mathrm{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. Thus, we have

$$\Pr[S_0] = 1 - \Pr[\mathrm{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1*:* This game is the same as Game_0 except that $\mathsf{H}_{\mathsf{prf}}(s, c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q : C \to \mathcal{K}$ is another random oracle. We remark that $\mathcal{A}$ is not given direct access to H_q.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_0] - \Pr[S_1]| \leq 2(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

Table 4. Summary of Games for the Proof of Theorem 5.1. We define $g(\mu) = \mathsf{Enc}(ek, \mu) = \mathsf{Enc}_0(ek, \mu; \mathsf{G}(\mu))$.

Game	H	G	c^*	K^*	Decryption valid c	Decryption invalid c	justification
Game_0	H	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathsf{prf}}(s,c)$	
Game_1	H	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	Lemma 2.2
$\mathsf{Game}_{1.1}$	H	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
$\mathsf{Game}_{1.2}$	$\mathsf{H}'_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	if key is not bad
Game_2	$\mathsf{H}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	if key is not bad
Game_3	$\mathsf{H}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	conceptual
$\mathsf{Game}_{3.1}$	$\mathsf{H}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_4	$\mathsf{H}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	DS-IND
Game_5	$\mathsf{H}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	statistical disjointness
$\mathsf{Game}_{5.1}$	$\mathsf{H}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_6	$\mathsf{H}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	conceptual
$\mathsf{Game}_{6.1}$	$\mathsf{H}'_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	if key is not bad
$\mathsf{Game}_{6.2}$	H	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	if key is not bad
Game_7	H	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_8	H	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathsf{prf}}(s,c)$	Lemma 2.2

Definition of $\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$*:* We next consider a set of good random oracles G. This definition follows [HHK17, JZC+18, HKSU20, LW21].
For $(ek, dk) \in \mathsf{Gen}_0()$ and $\mu \in \mathcal{M}$, we define a set of good randomness $\mathcal{R}^{\mathrm{good}}_{ek,dk,\mu} := \{r \in \mathcal{R} : \mathsf{Dec}_0(dk, \mathsf{Enc}_0(ek, \mu; r)) = \mu\}$, which could be empty. Let $\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$ be a set of functions $\mathsf{G}\colon \mathcal{M} \to \mathcal{R}$ satisfying $\mathsf{G}(\mu) \in \mathcal{R}^{\mathrm{good}}_{ek,dk,\mu}$ for all $\mu \in \mathcal{M}$. Define $\delta_{ek,dk,\mu} = \left|\mathcal{R} \setminus \mathcal{R}^{\mathrm{good}}_{ek,dk,\mu}\right|/|\mathcal{R}|$, which is the fraction of the bad randomness. Define $\delta_{ek,dk} := \max_{\mu\in\mathcal{M}} \delta_{ek,dk,\mu}$. We note that $\delta = \mathrm{Exp}_{(ek,dk)\leftarrow\mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}]$.

$\mathsf{Game}_{1.1}$*:* This game is the same as Game_1 except that the random oracle G is chosen from $\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$ instead of $\mathcal{F}(\mathcal{M},\mathcal{R})$.
If we fix (ek, dk), then we have $|\Pr[S_1 \mid (ek,dk)] - \Pr[S_{1.1} \mid (ek,dk)]| \le 8(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta_{ek,dk}$. (See [HKSU20, Theorem 3.2] and [LW21, Claim 1] for the analysis using Lemma 2.1.) Taking the average over $(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$, we obtain

$$|\Pr[S_1] - \Pr[S_{1.1}]| \le 8(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2 \mathrm{Exp}_{(ek,dk)\leftarrow\mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] = 8(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta.$$

Definition of Bad*:* We next define a bad event for key pairs. This definition follows [LW21]. Let us define an event Bad that there exists $\mu \in \mathcal{M}$ such that any $r \in \mathcal{R}$ is bad randomness, that is,

$$\mathsf{Bad} := \mathrm{boole}\left(\exists \mu \in \mathcal{M} : \mathcal{R}^{\mathrm{good}}_{ek,dk,\mu} = \emptyset\right),$$

where randomness is taken over $(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$.
We have $\Pr[\mathsf{Bad}] \le \delta$ ([LW21, Claim 3]). According to Lemma A.1, for any p, we also have

$$|\Pr[S_{1.1}] - p| \le |\Pr[S_{1.1} \wedge \neg\mathsf{Bad}] - p| + \delta.$$

$\mathsf{Game}_{1.2}$*:* This game is the same as $\mathsf{Game}_{1.1}$ except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}'_q(\mathsf{Enc}(ek,\cdot))$ where $\mathsf{H}'_q\colon \mathcal{C} \to \mathcal{K}$ is yet another random oracle. We remark that the decapsulation oracle and the generation of K^* also use $\mathsf{H}'_q(\mathsf{Enc}(ek,\cdot))$ as $\mathsf{H}(\cdot)$.
If $\neg\mathsf{Bad}$ occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$ is perfectly correct from the definition of G and $g(\mu) := \mathsf{Enc}(ek, \mu; \mathsf{G}(\mu))$ is *injective*. Thus, $\mathsf{H}'_q \circ g\colon \mathcal{M} \to \mathcal{K}$ is a random function and the two games $\mathsf{Game}_{1.1}$ and $\mathsf{Game}_{1.2}$ are equivalent if $\neg\mathsf{Bad}$ occurs. We have

$$\Pr[S_{1.1} \wedge \neg\mathsf{Bad}] = \Pr[S_{1.2} \wedge \neg\mathsf{Bad}].$$

See [XY19, Lemma 4.3] and [LW21, Claim 4] for the detail.

Game_2*:* This game is the same as $\mathsf{Game}_{1.2}$ except that the random oracle H is simulated by $\mathsf{H}_q \circ g$ instead of $\mathsf{H}'_q \circ g$.
If $\neg\mathsf{Bad}$ occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is perfectly correct from the definition of G. Hence, the two games $\mathsf{Game}_{1.2}$ and Game_2 are equivalent, because a value of $\mathsf{H}'_q(c)$ for an invalid c is not used in $\mathsf{Game}_{1.2}$: that is, we have

$$\Pr[S_{1.2} \wedge \neg\mathsf{Bad}] = \Pr[S_2 \wedge \neg\mathsf{Bad}].$$

See the proof of [XY19, Lemma 4.4] and [LW21, Claim 5] for the detail.

Game_3: This game is the same as Game_2 except that K^* is set as $\mathsf{H}_q(c^*)$ and the decapsulation oracle always returns $\mathsf{H}'_q(c)$ as long as $c \neq c^*$. This decapsulation oracle will denoted by $\textsc{Dec}$'.
If $\neg$Bad occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is perfectly correct from the definition of G. Thus, the two games Game_2 and Game_3 are equivalent and we have

$$\Pr[S_2 \wedge \neg\text{Bad}] = \Pr[S_3 \wedge \neg\text{Bad}].$$

See the proof of [XY19, Lemma 4.5] for the detail.
According to Lemma A.1, for any p, we have

$$|\Pr[S_3 \wedge \neg\text{Bad}] - p| \leq |\Pr[S_3] - p| + \delta.$$

$\text{Game}_{3.1}$: This game is the same as Game_3 except that G is chosen from $\mathcal{F}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$. We have

$$|\Pr[S_3] - \Pr[S_{3.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \text{Exp}_{(ek,dk) \leftarrow \mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] = 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \delta.$$

(We note that H and the challenge ciphertext also query to G internally.)

Game_4: This game is the same as Game_3 except that c^* is generated by $\mathcal{S}(1^\kappa)$.
The difference between two games Game_3 and Game_4 is bounded by the advantage of ciphertext indistinguishability in disjoint simulatability as in [XY19, Lemma 4.7]. We have

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}, \mathcal{D}_\mathcal{M}, \mathcal{S}, \mathcal{A}_{34}}(\kappa).$$

Game_5: This game is the same as Game_4 except that $K^* \leftarrow \mathcal{K}$ instead of $K^* \leftarrow \mathsf{H}_q(c^*)$.
In Game_4, if $c^* \leftarrow \mathcal{S}(1^\kappa)$ is not in $\mathsf{Enc}(ek, \mathcal{M})$, then the adversary has no information about $K^* = \mathsf{H}_q(c^*)$ and thus, K^* looks uniformly at random. Hence, the difference between two games Game_4 and Game_5 is bounded by the statistical disjointness in disjoint simulatability as in [XY19, Lemma 4.8]. We have

$$|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{Disj}_{\mathsf{PKE}, \mathcal{S}}(\kappa).$$

$\text{Game}_{5.1}$: This game is the same as Game_5 except that G is chosen from $\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}(\mathcal{M}, \mathcal{R})$. We have

$$|\Pr[S_5] - \Pr[S_{5.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}})^2 \text{Exp}_{(ek,dk) \leftarrow \mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \delta.$$

(We note that H and the challenge ciphertext also query to G internally.)
According to Lemma A.1, for any p, we have

$$|\Pr[S_{5.1} \wedge \neg\text{Bad}] - p| \leq |\Pr[S_{5.1}] - p| + \delta.$$

Game_6: This game is the same as Game_5 except that the decapsulation oracle is reset as $\textsc{Dec}$. Similar to the case for Game_2 and Game_3, if a key pair is accurate, the two games Game_5 and Game_6 are equivalent as in the proof of [XY19, Lemma 4.5]. We have

$$\Pr[S_{5.1} \wedge \neg\text{Bad}] = \Pr[S_6 \wedge \neg\text{Bad}].$$

$\text{Game}_{6.1}$: This game is the same as Game_6 except that the random oracle H is simulated by $\mathsf{H}'_q \circ g$ where $\mathsf{H}'_q \colon C \to \mathcal{K}$ is yet another random oracle as in $\text{Game}_{1.2}$. If a key pair is not bad, the two games Game_6 and $\text{Game}_{6.1}$ are equivalent as in the proof of [XY19, Lemma 4.4]. We have

$$\Pr[S_6 \wedge \neg\text{Bad}] = \Pr[S_{6.1} \wedge \neg\text{Bad}].$$

$\text{Game}_{6.2}$: This game is the same as $\text{Game}_{6.1}$ except that the random oracle $\mathsf{H}(\cdot)$ is set as the original. If a key pair is not bad, the two games $\text{Game}_{6.1}$ and $\text{Game}_{6.2}$ are equivalent as in the proof of [XY19, Lemma 4.4]. We have

$$\Pr[S_{6.1} \wedge \neg\text{Bad}] = \Pr[S_{6.2} \wedge \neg\text{Bad}].$$

We have, for any p,

$$|\Pr[S_{6.2} \wedge \neg\text{Bad}] - p| \leq |\Pr[S_{6.2}] - p| + \delta$$

from Lemma A.1.

Table 5. Summary of Games for the Proof of Theorem 5.3: '$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$' implies that the challenger generates $c^* \leftarrow \mathcal{S}(1^\kappa)$ and returns $\perp$ if $c^* \in \mathsf{Enc}(ek, \mathcal{M})$.

Game	H	c^*	K^*	Decryption valid c	Decryption invalid c	justification
Game_0	H	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c)$	
Game_1	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c)$	statistical disjointness
Game_2	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	Lemma 2.2
Game_3	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}_q(c^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c^*)$ is hidden
Game_4	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}_{\mathrm{prf}}(s, c^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c)$	Lemma 2.2
Game_5	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\overline{\mathsf{Dec}}(dk, c^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c)$	re-encryption check
Game_6	H	$\mathcal{S}(1^\kappa)$	$\overline{\mathsf{Dec}}(dk, c^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c)$	statistical disjointness

Game_7*:* This game is the same as $\mathrm{Game}_{6.2}$ except that the random oracle G is chosen from $\mathcal{F}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}_{\mathrm{good}}(\mathcal{M}, \mathcal{R})$. We have,

$$|\Pr[S_{6.2}] - \Pr[S_7]| \le 8(q_{\mathsf{G}} + q_{\mathrm{DEC}})^2\delta. \le 8(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta.$$

Game_8*:* This game is the same as Game_7 except that $\mathsf{H}_q(c)$ in the decapsulation is replaced by $\mathsf{H}_{\mathrm{prf}}(s, c)$. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_7] - \Pr[S_8]| \le 2(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

We note that This game is the original game $\mathrm{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$. Thus, we have

$$\Pr[S_8] = \Pr[\mathrm{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing those (in)equalities, we obtain the following bound:

$$\begin{aligned}\mathrm{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_8]| \\ &\le \mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathrm{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) \\ &\quad + 4\delta + 16(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta + 16(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2\delta + 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.\end{aligned}$$

Proof of Theorem 5.2: The proof of Theorem 5.2 is a simplified version of that of Theorem 5.1, since it does not require to consider G. Ignoring the transition between real G with good G, we obtain the bound as follows:

$$\begin{aligned}\mathrm{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_8]| \\ &\le 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2} + 4\delta + \mathrm{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{A}_{34},\mathcal{S}}(\kappa) + \mathrm{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).\end{aligned}$$

5.2 SSMT-CCA Security

Theorem 5.3. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathrm{prf}}]$ *is* SSMT-CCA*-secure.*
Formally speaking, for any adversary $\mathcal{A}$ against SSMT-CCA *security of* KEM *issuing at most $q_{\mathsf{H}_{\mathrm{prf}}}$ and q_{DEC} queries to $\mathsf{H}_{\mathrm{prf}}$ and* DEC*, we have*

$$\mathrm{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) \le 2\mathrm{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

We note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof Sketch: We use the game-hopping proof. We consider Game_i for $i = 0, \ldots, 6$. We summarize the games in Table 5. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i.

Game_0: This game is the original game $\mathsf{Expt}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$(c^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}.$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1: In this game, the ciphertext is set as $\perp$ if c^* is in $\mathsf{Enc}(ek, \mathcal{M})$. The difference between two games Game_0 and Game_1 is bounded by statistical disjointness.

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Game_2: This game is the same as Game_1 except that $\mathsf{H}_{\mathsf{prf}}(s, c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q : C \to \mathcal{K}$ is another random oracle.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_1] - \Pr[S_2]| \leq 2(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

Game_3: This game is the same as Game_2 except that $K^* := \mathsf{H}_q(c^*)$ instead of chosen random. Since c^* is always outside of $\mathsf{Enc}(ek, \mathcal{M})$, $\mathcal{A}$ cannot obtain any information about $\mathsf{H}_q(c^*)$. Hence, the two games Game_2 and Game_3 are equivalent and we have

$$\Pr[S_2] = \Pr[S_3].$$

Game_4: This game is the same as Game_3 except that $\mathsf{H}_q(\cdot)$ is replaced by $\mathsf{H}_{\mathsf{prf}}(s, \cdot)$. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_3] - \Pr[S_4]| \leq 2(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

Game_5: This game is the same as Game_4 except that $K^* := \overline{\mathsf{Dec}}(dk, c^*)$ instead of $\mathsf{H}_{\mathsf{prf}}(s, c^*)$. Recall that c^* is always in *outside* of $\mathsf{Enc}(ek, \mathcal{M})$. Thus, we always have $\mathsf{Dec}(c^*) = \perp$ or $\mathsf{Enc}(ek, \mathsf{Dec}(c^*)) \neq c^*$ and, thus, $K^* = \mathsf{H}_{\mathsf{prf}}(s, c^*)$. Hence, the two games are equivalent. We have

$$\Pr[S_4] = \Pr[S_5].$$

Game_6: We finally replace how to compute c^*. In this game, the ciphertext is chosen by $\mathcal{S}(1^\kappa)$ as in Game_0. The difference between two games Game_5 and Game_6 is bounded by statistical disjointness.

$$|\Pr[S_5] - \Pr[S_6]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Moreover, this game Game_6 is the original game $\mathsf{Expt}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)$ with $b = 1$.

$$\Pr[S_6] = \Pr[\mathsf{Expt}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain Theorem 5.3:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_6]| \\ &\leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.\end{aligned}$$

5.3 SCFR-CCA Security

Theorem 5.4. *If* PKE *is* XCFR-*secure or* SCFR-CCA-*secure, then* $\mathsf{KEM} = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Proof. Suppose that an adversary outputs a ciphertext c which is decapsulated into $K \neq \perp$ by both $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c) \neq \perp$. Let us define $\mu'_i = \mathsf{Dec}(dk_i, c)$ for $i \in \{0, 1\}$. We also define $\mu_i := \mu'_i$ if $c = \mathsf{Enc}(ek_i, \mu'_i)$ and $\perp$ otherwise.
We have five cases defined as follows:

1. Case 1 ($\mu_0 = \mu_1 \neq \bot$): This violates XCFR-security of SCFR-CCA-security of the underlying PKE and it is easy to make a reduction.
2. Case 2 ($\bot \neq \mu_0 \neq \mu_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0) = \mathsf{H}(\mu_1)$. Thus, we succeed to find a collision for H, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 3 ($\mu_0 = \bot$ and $\mu_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}_{\mathsf{prf}}(s_0, c) = \mathsf{H}(\mu_1)$ and we find a claw $((s_0, c), \mu_1)$ of $\mathsf{H}_{\mathsf{prf}}$ and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
4. Case 4 ($\mu_0 \neq \bot$ and $\mu_1 = \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0) = \mathsf{H}_{\mathsf{prf}}(s_1, c)$ and In this case, we find a claw $(\mu_0, (s_1, c))$ of H and $\mathsf{H}_{\mathsf{prf}}$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 5 (The other cases): In this case, we find a collision $((s_0, c), (s_1, c))$ of $\mathsf{H}_{\mathsf{prf}}$, which is indeed collision if $s_0 \neq s_1$ which occurs with probability at lease $1 - 1/2^{\ell}$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

6 NTRU

We briefly review NTRU [CDH+20] and discuss its security properties.

6.1 Review of NTRU

Preliminaries: Φ_1 denotes the polynomial $x - 1$ and Φ_n denotes $(x^n - 1)/(x - 1) = x^{n-1} + x^{n-2} + \cdots + 1$. We have $x^n - 1 = \Phi_1 \Phi_n$. R, $R/3$, and R/q denotes $\mathbb{Z}[x]/(\Phi_1\Phi_n)$, $\mathbb{Z}[x]/(3, \Phi_1\Phi_n)$, and $\mathbb{Z}[x]/(q, \Phi_1\Phi_n)$, respectively. S, $S/3$, and S/q denotes $\mathbb{Z}[x]/(\Phi_n)$, $\mathbb{Z}[x]/(3, \Phi_n)$, and $\mathbb{Z}[x]/(q, \Phi_n)$, respectively.
We say a polynomial *ternary* if its coefficients are in $\{-1, 0, +1\}$. $\underline{\mathsf{S3}}(a)$ returns a canonical $S/3$-representative of $z \in \mathbb{Z}[x]$, that is, $b \in \mathbb{Z}[x]$ of degree at most $n - 2$ with ternary coefficients in $\{-1, 0, +1\}$ such that $a \equiv b \pmod{(3, \Phi_n)}$. Let $\mathcal{T}$ be a set of non-zero ternary polynomials of degree at most $n-2$, that is, $\mathcal{T} = \{a = \sum_{i=0}^{n-2} a_i x^i : a \neq 0 \wedge a_i \in \{-1, 0, +1\}\}$. We say a ternary polynomial $v = \sum_i v_i x^i$ has the *non-negative correlation* property if $\sum_i v_i v_{i+1} \geq 0$. $\mathcal{T}_+$ is a set of non-zero ternary polynomials of degree at most $n - 2$ with *non-negative correlation* property. $\mathcal{T}(d)$ is a set of non-zero balanced ternary polynomials of degree at most $n - 2$ with Hamming weight d, that is, $\{a \in \mathcal{T} : |\{a_i : a_i = 1\}| = |\{a_i : a_i = -1\}| = d/2\}$.
The following lemma is due to Schanck [Sch21]. (See, e.g., [CDH+20] for this design choice.)

Lemma 6.1. *Suppose that $(n, q) = (509, 2048), (677, 2048), (821, 4096)$, and $(701, 8192)$. If $r \in \mathcal{T}$, then r has an inverse in S/q.*

Proof. Φ_n is irreducible over $\mathbb{F}_2$ if and only if n is prime and 2 is primitive element in $\mathbb{F}_n^{\times}$ (See e.g., Cohen et al. [CFA05]). The conditions are satisfied by all $n = 509, 677, 701$, and 821. Hence, $\mathbb{Z}[x]/(2, \Phi_n)$ is a finite field and every polynomial r in $\mathcal{T}$ has an inverse in $\mathbb{Z}[x]/(2, \Phi_n)$. Such r is also invertible in $S/q = \mathbb{Z}[x]/(q, \Phi_n)$ with $q = 2^k$ for some k. One can find it using the Newton method/the Hensel lifting. □

NTRU: NTRU has two types of parameter sets, NTRU-HPS and NTRU-HRSS. The underlying DPKE of NTRU, which we call NTRU-DPKE, is define as Figure 6. It involves four subsets $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m$ of R. It uses $\mathsf{Lift}(m) \colon \mathcal{L}_m \to R$.

- NTRU-HPS: The parameters are defined as follows: $\mathcal{L}_f = \mathcal{T}, \mathcal{L}_g = \mathcal{T}(q/8 - 2), \mathcal{L}_r = \mathcal{T}, \mathcal{L}_m = \mathcal{T}(q/8 - 2)$, and $\mathsf{Lift}(m) = m$.
- NTRU-HRSS: The parameters are defined as follows: $\mathcal{L}_f = \mathcal{T}_+, \mathcal{L}_g = \{\Phi_1 \cdot v \mid v \in \mathcal{T}_+\}, \mathcal{L}_r = \mathcal{T}, \mathcal{L}_m = \mathcal{T}$, and $\mathsf{Lift}(m) = \Phi_1 \cdot \underline{\mathsf{S3}}(m/\Phi_1)$.

It uses Sample_fg() to sample f and g from $\mathcal{L}_f$ and $\mathcal{L}_g$. NTRU also uses Sample_rm() to sample r and m from $\mathcal{L}_r$ and $\mathcal{L}_m$.
We note that $h \equiv 0 \pmod{(q, \Phi_1)}$, h is invertible in S/q, and $hr + m \equiv 0 \pmod{(q, \Phi_1)}$. (See [CDH+20, Section2.3].)
NTRU then uses SXY for IND-CCA-secure KEM as in Figure 7, where $\mathsf{H} = \text{SHA3-256}$ and $\mathsf{H}_{\mathsf{prf}} = \text{SHA3-256}$. Since the lengths of their input space differ, we can treat them as different random oracles.

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Enc}(h, (r, m) \in \mathcal{L}_r \times \mathcal{L}_m)$	$\mathsf{Dec}((f, f_p, h_q), c)$
$(f, g) \leftarrow \mathsf{Sample\_fg}()$	$\mu' \leftarrow \mathsf{Lift}(m)$	**if** $c \not\equiv 0 \bmod (q, \Phi_1)$ **then return** $(0, 0, 1)$
$f_q \leftarrow (1/f) \bmod (q, \Phi_n)$	$c \leftarrow (h \cdot r + \mu') \bmod (q, \Phi_1\Phi_n)$	$a \leftarrow (c \cdot f) \bmod (q, \Phi_1\Phi_n)$
$h \leftarrow (3 \cdot g \cdot f_q) \bmod (q, \Phi_1\Phi_n)$	**return** c	$m \leftarrow (a \cdot f_p) \bmod (3, \Phi_n)$
$h_q \leftarrow (1/h) \bmod (q, \Phi_n)$		$\mu' \leftarrow \mathsf{Lift}(m)$
$f_p \leftarrow (1/f) \bmod (3, \Phi_n)$		$r \leftarrow ((c - \mu') \cdot h_q) \bmod (q, \Phi_n)$
$ek := h$, $dk := (f, f_p, h_q)$		**if** $(r, m) \in \mathcal{L}_r \times \mathcal{L}_m$ **then return** $(r, m, 0)$
return (ek, dk)		**else return** $(0, 0, 1)$

Fig. 6. NTRU-DPKE

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek = h)$	$\overline{\mathsf{Dec}}(\overline{dk} = (dk, s), c)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mathsf{coins} \leftarrow \{0,1\}^{256}$	$(r, m, \mathsf{fail}) := \mathsf{Dec}(dk, c)$
$s \leftarrow \{0,1\}^{256}$	$(r, m) \leftarrow \mathsf{Sample\_rm}(\mathsf{coins})$	$k_1 := \mathsf{H}(r, m)$
$\overline{dk} := (dk, s)$	$c := \mathsf{Enc}(h, (r, m))$	$k_2 := \mathsf{H}_{\mathsf{prf}}(s, c)$
return $(ek, \overline{dk})$	$K := \mathsf{H}(r, m)$	**if** $\mathsf{fail} = 0$ **then return** k_1
	return (c, K)	**else return** k_2

Fig. 7. NTRU

Rigidity: NTRU uses SXY, while its KEM version seems lack of re-encryption check. We note that NTRU implicitly checks $hr + \mathsf{Lift}(m) = c$ by checking if $(r, m) \in \mathcal{L}_r \times \mathcal{L}_r$ in the DPKE. See [CDH+20] for the details.

6.2 NTRU is Strongly Pseudorandom, Smooth, and Collision-Free

We have known that the generalized NTRU PKE is pseudorandom [SS10] and disjointly simulatable [SXY18] if the decisional small polynomial ratio (DSPR) assumption [LTV12] and the polynomial learning with errors (PLWE) assumption [SSTX09, LPR10] hold. See [SXY18, Section 3.3 of the ePrint version.].
Let us adapt their arguments to NTRU. We modify the DSPR and the PLWE assumptions as follows:

Definition 6.1. *Fix the parameter set. Define $R' := \{c \in R/q : c \equiv 0 \pmod{(q, \Phi_1)}\}$, which is efficiently sampleable.*

- *The modified DSPR assumption: It is hard to distinguish $h := 3 \cdot g \cdot f_q \pmod{q, \Phi_1\Phi_n}$ from u, where $(f, g) \leftarrow \mathsf{Sample\_fg}()$ and $u \leftarrow R'$.*
- *The modified PLWE assumption: It is hard to distinguish $(h, hr + \mathsf{Lift}(m) \pmod{q, \Phi_1\Phi_n})$ from (h, c) with $h, c \leftarrow R'$ and $(r, m) \leftarrow \mathsf{Sample\_rm}()$.*

Lemma 6.2. *Suppose that the modified DSPR and PLWE assumptions hold. Then, NTRU-DPKE is strongly disjoint-simulatable with a simulator $\mathcal{S}$ that outputs a random polynomial chosen from R'.*

Proof (Proof Sketch). The proof for ciphertext-indistinguishability is obtained by modifying the proof in [SXY18], Statistical disjointness follows from the fact that $|R'| = q^{n-1} \gg 3^{2n} = |\mathcal{T} \times \mathcal{T}| \geq |\mathcal{L}_m \times \mathcal{L}_r| \geq |\mathsf{Enc}(h, \mathcal{L}_m \times \mathcal{L}_r)|$.

Combining this strong disjoint-simulatability with previous theorems, we obtain the following theorem.

Theorem 6.1. *Suppose that the modified DSPR and PLWE assumptions hold. Then, NTRU is* SPR-CCA*-secure and* SSMT-CCA*-secure in the QROM.*

Theorem 6.2. *NTRU is* SCFR-CCA*-secure in the QROM.*

Proof. We first show XCFR security of NTRU-DPKE.
Suppose that the adversary outputs c on input $ek_0 = h_0, dk_0, ek_1 = h_1, dk_1$. Let us define $\mu_0 = \mathrm{Dec}(dk_0, c)$ and $\mu_1 = \mathrm{Dec}(dk_1, c)$. Let $\mu_0 = \mu_1 = (r, m, 0) \in \mathcal{L}_r \times \mathcal{L}_m \times \{0, 1\}$. Otherwise, that is, if $\mu_0 = \mu_1 = (0, 0, 1)$, the output is treated as $\perp$ and the adversary loses.
We have $h_0 \cdot r + \mathsf{Lift}(m) \equiv h_1 \cdot r + \mathsf{Lift}(m) \pmod{q, \Phi_1\Phi_n}$, which implies $r(h_0 - h_1) \equiv 0 \pmod{(q, \Phi_n)}$. On the other hand, according to Lemma 6.1, for any $r \in \mathcal{L}_r = \mathcal{T}$, we have $r \neq 0 \in S/q$ In addition, we have $h_0 \equiv h_1 \in S/q$ with negligible probability. Thus, the probability that the adversary wins is negligible.
Applying Theorem 5.4, we conclude that NTRU is SCFR-CCA-secure in the QROM. ◻

6.3 Summary

We show that NTRU-DPKE is strongly disjoint-simulatable under the modified DSPR and PLWE assumptions and it is XCFR-secure (subsection 6.2). Those imply that NTRU is SPR-CCA-secure, SSMT-CCA-secure, and SCFR-CCA-secure in the QROM. Thus, NTRU is ANON-CCA-secure (Theorem 3.1) and NTRU leads to ANON-CCA-secure, SROB-CCA-secure hybrid PKE (Theorem 4.2, Theorem 3.1, and Theorem 2.2).

Acknowledgement

The author is grateful to John Schanck for insightful comments and suggestions on NTRU and Akinori Hosoyamada and Takashi Yamakawa for insightful comments and discussion on quantum random oracles. The author would like to thank Daniel J. Bernstein for insightful comments and discussion on the nested random oracles.

References

AAB+20. Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and Jurjen Bos. HQC. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 5, 32, 34, 59, 60, 61

ABB+20. Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loic Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Phillipe Gaborit, Shay Gueron, Tim Guneysu, Carlos Aguilar Melchor, Rafael Misoczki, Edoardo Persichetti, Nicolas Sendrier, Jean-Pierre Tillich, Gilles Zémor, Valentin Vasseur, and Santosh Ghosh. BIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 5, 30, 34, 55, 56, 57

ABC+05. Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 205–222. Springer, Heidelberg, August 2005. 3

ABC+20. Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang. Classic McEliece. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 3, 5, 33, 34, 52

Abe10. Masayuki Abe, editor. *ASIACRYPT 2010*, volume 6477 of *LNCS*. Springer, Heidelberg, December 2010. 26, 27

ABN10. Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 480–497. Springer, Heidelberg, February 2010. 2

BBC+20. Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 5, 32, 33, 34, 62, 63, 64

BBDP01. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, Heidelberg, December 2001. 2, 6

BCGNP09. Colin Boyd, Yvonne Cliff, Juan Manuel González Nieto, and Kenneth G. Paterson. One-round key exchange in the standard model. *Int. J. Appl. Cryptogr.*, 1(3):181–199, 2009. 3

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, December 2011. 5

BDPR98. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 26–45. Springer, Heidelberg, August 1998. 6, 9

Ber21. Daniel J. Bernstein. personal communication, October 2021. 4, 63

BHH$^+$19. Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 61–90. Springer, Heidelberg, December 2019. 29, 30, 32, 35, 44, 45, 48, 49

CDH$^+$20. Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 3, 5, 22, 23, 33, 34

CFA05. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. 2005. 22

CL01. Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, Heidelberg, May 2001. 3

CS02. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002. 3, 9

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003. 14, 15, 16

DKR$^+$20. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 3, 5, 31, 34, 55

FNP14. Nelly Fazio, Antonio Nicolosi, and Irippuge Milinda Perera. Broadcast steganography. In Josh Benaloh, editor, *CT-RSA 2014*, volume 8366 of *LNCS*, pages 64–84. Springer, Heidelberg, February 2014. 66

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, August 1999. 3

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

FOR17. Pooya Farshim, Claudio Orlandi, and Răzvan Roşie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symm. Cryptol.*, 2017(1):449–473, 2017. 9

FSXY13. Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Practical and post-quantum authenticated key exchange from one-way secure key encapsulation mechanism. In Kefei Chen, Qi Xie, Weidong Qiu, Ninghui Li, and Wen-Guey Tzeng, editors, *ASIACCS 13*, pages 83–94. ACM Press, May 2013. 3

FSXY15. Atsushi Fujioka, Koutarou Suzuki, Keita Xagawa, and Kazuki Yoneyama. Strongly secure authenticated key exchange from factoring, codes, and lattices. *Des. Codes Cryptogr.*, 76(3):469–504, 2015. 3

GMP21a. Paul Grubbs, Varun Maram, and Kenneth G. Paterson. Anonymous, robust post-quantum public key encryption. Cryptology ePrint Archive, Report 2021/708, 2021. https://eprint.iacr.org/2021/708. 2, 3, 4, 9, 11, 31, 52, 53, 54, 55, 56, 59, 60, 63

GMP21b. Paul Grubbs, Varun Maram, and Kenneth G. Paterson. Anonymous, robust post-quantum public key encryption (presentation slides). The third NIST PQC Standardization Conference, 2021. https://csrc.nist.gov/Presentations/2021/anonymous-robust-post-quantum-public-key-encryptio. 3, 4

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017. 4, 6, 17, 18, 28, 29, 32, 54

HKSU20. Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 389–422. Springer, Heidelberg, May 2020. 5, 7, 18

Hop05. Nicholas Hopper. On steganographic chosen covertext security. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *ICALP 2005*, volume 3580 of *LNCS*, pages 311–323. Springer, Heidelberg, July 2005. 6, 9

Hos21. Akinori Hosoyamada. personal communication, June 2021. 6

IZ89. Russell Impagliazzo and David Zuckerman. How to recycle random bits. In *30th FOCS*, pages 248–253. IEEE Computer Society Press, October / November 1989. 66

JAC+20. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson. SIKE. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 5, 30, 34, 65

JD11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011. 65, 66

JZC+18. Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 96–125. Springer, Heidelberg, August 2018. 5, 11, 18, 28, 32, 50

JZM19. Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 618–645. Springer, Heidelberg, April 2019. 4, 29, 32, 33, 41, 43

KSS+20. Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 703–728. Springer, Heidelberg, May 2020. 30, 32

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010. 23

LTV12. Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 1219–1234. ACM Press, May 2012. 23

LW21. Xu Liu and Mingqiang Wang. QCCA-secure generic key encapsulation mechanism with tighter security in the quantum random oracle model. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 3–26. Springer, Heidelberg, May 2021. 7, 17, 18, 37, 41

Moh10. Payman Mohassel. A closer look at anonymity and robustness in encryption schemes. In Abe [Abe10], pages 501–518. 3, 6, 11

MTSB13. Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proceedings of the 2013 IEEE International Symposium on Information Theory (ISIT), Istanbul, Turkey, July 7-12, 2013*, pages 2069–2073. IEEE, 2013. 55

NAB+20. Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 5, 31, 34, 59

RS92. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992. 6, 9

SAB+20. Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions. 3, 5, 31, 34, 54

Sak00. Kazue Sako. An auction protocol which hides bids of losers. In Hideki Imai and Yuliang Zheng, editors, *PKC 2000*, volume 1751 of *LNCS*, pages 422–432. Springer, Heidelberg, January 2000. 3

Sch21. John Schanck. personal communication, June 2021. 22

SS10. Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In Abe [Abe10], pages 377–394. 23

SSTX09. Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, December 2009. 23

SSW20. Peter Schwabe, Douglas Stebila, and Thom Wiggers. Post-quantum TLS without handshake signatures. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1461–1480. ACM Press, November 2020. 3

SXY18. Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018. 4, 5, 7, 17, 23, 28, 29, 33, 57

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, October / November 2016. 32, 54

Unr14. Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 129–146. Springer, Heidelberg, May 2014. 28

vH04. Luis von Ahn and Nicholas J. Hopper. Public-key steganography. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 323–341. Springer, Heidelberg, May 2004. 6, 9

XY19. Keita Xagawa and Takashi Yamakawa. (Tightly) QCCA-secure key-encapsulation mechanism in the quantum random oracle model. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 249–268. Springer, Heidelberg, 2019. 6, 17, 18, 19, 20, 21, 36, 37, 38, 39, 40, 41, 46, 50

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7–8):557–567, May 2015. 5

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019. 4, 63

A Missing Lemmas

Lemma A.1. *Let* A *and* B *denote events. Suppose that we have* $\Pr[\mathsf{A}] \leq \delta$. *For any* $p \geq 0$, *we have*

$$|\Pr[\mathsf{B}] - p| \leq |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| + \delta \quad \textit{and} \quad |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| \leq |\Pr[\mathsf{B}] - p| + \delta.$$

Proof. We have

$$\begin{aligned}
|\Pr[\mathsf{B}] - p| &= |\Pr[\mathsf{B} \wedge \mathsf{A}] + \Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| \\
&\leq \Pr[\mathsf{B} \wedge \mathsf{A}] + |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| \\
&\leq \Pr[\mathsf{A}] + |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| \\
&\leq |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| + \delta.
\end{aligned}$$

We also have

$$\begin{aligned}
|\Pr[\mathsf{B} \wedge \neg\mathsf{A}] - p| &= |\Pr[\mathsf{B} \wedge \neg\mathsf{A}] + \Pr[\mathsf{B} \wedge \mathsf{A}] - \Pr[\mathsf{B} \wedge \mathsf{A}] - p| \\
&= |\Pr[\mathsf{B}] - p - \Pr[\mathsf{B} \wedge \mathsf{A}]| \\
&\leq |\Pr[\mathsf{B}] - p| + \Pr[\mathsf{B} \wedge \mathsf{A}] \\
&\leq |\Pr[\mathsf{B}] - p| + \Pr[\mathsf{A}] \\
&\leq |\Pr[\mathsf{B}] - p| + \delta
\end{aligned}$$

Those complete the proof. $\square$

The lemma of the following form is a slightly generalized version of the O2H lemma taken from [SXY18, Lemma 2.1]. While there are improvements of the O2H lemma, this basic O2H lemma is enough for our cases.

Lemma A.2 (The Oneway-to-Hiding (O2H) Lemma [Unr14, HHK17, JZC⁺18, SXY18]). *Let $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$ be a quantum random oracle, and 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 $0/1$. Let $\mathcal{D}_{\mathcal{X}}$ be a some distribution over $\mathcal{X}$. 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*

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

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 fewer than i queries, EXT outputs $\perp \notin \mathcal{X}$).

B Missing Proofs

B.1 Proof of Theorem 3.1

Proof (Proof of Theorem 3.1). Let us define four games $\mathsf{Game}_{i,b}$ for $i, b \in \{0, 1\}$. Let $S_{i,b}$ be the event that the adversary outputs 1 in $\mathsf{Game}_{i,b}$.

- $\mathsf{Game}_{0,b}$ for $b \in \{0, 1\}$: This is the original game $\mathsf{Expt}^{\text{anon-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$ with $b = 0$ and 1.
- $\mathsf{Game}_{1,b}$ for $b \in \{0, 1\}$: This game is the same as $\mathsf{Game}_{0,b}$ except that the target ciphertext is randomly taken from $\mathcal{S}(1^\kappa) \times \mathcal{C}_{\mathsf{DEM},|m|}$.

It is easy to see that there exist two adversaries $\mathcal{A}_{10}$ and $\mathcal{A}_{11}$ whose running times are the same as that of $\mathcal{A}$ satisfying

$$\left|\Pr[S_{0,b}] - \Pr[S_{1,b}]\right| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}_{1b}}(\kappa) \text{ and } \Pr[S_{1,0}] = \Pr[S_{1,1}].$$

Hence, we have

$$\begin{aligned}
\mathsf{Adv}^{\text{anon-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) &= \left|\Pr[S_{0,0}] - \Pr[S_{0,1}]\right| \\
&\leq \left|\Pr[S_{0,0}] - \Pr[S_{1,0}]\right| + \left|\Pr[S_{1,0}] - \Pr[S_{1,1}]\right| + \left|\Pr[S_{1,1}] - \Pr[S_{0,1}]\right| \\
&\leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}_{10}}(\kappa) + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{S},\mathcal{A}_{11}}(\kappa).
\end{aligned}$$

This completes the proof. ◻

C Variants of the Fujisaki-Okamoto Transformation

We review the variants of the FO transformations: Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE, whose ciphertext space is $\mathcal{C}_{\mathsf{PKE}}$ and message space is $\mathcal{M}$. If PKE is probabilistic, then $\mathcal{R}_{\mathsf{Enc}}$ denotes the randomness space of Enc. Let $\{0, 1\}^{k(\kappa)}$ be the key space of KEM.

C.1 Transformation T

Hofheinz et al. [HHK17] decomposed the Fujisaki-Okamoto transformation FO into two transformations T and U. In the original T in [HHK17, Section 3.1], the decryption algorithm checks the validity of c by re-encryption check. We omit this re-encryption check. Our version is summarized in Figure 8.

$\mathsf{Gen}'(1^\kappa)$	$\mathsf{Enc}'(ek, \mu)$	$\mathsf{Dec}'(dk, c)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c)$
return $(ek, \overline{dk})$	$c := \mathsf{Enc}(ek, \mu; \mathsf{G}(\mu))$	**return** μ'
	return c	

Fig. 8. $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$

C.2 Variants of U

Hofheinz et al. defined U's variants, $\mathsf{U}^{\not\perp}$, $\mathsf{U}^{\perp}$, $\mathsf{U}^{\not\perp}_m$, and $\mathsf{U}^{\perp}_m$ [HHK17], where the superscript "$\not\perp$" and "$\perp$" implies *implicit rejection* and *explicit rejection*, respectively, and the subscript "$_m$" implies the computation of key K involves a plaintext μ only, while if there is no subscript, then it involves μ and ciphertext c.

Saito et al. define SXY, which is essentially the same as $\mathsf{U}^{\not\perp}_m$ [SXY18]. Bindel et al. discussed the relations of IND-CCA-security of KEM schemes obtained by them via indifferentiable reductions [BHH⁺19]. In their discussion, they modify $\mathsf{U}^{\not\perp}$, which we write $\mathsf{U}^{\not\perp,\mathsf{prf}}$. They use $K := \mathsf{H}_{\mathsf{prf}}(s, c)$ for invalid ciphertext c instead of $K := \mathsf{H}(s, c)$ as in [HHK17].

Let us review the definitions.

- $\mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$: This is defined in Figure 9.
- $\mathsf{U}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$: The decapsulation returns $K := \mathsf{H}_{\mathsf{prf}}(s, c)$ if $\mu' = \perp$ or $c \neq \mathsf{Enc}(ek, \mu')$.
- $\mathsf{U}^{\perp}[\mathsf{PKE}, \mathsf{H}]$: The decapsulation returns $K := \perp$ if $\mu' = \perp$ or $c \neq \mathsf{Enc}(ek, \mu')$. This variants does not require s in $\overline{dk}$.
- $\mathsf{U}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$: The encapsulation defines $K := \mathsf{H}(\mu, c)$. The decapsulation returns $K := \mathsf{H}(\mu, c)$ if $\mu' \neq \perp$ and $c = \mathsf{Enc}(ek, \mu')$.
- $\mathsf{U}^{\perp}_m[\mathsf{PKE}, \mathsf{H}]$: The encapsulation defines $K := \mathsf{H}(\mu, c)$. The decapsulation returns $K := \mathsf{H}(\mu, c)$ if $\mu' \neq \perp$ and $c = \mathsf{Enc}(ek, \mu')$. The decapsulation returns $K := \perp$ if $\mu' = \perp$ or $c \neq \mathsf{Enc}(ek, \mu')$. This variants does not require s in $\overline{dk}$.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c)$
$s \leftarrow \mathcal{M}$	$c := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \perp$ or $c \neq \mathsf{Enc}(ek, \mu')$
$\overline{dk} := (dk, ek, s)$	$K := \mathsf{H}(\mu, c)$	**then return** $K := \mathsf{H}(s, c)$
return $(ek, \overline{dk})$	**return** (c, K)	**else return** $K := \mathsf{H}(\mu', c)$

Fig. 9. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$

We adapt the discussions of Bindel et al. to SPR-CCA-security of KEM schemes obtained by the variants of U. See the left hand side of Figure 10.

C.3 Variants of HU

Hofheinz et al. defined QU's variants, $\mathsf{QU}^{\not\perp}_m$ and $\mathsf{QU}^{\perp}_m$ [HHK17]. In those variants a ciphertext includes 'key-confirmation' hash $d := \mathsf{F}(\mu)$, where $\mathsf{F} \colon \mathcal{M} \to \mathcal{M}$. (For the proof, We will require $\mathcal{M}$ to be a subset of a finite field.) Jiang et al. [JZM19] defined $\mathsf{HU}^{\perp}_m$ as a variant of $\mathsf{QU}^{\perp}_m$, where $\mathsf{F} \colon \mathcal{M} \to \mathcal{H}$ with arbitrary $\mathcal{M}$ and $\mathcal{H}$. This allows us to make a ciphertext shorter. We define its variants $\mathsf{HU}^{\perp}_m$, $\mathsf{HU}^{\not\perp}_m$, $\mathsf{HU}^{\perp}$, $\mathsf{HU}^{\not\perp}_m$, and $\mathsf{HU}^{\not\perp,\mathsf{prf}}$ as the variants of U. In the definition, we allow F to take ek optional.

Let us review the definitions.

- $\mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$: This is defined in Figure 11.

$$
\begin{tikzcd}
 & \mathsf{SXY} \arrow[d, phantom, "\parallel"] \\
\mathsf{U}^{\perp}_{m} \arrow[r, dashed, "{\text{[BHH$^+$19, Thm. 3]}}"] \arrow[dd, dashed, leftrightarrow, "{\text{[BHH$^+$19, Thm. 5]}}"'] & \mathsf{U}^{\not\perp}_{m} \arrow[d, dashed, leftrightarrow, "{\text{[BHH$^+$19, Thm. 5]}}"] \\
 & \mathsf{U}^{\not\perp,\mathsf{prf}} \\
\mathsf{U}^{\perp} \arrow[r, dashed, "{\text{[BHH$^+$19, Thm. 5, adapted]}}"'] & \mathsf{U}^{\not\perp}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{HU}^{\perp}_{m} \arrow[r, "\text{Thm. I.1}"] \arrow[dd, leftrightarrow, "\text{Thm. H.3}"] & \mathsf{HU}^{\not\perp}_{m} \arrow[d, leftrightarrow, "\text{Thm. J.3}"] \\
 & \mathsf{HU}^{\not\perp,\mathsf{prf}} \\
\mathsf{HU}^{\perp} \arrow[r, "\text{Thm. K.3}"] & \mathsf{HU}^{\not\perp}
\end{tikzcd}
$$

Fig. 10. The relation between IND-CCA and SPR-CCA security of KEMs using the variants of U and HU. Dashed arrow implies the implications in [BHH+19].

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c_0)$
$s \leftarrow \mathcal{M}$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \bot$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \mathsf{H}(s, c_0, c_1)$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c_0, c_1)$	**else return** $K := \mathsf{H}(\mu', c_0, c_1)$
	return $((c_0, c_1), K)$	

Fig. 11. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$

- $\mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$: The decapsulation returns $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$ if $\mu' = \bot$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$.
- $\mathsf{HU}^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$: The decapsulation returns $K := \bot$ if $\mu' = \bot$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$. This variants does not require s in $\overline{dk}$.
- $\mathsf{HU}^{\not\perp}_{m}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$: The encapsulation defines $K := \mathsf{H}(\mu, c_0, c_1)$. The decapsulation returns $K := \mathsf{H}(\mu, c_0, c_1)$ if $\mu' \neq \bot$ and $c_0 = \mathsf{Enc}(ek, \mu')$ and $c_1 = \mathsf{F}(\mu'[, ek])$.
- $\mathsf{HU}^{\perp}_{m}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$: The encapsulation defines $K := \mathsf{H}(\mu, c_0, c_1)$. The decapsulation returns $K := \mathsf{H}(\mu, c_0, c_1)$ if $\mu' \neq \bot$ and $c_0 = \mathsf{Enc}(ek, \mu')$ and $c_1 = \mathsf{F}(\mu'[, ek])$. The decapsulation returns $K := \bot$ if $\mu' = \bot$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$. This variants does not require s in $\overline{dk}$.

We will adapt the discussions of Bindel et al.to SPR-CCA-security of KEM schemes obtained by the variants of U. See the right hand side of Figure 10.

C.4 Variants of FO

Combining T and the variants of U or HU, we obtain several variants of FO as follows: Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a probabilistic PKE scheme: If we combine T and U^{y}_{x}, then we obtain FO^{y}_{x}. If we combine T and HU^{y}_{x}, then we obtain HFO^{y}_{x}.

D Transformation in NIST PQC KEM Candidates

In this section, we review the transformations used in NIST PQC Round 3 KEM Candidates.

D.1 FO with implicit rejection

$\mathsf{FO}^{\not\perp}$ transforms a weakly-secure probabilistic PKE into IND-CCA-secure KEM. This variant can be considered the composition of T and $\mathsf{U}^{\not\perp}$, that is, $\mathsf{KEM} = \mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}] = \mathsf{U}^{\not\perp}[\mathsf{T}[\mathsf{PKE}, \mathsf{G}], \mathsf{H}]$. This variant is used by BIKE [ABB+20] and SIKE [JAC+20].

Let $\{0,1\}^{\ell(\kappa)}$ be the plaintext space of PKE. Let $\mathsf{G}\colon \{0,1\}^* \to \mathcal{R}_{\mathsf{Enc}}$ and $\mathsf{H}\colon \{0,1\}^{\ell(\kappa)} \times C_{\mathsf{PKE}} \to \{0,1\}^{k(\kappa)}$ be hash functions modeled by the random oracles. The $\mathsf{FO}^{\not\perp}$ is summarized as Figure 12. Assuming the IND-CPA security of PKE, the obtained KEM scheme is IND-CCA-secure in the QROM (see e.g., [KSS+20]).

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$r := \mathsf{G}(\mu)$ // for BIKE	$r' := \mathsf{G}(\mu')$ // for BIKE
$\overline{dk} := (dk, ek, s)$	$r := \mathsf{G}(\mu, ek)$ // for SIKE	$r' := \mathsf{G}(\mu', ek)$ // for SIKE
return $(ek, \overline{dk})$	$c := \mathsf{Enc}(ek, \mu; r)$	$c' := \mathsf{Enc}(ek, \mu'; r')$
	$K := \mathsf{H}(\mu, c)$	**if** $c \neq c'$, **then return** $K := \mathsf{H}(s, c)$
	return (c, K)	**else return** $K := \mathsf{H}(\mu', c)$

Fig. 12. $\mathsf{KEM} := \mathsf{FO}^{\not\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}]$ for BIKE and SIKE.

Remark D.1. BIKE and SIKE do *not* test whole re-encryption check. Roughly speaking, their encryption algorithm Enc is separable into two algorithms Enc_1 and Enc_2. Enc_1 takes ek and randomness r and outputs c_1 and $k \in \{0,1\}^{\ell(\kappa)}$. Enc_2 takes μ and k and outputs $c_2 := k \oplus \mu$.
Using this property, BIKE omits the re-encryption check. Concretely speaking, k in BIKE's Enc_1 is computed as $k := \mathsf{H}(r)$, where H is a hash function modeled by the random oracle. BIKE's Dec internally obtains r' and checks the validity of c_1. It then retrieves $\mu' := c_2 \oplus \mathsf{H}(r')$ and checks the validity of the ciphertext by checking $r' = \mathsf{G}(\mu')$ or not.
SIKE's $\overline{\mathsf{Dec}}$ performs the test $c_1' = c_1$ but omits the test $c_2' = c_2$. Since Dec retrieves $\mu' := c_2 \oplus k$ *deterministically*, we do not need to check the equality of c_2 and c_2'.

D.2 Other FO with implicit rejection and pre-key

$\mathsf{FO}^{\not\perp\prime}$ is a modified version of $\mathsf{FO}^{\not\perp}$, which is used by Kyber [SAB+20, Section 1] and Saber [DKR+20, Section 8]. $\mathsf{FO}^{\not\perp\prime\prime}$ is another modified versions of $\mathsf{FO}^{\not\perp}$, which are used by FrodoKEM [NAB+20, Section 2]. The differences from $\mathsf{FO}^{\not\perp}$ are how to generate K in $\overline{\mathsf{Enc}}$ and $\overline{\mathsf{Dec}}$. They first compute 'pre-key' $\bar{K}$ from μ and $\mathsf{H}'(ek)$ and then compute key $K := \mathsf{H}(\bar{K}, \mathsf{H}'(c))$ in $\mathsf{FO}^{\not\perp\prime}$ or $\mathsf{H}(\bar{K}, c)$ in $\mathsf{FO}^{\not\perp\prime\prime}$.
Let $\{0,1\}^{\ell(\kappa)}$ be the plaintext space of PKE. Let $\mathsf{G}\colon \{0,1\}^* \to \{0,1\}^{\ell(\kappa)} \times \mathcal{R}_{\mathsf{Enc}}$, $\mathsf{H}'\colon \{0,1\}^* \to \{0,1\}^{\ell(\kappa)}$, and $\mathsf{H}\colon \{0,1\}^{\ell(\kappa)} \times \{0,1\}^{\ell(\kappa)} \to \{0,1\}^{k(\kappa)}$ be hash functions modeled by the random oracles. $\mathsf{FO}^{\not\perp\prime}$ and $\mathsf{FO}^{\not\perp\prime\prime}$ are summarized as Figure 13 and Figure 14, respectively.
One might consider assuming the IND-CPA security of PKE, the obtained KEM schemes are IND-CCA-secure in the QROM. Unfortunately, Grubbs, Maram, and Paterson [GMP21a] pointed out that we cannot directly apply the existing security proof in the QROM to those variants, because computing K requires nested applications of random oracles G and H to m. Grubbs et al. overcome this barrier for the case of $\mathsf{FO}^{\not\perp\prime\prime}$ in [GMP21a, Section 5.2]. Thus, FrodoKEM using $\mathsf{FO}^{\not\perp\prime\prime}$ can be shown IND-CCA-secure in the QROM. However, they failed to apply their technique to the case of $\mathsf{FO}^{\not\perp\prime}$ which computes $K = \mathsf{H}(\bar{K}, \mathsf{H}'(c))$ instead of $K = \mathsf{H}(\bar{K}, c)$. They left the IND-CCA security of $\mathsf{FO}^{\not\perp\prime}$ in the QROM as an open problem [GMP21a, Section 5.3].

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, h, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$h \leftarrow \mathsf{H}'(ek)$	$\mu := \mathsf{H}'(\mu)$	$(\bar{K}', r') := \mathsf{G}(\mu', h)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$(\bar{K}, r) := \mathsf{G}(\mu, \mathsf{H}'(ek))$	$c' := \mathsf{Enc}(ek, \mu'; r')$
$\overline{dk} := (dk, ek, h, s)$	$c := \mathsf{Enc}(ek, \mu; r)$	**if** $c \neq c'$, **then return** $K := \mathsf{H}(s, \mathsf{H}'(c))$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\bar{K}, \mathsf{H}'(c))$	**else return** $K := \mathsf{H}(\bar{K}', \mathsf{H}'(c))$
	return (c, K)	

Fig. 13. $\mathsf{KEM} := \mathsf{FO}^{\not\perp\prime}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}']$ in Kyber and Saber.

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, h, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$h \leftarrow \mathsf{H}'(ek)$	$(\bar{K}, r) := \mathsf{G}(\mu, \mathsf{H}'(ek))$	$(\bar{K}', r') := \mathsf{G}(\mu', h)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$c := \mathsf{Enc}(ek, \mu; r)$	$c' := \mathsf{Enc}(ek, \mu'; r')$
$\overline{dk} := (dk, ek, h, s)$	$K := \mathsf{H}(\bar{K}, c)$	**if** $c \neq c'$, **then return** $K := \mathsf{H}(s, c)$
return $(ek, \overline{dk})$	**return** (c, K)	**else return** $K := \mathsf{H}(\bar{K}', c)$

Fig. 14. KEM := $\mathsf{FO}^{\not\perp\prime\prime}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{H}']$ in FrodoKEM.

D.3 FO with additional hash

$\mathsf{HFO}^{\perp}$ and $\mathsf{HFO}^{\not\perp}$ (as known as $\mathsf{QFO}^{\perp}$ and $\mathsf{QFO}^{\not\perp}$) [TU16, HHK17, JZC+18, JZM19] transform a weakly-secure probabilistic PKE into IND-CCA-secure KEM like FO and add hash value of the message. HQC [AAB+20] uses $\mathsf{HFO}^{\perp}$. NTRU LPRime of NTRU Prime [BBC+20] uses a variant of $\mathsf{HFO}^{\not\perp,\mathsf{prf}}$.
Let $\{0,1\}^{\ell(\kappa)}$ be the plaintext space of PKE. Let $\mathsf{G}\colon \{0,1\}^* \to \mathcal{R}_{\mathsf{Enc}}$, $\mathsf{F}\colon \{0,1\}^{\ell(\kappa)} \times \{0,1\}^* \to \{0,1\}^{\ell'(\kappa)}$, $\mathsf{H}\colon \{0,1\}^{\ell(\kappa)} \times (\mathcal{C}_{\mathsf{PKE}} \times \{0,1\}^{\ell'(\kappa)}) \to \{0,1\}^{k(\kappa)}$, and $\mathsf{H}_{\mathsf{prf}}\colon \{0,1\}^{\ell(\kappa)} \times (\mathcal{C}_{\mathsf{PKE}} \times \{0,1\}^{\ell'(\kappa)}) \to \{0,1\}^{k(\kappa)}$ be hash functions modeled by the random oracles. $\mathsf{HFO}^{\perp}$ and $\mathsf{HFO}^{\not\perp}$ is summarized as Figure 15 and Figure 16, respectively. Assuming the IND-CPA security of PKE, the obtained KEM scheme is IND-CCA-secure in the QROM. See e.g., [KSS+20]. For the case of explicit rejection $\mathsf{HFO}^{\perp}$, we need to invoke [BHH+19, Theorem 4].

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$\overline{dk} := (dk, ek)$	$r := \mathsf{G}(\mu)$	$r' := \mathsf{G}(\mu')$
return $(ek, \overline{dk})$	$c_0 := \mathsf{Enc}(ek, \mu; r)$	$c_0' := \mathsf{Enc}(ek, \mu'; r')$
	$c_1 := \mathsf{F}(\mu)$	$c_1' := \mathsf{F}(\mu')$
	$K := \mathsf{H}(\mu, c_0, c_1)$	**if** $(c_0, c_1) \neq (c_0', c_1')$, **then return** $K := \perp$
	return $((c_0, c_1), K)$	**then return** $K := \mathsf{H}(\mu', c_0, c_1)$

Fig. 15. KEM := $\mathsf{HFO}^{\perp}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{F}]$ for HQC.

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$r := \mathsf{G}(\mu)$	$r' := \mathsf{G}(\mu')$
$\overline{dk} := (dk, ek, s)$	$c_0 := \mathsf{Enc}(ek, \mu; r)$	$c_0' := \mathsf{Enc}(ek, \mu'; r')$
return $(ek, \overline{dk})$	$c_1 := \mathsf{F}(\mu, ek)$	$c_1' := \mathsf{F}(\mu', ek)$
	$K := \mathsf{H}(\mu, c_0, c_1)$	**if** $(c_0, c_1) \neq (c_0', c_1')$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
	return $((c_0, c_1), K)$	**then return** $K := \mathsf{H}(\mu', c_0, c_1)$

Fig. 16. KEM := $\mathsf{HFO}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{G}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ for NTRU LPRime of NTRU Prime.

D.4 SXY

SXY transforms a weakly-secure *deterministic* PKE into IND-CCA-secure KEM. This variant is employed by NTRU (NTRU-HPS and NTRU-HRSS) [CDH+20]. See Figure 5 for the summary. Assuming disjoint-simulatability of PKE, the obtained KEM scheme is IND-CCA-secure in the QROM [SXY18].

D.5 HU with implicit rejection

The final one is a transformation that transforms a weakly-secure *deterministic* PKE into IND-CCA-secure KEM, employed by Classic McEliece [ABC+20] and Streamlined NTRU Prime of NTRU Prime [BBC+20]. We interpret the transformation as $\mathsf{HU}^{\not\perp,\mathsf{prf}}$ [JZM19].

Let $\mathcal{M}$ be the plaintext space of PKE. Let $\mathsf{F}\colon \mathcal{M} \to \{0,1\}^{\ell'(\kappa)}$, $\mathsf{H}\colon \mathcal{M} \times (C_{\mathsf{PKE}} \times \{0,1\}^{\ell'(\kappa)}) \to \{0,1\}^{k(\kappa)}$, and $\mathsf{H}_{\mathsf{prf}}\colon \{0,1\}^{\ell(\kappa)} \times (C_{\mathsf{PKE}} \times \{0,1\}^{\ell'(\kappa)}) \to \{0,1\}^{k(\kappa)}$ be hash functions modeled by the random oracle. The $\mathsf{HU}^{\not\perp}$ is summarized as Figure 17. Assuming disjoint-simulatability of PKE, the obtained KEM scheme is IND-CCA-secure in the QROM [SXY18, ABC+20]. We note that the implementation of F, H, and $\mathsf{H}_{\mathsf{prf}}$ of Streamlined NTRU Prime has a problem of nested random oracles and we cannot show it is IND-CCA-secure. See section R for the detail.

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$ and $c = (c_0, c_1)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' := \mathsf{Dec}(dk, c_0)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \perp$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c)$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu)$ // CM	$c_0' := \mathsf{Enc}(ek, \mu')$
return $(ek, \overline{dk})$	$c_1 := \mathsf{F}(\mu, ek)$ // sntrupr	$c_1' := \mathsf{F}(\mu')$ // CM
	$K := \mathsf{H}(\mu, c_0, c_1)$	$c_1' := \mathsf{F}(\mu', ek)$ // sntrupr
	return $((c_0, c_1), K)$	**if** $(c_0, c_1) \neq (c_0', c_1')$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
		else return $K := \mathsf{H}(\mu', c_0, c_1)$

Fig. 17. $\mathsf{KEM} := \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ in Classic McEliece (CM) and Streamlined NTRU Prime (sntrupr) of NTRU Prime.

Remark D.2. One might wonder $\overline{\mathsf{Dec}}$ in Classic McEliece has no explicit re-encryption check ([ABC+20, Sec.2.3.3]). In their specification, Dec in Classic McEliece internally checks $c_0' = \mathsf{Enc}(ek, \mu')$ or not ([ABC+20, Sec.2.2.4]).

D.6 Hashes in the wild

Finally, we summarize how KEMs implement G, F, H, and $\mathsf{H}_{\mathsf{prf}}$.

Table 6. Summary of variants of FOs in NIST PQC Round 3 KEM Candidates (finalists and alternates): Before version 4.2, BIKE's G uses SHA384 and AES256-CTR. $\mathsf{SHAKE256}_{\ell}$ will outputs the first ℓ bits of SHAKE256. $\mathsf{SHA3\text{-}512}_{r}$ and $\mathsf{SHA3\text{-}512}_{l}$ outputs the first and second 256 bits of SHA3-512. BIKE and SIKE use L in the underlying PKE to mask a message with masking value computed from the shared random value L(shared). BIKE uses $\mathsf{SHA3\text{-}384}_{256}(r)$ and SIKE uses $\mathsf{SHAKE256}_{n}(j)$ as L. In FrodoKEM, FrodoKEM-640 uses $(\mathsf{SHAKE}, k) = (\mathsf{SHAKE128}, 128)$, FrodoKEM-976 uses $(\mathsf{SHAKE}, k) = (\mathsf{SHAKE256}, 192)$, and FrodoKEM-1344 use $(\mathsf{SHAKE}, k) = (\mathsf{SHAKE256}, 256)$.

Name	Trans.	G	F
Classic McEliece [ABC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	–	$\mathsf{SHAKE256}_{256}(\mathtt{0x02}, \mu)$
Kyber [SAB+20]	$\mathsf{FO}^{\not\perp\prime}$	$\mathsf{SHA3\text{-}512}_{r}(\mu, \mathsf{SHA3\text{-}256}(ek))$ [a]	–
NTRU [CDH+20]	SXY	–	–
Saber [DKR+20]	$\mathsf{FO}^{\not\perp\prime}$	$\mathsf{SHA3\text{-}512}_{r}(\mu, \mathsf{SHA3\text{-}256}(ek)))$ [b]	–
BIKE [ABB+20]	$\mathsf{FO}^{\not\perp}$	$\mathsf{SHAKE256}(\mu)$	–
FrodoKEM [NAB+20]	$\mathsf{FO}^{\not\perp\prime\prime}$	$\mathsf{SHAKE}_{2k,l}(\mathsf{SHAKE}_{k}(ek), \mu)$ [c]	–
HQC [AAB+20]	$\mathsf{HFO}^{\perp}$	$\mathsf{SHAKE256}_{512}(\mu, \mathtt{0x03})$ [d]	$\mathsf{SHAKE256}_{512}(\mu, \mathtt{0x04})$
Streamlined NTRU Prime [BBC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	–	$\mathsf{SHA512}_{256}(\mathtt{0x02}, \mathsf{SHA512}_{256}(\mathtt{0x03}, \mu), \mathsf{SHA512}_{256}(\mathtt{0x04}, ek))$
NTRU LPRime [BBC+20]	$\mathsf{HFO}^{\not\perp,\mathsf{prf}}$	$\mathsf{SHA512}_{256}(\mathtt{0x05}, \mu)$ [e]	$\mathsf{SHA512}_{256}(\mathtt{0x02}, \mu, \mathsf{SHA512}_{256}(\mathtt{0x04}, ek))$
SIKE [JAC+20]	$\mathsf{FO}^{\not\perp}$	$\mathsf{SHAKE256}_{e_2}(\mu, ek)$	–

Name	Trans.	H	$\mathsf{H}_{\mathsf{prf}}$
Classic McEliece [ABC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	$\mathsf{SHAKE256}_{256}(\mathtt{0x01}, \mu, (c_0, c_1))$	$\mathsf{SHAKE256}_{256}(\mathtt{0x00}, s, (c_0, c_1))$
Kyber [SAB+20]	$\mathsf{FO}^{\not\perp\prime\prime}$	$\mathsf{SHAKE256}_{X}(\mathsf{SHA3\text{-}512}_{l}(\mu, \mathsf{SHA3\text{-}256}(ek)), \mathsf{SHA3\text{-}256}(c))$	$\mathsf{SHAKE256}_{X}(s, \mathsf{SHA3\text{-}256}(c))$
NTRU [CDH+20]	SXY	$\mathsf{SHA3\text{-}256}(\mu)$	$\mathsf{SHA3\text{-}256}(s, c)$
Saber [DKR+20]	$\mathsf{FO}^{\not\perp\prime\prime}$	$\mathsf{SHA3\text{-}256}(\mathsf{SHA3\text{-}512}_{l}(\mu, \mathsf{SHA3\text{-}256}(ek)), \mathsf{SHA3\text{-}256}(c))$	$\mathsf{SHA3\text{-}256}(s, \mathsf{SHA3\text{-}256}(c))$
BIKE [ABB+20]	$\mathsf{FO}^{\not\perp}$	$\mathsf{SHA3\text{-}384}_{256}(\mu, c)$	$\mathsf{SHA3\text{-}384}_{256}(s, c)$
FrodoKEM [NAB+20]	$\mathsf{FO}^{\not\perp\prime}$	$\mathsf{SHAKE}_{k}(c, \mathsf{SHAKE}_{2k,r}(\mathsf{SHAKE}_{k}(ek), \mu))$	$\mathsf{SHAKE}(c, s)$
HQC [AAB+20]	$\mathsf{HFO}^{\perp}$	$\mathsf{SHAKE256}_{512}(\mu, c, \mathtt{0x05})$	–
Streamlined NTRU Prime [BBC+20]	$\mathsf{HU}^{\not\perp,\mathsf{prf}}$	$\mathsf{SHA512}_{256}(\mathtt{0x01}, \mathsf{SHA512}_{256}(\mathtt{0x03}, \mu), c)$	$\mathsf{SHA512}_{256}(\mathtt{0x00}, \mathsf{SHA512}_{256}(\mathtt{0x03}, s), c)$
NTRU LPRime [BBC+20]	$\mathsf{HFO}^{\not\perp,\mathsf{prf}}$	$\mathsf{SHA512}_{256}(\mathtt{0x01}, \mu, c)$	$\mathsf{SHA512}_{256}(\mathtt{0x00}, s, c)$
SIKE [JAC+20]	$\mathsf{FO}^{\not\perp}$	$\mathsf{SHAKE256}_{k}(\mu, c)$	$\mathsf{SHAKE256}_{k}(s, c)$

[a] Kyber uses an intermediate PKE scheme with short randomness which internally uses PRF $\mathsf{SHAKE256}_{X}(r, i)$ for $i = 1, 2, \ldots$ with appropriate length parameter X.
[b] Saber uses an intermediate PKE scheme with short randomness which internally uses XOF $\mathsf{SHAKE128}(r)$.
[c] FrodoKEM uses an intermediate PKE scheme with short randomness which internally uses XOF $\mathsf{SHAKE}(\mathtt{0x96}, r)$.
[d] HQC uses an intermediate PKE scheme with short randomness which internally uses XOF $\mathsf{SHAKE256}(r, \mathtt{0x02})$.
[e] NTRU LPRime uses an intermediate PKE scheme with short randomness which internally uses XOF $\mathsf{AES256\text{-}CTR}(r)$.

E Property of T

In this section, we show that T preserves ciphertext indistinguishability of disjoint simulatability.

Theorem E.1. *Suppose that a probabilistic PKE* PKE *is ciphertext indistinguishable and* OW-CPA*-secure. Then,* $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ *is also ciphertext indistinguishable in the QROM.*
Precisely speaking, for any quantum adversary $\mathcal{A}$ *against* PKE' *issuing at most* q_{G} *quantum queries to* G*, there exist quantum adversaries* $\mathcal{A}_{01}$ *against* OW-CPA *security of PKE and* $\mathcal{A}_{12}$ *against ciphertext indistinguishability of* PKE *such that*

$$\mathsf{Adv}^{\mathsf{ds\text{-}ind}}_{\mathsf{PKE}', \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{A}}(\kappa) \leq 2q_{\mathsf{G}} \sqrt{\mathsf{Adv}^{\mathsf{ow\text{-}cpa}}_{\mathsf{PKE}, \mathcal{D}_{\mathcal{M}}, \mathcal{A}_{01}}(\kappa)} + \mathsf{Adv}^{\mathsf{ds\text{-}ind}}_{\mathsf{PKE}, \mathcal{D}_{\mathcal{M}}, \mathcal{S}, \mathcal{A}_{12}}(\kappa).$$

Proof: Let us consider the following sequence of games, Game_0, Game_1, and Game_2. Let S_i denote the event that the adversary outputs $b' = 1$ in Game_i.

Game_0*:* This game is defined as follows:

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

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

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

$\mathsf{F}(m^*, r^*)$	$\mathcal{A}_{01}^{\mathsf{G}}(ek, c^*)$:
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mathsf{inp} := (ek, c^*)$
$c^* := \mathsf{Enc}(ek, m^*; r^*)$	$i \leftarrow [q_{\mathsf{H}}]$
$\mathsf{inp} := (ek, c^*)$	Run $\mathcal{A}^{\mathsf{G}}(\mathsf{inp})$ until i-th query $\lvert\hat{x}\rangle$ to G
return inp	**if** $i >$ number of queries to G, **return** $\perp$
	else return $x' := \mathsf{Measure}(\lvert\hat{x}\rangle)$

Fig. 18. Algorithm F and adversary $\mathcal{A}_{01}$

Game_2*:* This game is the same as Game_1 except that a challenge ciphertext is generated by the simulator $\mathcal{S}(1^\kappa, ek)$:

$$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa); c^* \leftarrow \mathcal{S}(1^\kappa, ek); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(ek, c^*); \textbf{return } b'.$$

This completes the descriptions of games. It is easy to see that we have

$$\mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}', \mathcal{D}_\mathcal{M}, \mathcal{S}, \mathcal{A}}(\kappa) = |\Pr[S_0] - \Pr[S_2]|.$$

We give an upperbound for this by the following lemmas.

Lemma E.1. *There exists an adversary $\mathcal{A}_{01}$ such that*

$$|\Pr[S_0] - \Pr[S_1]| \leq 2q_{\mathsf{G}}\sqrt{\mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{PKE}, \mathcal{D}_\mathcal{M}, \mathcal{A}_{01}}(\kappa)}.$$

Proof (Proof of Lemma E.1). Let F be an algorithm described in Figure 18. It is easy to see that Game_0 can be restated as

$$m^* \leftarrow \mathcal{D}_\mathcal{M}; r^* \leftarrow \mathsf{G}(m^*); \mathsf{inp} := \mathsf{F}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(\mathsf{inp}); \textbf{return } b'.$$

and Game_1 can be restated as

$$m^* \leftarrow \mathcal{D}_\mathcal{M}; r^* \leftarrow \mathcal{R}; \mathsf{inp} := \mathsf{F}(ek, m^*; r^*); b' \leftarrow \mathcal{A}^{\mathsf{G}(\cdot)}(\mathsf{inp}); \textbf{return } b'.$$

Applying the O2H lemma (Lemma A.2) with $\mathcal{X} = \mathcal{M}'$, $\mathcal{Y} = \mathcal{R}$, $\mathcal{D}_\mathcal{X} = \mathcal{D}_\mathcal{M}$, $x = m^*$, $y = r^*$, and algorithms $\mathcal{A}$ and F, we have

$$|\Pr[S_0] - \Pr[S_1]| \leq 2q_{\mathsf{G}}\sqrt{\Pr[m^* \leftarrow \mathcal{A}_{01}^{\mathsf{G}}(ek, c^*)]}.$$

where $\mathcal{A}_{01}^{\mathsf{G}}$ is an algorithm described in Figure 18, $(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$, $m^* \leftarrow \mathcal{D}_\mathcal{M}$, $r^* \leftarrow \mathcal{R}$, and $c^* := \mathsf{Enc}(ek, m^*, r^*)$.

We have $\Pr[m^* \leftarrow \mathcal{A}_{01}^{\mathsf{G}}(ek, c^*)] \leq \mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{PKE}, \mathcal{D}_\mathcal{M}, \mathcal{A}_{01}}(\kappa)$. By combining these inequalities, the lemma is proven. □

Lemma E.2. *There exists an adversary $\mathcal{A}_{12}$ such that*

$$|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE}, \mathcal{D}_\mathcal{M}, \mathcal{S}, \mathcal{A}_{12}}(\kappa).$$

The proof is very clear and we omit it.

Combining the above two lemmas, we obtain the wanted result. □

F Property of $\mathsf{U}^{\not\perp}$

As we seen in Figure 10, $\mathsf{U}^{\not\perp}$ and $\mathsf{SXY} = \mathsf{U}_m^{\not\perp}$ are not connected. Indeed, we face a subtle problem to apply indifferentiable reduction in Bindel et al. [BHH+19]: Suppose that we have $\mathcal{A}$ against SPR-CCA security of KEM obtained by $\mathsf{U}^{\not\perp}$. In their indifferentiable reduction, they construct $\mathcal{A}_m$ against SPR-CCA security of KEM obtained by $\mathsf{U}_m^{\not\perp}$. $\mathcal{A}_m$ given $\mathsf{H}_m\colon \mathcal{M} \to \mathcal{K}$ simulates $\mathsf{H}\colon \mathcal{M} \times \mathcal{C} \to \mathcal{K}$ by

$$\mathsf{H}(\mu, c) = \begin{cases} \mathsf{H}_m(\mu) & \text{if } c = \mathsf{Enc}(ek, \mu) \\ \mathsf{H}'(\mu, c) & \text{otherwise.} \end{cases}$$

Unfortunately, this simulation makes $\mathsf{H}(s, c)$ different from $\mathsf{H}_{\mathsf{prf}}(s, c)$ at the point (s, c) with $c = \mathsf{Enc}(ek, s)$. Hence, we directly prove the security properties.

Table 7. Summary of Games for the Proof of Theorem F.1. We define $g(\mu) = \mathsf{Enc}(ek, \mu) = \mathsf{Enc}_0(ek, \mu; \mathsf{G}(\mu))$.

Game	H	G	c^*	K^*	Decryption valid c	Decryption invalid c	justification
Game_0	H	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}(\mu^*, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	
Game_1	H	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}(\mu^*, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	Lemma 2.2
$\mathsf{Game}_{1.1}$	H	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}(\mu^*, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
$\mathsf{Game}_{1.2}$	$\mathsf{H}'_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}(\mu^*, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	if key is not bad
Game_2	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}(\mu^*, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	if key is not bad
Game_3	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	conceptual
$\mathsf{Game}_{3.1}$	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathsf{Enc}(ek, \mu^*)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_4	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	DS-IND
Game_5	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	statistical disjointness
$\mathsf{Game}_{5.1}$	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_6	$\mathsf{H}_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c)$	conceptual
$\mathsf{Game}_{6.1}$	$\mathsf{H}'_q \circ g \,/\, \mathsf{H}'$	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	if key is not bad
$\mathsf{Game}_{6.2}$	H	$\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	if key is not bad
Game_7	H	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	Lemma 2.1 + correctness
Game_8	H	$\mathcal{F}(\mathcal{M}, \mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	Lemma 2.2

F.1 SPR-CCA Security

We need to show $\mathsf{U}^{\not\perp}$'s SPR-CCA-security directly. Fortunately, we can use the security proofs for $\mathsf{SXY} = \mathsf{U}^{\not\perp}_m$ with slight modifications. Roughly speaking, we replace $\mathsf{H}(s, c)$ with $\mathsf{H}_q(c)$ and, then, apply the above indifferentiable reduction. Doing so, we can find the situation is essentially equivalent to Game_1 (or Game_7) of Table 4.

Theorem F.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable and δ-correct with negligible δ, then* $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$ *is* SPR-CCA*-secure. Formally speaking, for any $\mathcal{A}$ against the* SPR-CCA *security of* KEM *issuing at most q_{DEC} queries to the decapsulation oracle and q_{G} and q_{H} queries to* G *and* H *respectively, there exist $\mathcal{A}_{34}$ against ciphertext-indistinguishability of* PKE *such that*

$$\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) \le \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4\delta + 16(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta + 16(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2\delta + 4(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|}.$$

Theorem F.2. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable and δ-correct with negligible δ, then* $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$ *is* SPR-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$ against the SPR-CCA *security of* KEM *issuing at most q_{DEC} queries to the decapsulation oracle and q_{G} and q_{H} queries to* G *and* H, *respectively, there exist $\mathcal{A}_{34}$ against ciphertext-indistinguishability of* PKE *such that*

$$\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A},\mathcal{S}}(\kappa) \le \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|} + 4\delta.$$

Proof of Theorem F.1: We use the game-hopping proof. We consider Game_i for $i = 0, \ldots, 8$. We summarize the games in Table 7. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i. Let Acc and $\overline{\mathsf{Acc}}$ denote the event that the key pair (ek, dk) is accurate and inaccurate, respectively.

Game_0*:* This game is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. Thus, we have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1*:* This game is the same as Game_0 except that $\mathsf{H}(s, c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q \colon C \to \mathcal{K}$ is another random oracle. We remark that $\mathcal{A}$ is not given direct access to H_q.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_0] - \Pr[S_1]| \le 2(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|},$$

where q_{H} and q_{DEC} denote the number of queries to H and DEC the adversary makes, respectively.

$\mathsf{Game}_{1.1}$: This game is the same as Game_1 except that the random oracle $\mathsf{G}(\cdot)$ is chosen from $\mathcal{F}_{\text{good}}(\mathcal{M},\mathcal{R})$ instead of $\mathcal{F}(\mathcal{M},\mathcal{R})$.
Fix (ek, dk). Then, we have $|\Pr[S_1 \mid (ek, dk)] - \Pr[S_{1.1} \mid (ek, dk)]| \leq 8(q_{\mathsf{G}}+q_{\text{Dec}}+1)^2\delta_{ek,dk}$. Taking average over $(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$, we obtain

$$|\Pr[S_1] - \Pr[S_{1.1}]| \leq 8(q_{\mathsf{G}} + q_{\text{Dec}} + 1)^2 \mathrm{Exp}_{(ek,dk)\leftarrow\mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] = 8(q_{\mathsf{G}} + q_{\text{Dec}} + 1)^2\delta.$$

We have $\Pr[\mathrm{Bad}] \leq \delta$ ([LW21, Claim 3]). According to Lemma A.1, for any p, we also have

$$|\Pr[S_{1.1}] - p| \leq |\Pr[S_{1.1} \wedge \neg\mathrm{Bad}] - p| + \delta.$$

$\mathsf{Game}_{1.2}$: This game is the same as $\mathsf{Game}_{1.1}$ except that the random oracle $\mathsf{H}(\cdot,\cdot)$ is simulated as follows: Let $\mathsf{H}'_q\colon \mathcal{C} \to \mathcal{K}$ and $\mathsf{H}'\colon \mathcal{M}\times\mathcal{C} \to \mathcal{K}$ be random oracles. Define

$$\mathsf{H}(\mu, c) = \begin{cases} \mathsf{H}'_q(\mathsf{Enc}(ek, \mu)) & \text{if } c = \mathsf{Enc}(ek, \mu), \\ \mathsf{H}'(\mu, c) & \text{otherwise.} \end{cases}$$

We remark that the decapsulation oracle and the generation of K^* also use this simulation.
If $\neg\mathrm{Bad}$ occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$ is perfectly correct from the definition of G and $g(\mu) := \mathsf{Enc}(ek, \mu; \mathsf{G}(\mu))$ is *injective*. Thus, $\mathsf{H}'_q \circ g\colon \mathcal{M} \to \mathcal{K}$ is a random function and the two games $\mathsf{Game}_{1.1}$ and $\mathsf{Game}_{1.2}$ are equivalent if Bad does not occurs. We have

$$\Pr[S_{1.1} \wedge \neg\mathrm{Bad}] = \Pr[S_{1.2} \wedge \neg\mathrm{Bad}].$$

See [XY19, Lemma 4.3] and [LW21, Claim 4] for the detail.

Game_2: This game is the same as $\mathsf{Game}_{1.2}$ except that the random oracle H is simulated by $\mathsf{H}_q \circ g$ and H' instead of $\mathsf{H}'_q \circ g$ and H'.
If $\neg\mathrm{Bad}$ occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is perfectly correct from the definition of G. Hence, the two games $\mathsf{Game}_{1.2}$ and Game_2 are equivalent, because a value of $\mathsf{H}'_q(c)$ for an invalid c is not used in $\mathsf{Game}_{1.2}$: that is, we have

$$\Pr[S_{1.2} \wedge \neg\mathrm{Bad}] = \Pr[S_2 \wedge \neg\mathrm{Bad}]$$

See the proof of [XY19, Lemma 4.4] and [LW21, Claim 5] for the detail.

Game_3: This game is the same as Game_2 except that K^* is set as $\mathsf{H}_q(c^*)$ and the decapsulation oracle always returns $\mathsf{H}_q(c)$ as long as $c \neq c^*$. This decapsulation oracle will denoted by $\textsc{Dec}$'.
If $\neg\mathrm{Bad}$ occurs, then $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is perfectly correct from the definition of G. , the two games Game_2 and Game_3 are equivalent: that is, we have

$$\Pr[S_2 \wedge \neg\mathrm{Bad}] = \Pr[S_3 \wedge \neg\mathrm{Bad}].$$

See the proof of [XY19, Lemma 4.5] for the detail.
According to Lemma A.1, for any p, we have

$$|\Pr[S_3 \wedge \neg\mathrm{Bad}] - p| \leq |\Pr[S_3] - p| + \delta.$$

$\mathsf{Game}_{3.1}$: This game is the same as Game_3 except that G is chosen from $\mathcal{F}(\mathcal{M},\mathcal{R})$ instead of $\mathcal{F}_{\text{good}}(\mathcal{M},\mathcal{R})$.

$$|\Pr[S_3] - \Pr[S_{3.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \mathrm{Exp}_{(ek,dk)\leftarrow\mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] = 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2\delta.$$

(We note that H and the challenge ciphertext also query to G internally.)

Game_4: This game is the same as Game_3 except that c^* is generated by $\mathcal{S}(1^\kappa)$.
The difference between two games Game_3 and Game_4 is bounded by the advantage of ciphertext indistinguishability in disjoint simulatability as in [XY19, Lemma 4.7]. We have

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa).$$

Game_5*:* This game is the same as Game_4 except that $K^* \leftarrow \mathcal{K}$ instead of $K^* \leftarrow \mathsf{H}_q(c^*)$.
In Game_4, if $c^* \leftarrow \mathcal{S}(1^\kappa)$ is not in $\mathsf{Enc}(ek, \mathcal{M})$, then the adversary has no information about $K^* = \mathsf{H}_q(c^*)$ and thus, K^* looks uniformly at random. Hence, the difference between two games Game_4 and Game_5 is bounded by the statistical disjointness in disjoint simulatability as in [XY19, Lemma 4.8].
We have

$$|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

$\mathsf{Game}_{5.1}$*:* This game is the same as Game_5 except that G is chosen from $\mathcal{F}_{\mathrm{good}}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}(\mathcal{M}, \mathcal{R})$.

$$|\Pr[S_5] - \Pr[S_{5.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}})^2 \mathrm{Exp}_{(ek,dk) \leftarrow \mathsf{Gen}_0(1^\kappa)}[\delta_{ek,dk}] \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \delta.$$

(We note that H and the challenge ciphertext also query to G internally.)
According to Lemma A.1, for any p, we have

$$|\Pr[S_{5.1} \wedge \neg \mathrm{Bad}] - p| \leq |\Pr[S_{5.1}] - p| + \delta.$$

Game_6*:* This game is the same as Game_5 except that the decapsulation oracle is reset as $\textsc{Dec}$. Similar to the case for Game_2 and Game_3, if a key pair is accurate, the two games Game_5 and Game_6 are equivalent as in the proof of [XY19, Lemma 4.5]. We have

$$\Pr[S_{5.1} \wedge \neg \mathrm{Bad}] = \Pr[S_6 \wedge \neg \mathrm{Bad}].$$

$\mathsf{Game}_{6.1}$*:* This game is the same as Game_6 except that the random oracle H is simulated by $\mathsf{H}'_q \circ g$ and H' as in $\mathsf{Game}_{1.2}$. If a key pair is not bad, the two games Game_6 and $\mathsf{Game}_{6.1}$ are equivalent as in the proof of [XY19, Lemma 4.4]. We have

$$\Pr[S_6 \wedge \neg \mathrm{Bad}] = \Pr[S_{6.1} \wedge \neg \mathrm{Bad}].$$

$\mathsf{Game}_{6.2}$*:* This game is the same as $\mathsf{Game}_{6.1}$ except that the random oracle $\mathsf{H}(\cdot)$ is set as the original. If a key pair is not bad, the two games $\mathsf{Game}_{6.1}$ and $\mathsf{Game}_{6.2}$ are equivalent as in the proof of [XY19, Lemma 4.4]. We have

$$\Pr[S_{6.1} \wedge \neg \mathrm{Bad}] = \Pr[S_{6.2} \wedge \neg \mathrm{Bad}].$$

We have, for any p,

$$|\Pr[S_{6.2} \wedge \neg \mathrm{Bad}] - p| \leq |\Pr[S_{6.2}] - p| + \delta$$

from Lemma A.1.

Game_7*:* This game is the same as $\mathsf{Game}_{6.2}$ except that the random oracle G is chosen from $\mathcal{F}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}_{\mathrm{good}}(\mathcal{M}, \mathcal{R})$. We have,

$$|\Pr[S_{6.2}] - \Pr[S_7]| \leq 8(q_{\mathsf{G}} + q_{\textsc{Dec}})^2 \delta. \leq 8(q_{\mathsf{G}} + q_{\textsc{Dec}} + 1)^2 \delta.$$

Game_8*:* This game is the same as Game_7 except that $\mathsf{H}_q(c)$ in the decapsulation is replaced by $\mathsf{H}(s, c)$.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_7] - \Pr[S_8]| \leq 2(q_{\mathsf{H}} + q_{\textsc{Dec}})/\sqrt{|\mathcal{M}|}.$$

We note that This game is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$. Thus, we have

$$\Pr[S_8] = \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing those (in)equalities, we obtain the following bound:

$$\begin{aligned}
\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_8]| \\
&\leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4\delta \\
&\quad + 16(q_{\mathsf{G}} + q_{\textsc{Dec}} + 1)^2 \delta + 16(q_{\mathsf{G}} + q_{\mathsf{H}} + 1)^2 \delta + 4(q_{\mathsf{H}} + q_{\textsc{Dec}})/\sqrt{|\mathcal{M}|}.
\end{aligned}$$

Table 8. Summary of Games for the Proof of Theorem F.3: '$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$' implies that the challenger generates $c^* \leftarrow \mathcal{S}(1^\kappa)$ and returns $\perp$ if $c^* \in \mathsf{Enc}(ek, \mathcal{M})$.

Game	H	c^*	K^*	Decryption valid c	Decryption invalid c	justification
Game_0	H	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	
Game_1	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	statistical disjointness
Game_2	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	Lemma 2.2
Game_3	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}_q(c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c^*)$ is hidden
Game_4	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}(s, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	Lemma 2.2
Game_5	H	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\overline{\mathsf{Dec}}(dk, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	re-encryption check
Game_6	H	$\mathcal{S}(1^\kappa)$	$\overline{\mathsf{Dec}}(dk, c^*)$	$\mathsf{H}(\mu, c)$	$\mathsf{H}(s, c)$	statistical disjointness

Proof of Theorem F.2: The proof of Theorem F.2 is a simplified version of that of Theorem F.1, since it does not require to consider G. Ignoring the transition between real G with good G, we obtain the bound as follows:

$$\begin{aligned}\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_8]| \\ &\le 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|} + 4\delta + \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{A}_{34},\mathcal{S}}(\kappa) + \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).\end{aligned}$$

F.2 SSMT-CCA Security

We can show SSMT-CCA security of $\mathsf{U}^{\not\perp}$ by using the essentially same proof of that for SXY.

Theorem F.3. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$ *is* SSMT-CCA*-secure.*
Formally speaking, for any adversary $\mathcal{A}$ against SSMT-CCA *security of* KEM, *we have*

$$\mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) \le 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|}.$$

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof Sketch: We use the game-hopping proof. We consider Game_i for $i = 0, \dots, 6$. We summarize the games in Table 8. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i. Let Acc and $\overline{\mathrm{Acc}}$ denote the event that the key pair (ek, dk) is accurate and inaccurate, respectively.

Game_0*:* This game is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$(c^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}.$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1*:* In this game, the ciphertext is set as $\perp$ if c^* is in $\mathsf{Enc}(ek, \mathcal{M})$. The difference between two games Game_0 and Game_1 is bounded by statistical disjointness.

$$|\Pr[S_0] - \Pr[S_1]| \le \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Game_2*:* This game is the same as Game_1 except that $\mathsf{H}(s, c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q : C \to \mathcal{K}$ is another random oracle.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_1] - \Pr[S_2]| \le 2(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|},$$

where q_{H} denote the number of queries to $\mathsf{H}_{\mathrm{prf}}$ the adversary makes.

Game_3*:* This game is the same as Game_2 except that $K^* := \mathsf{H}_q(c^*)$ instead of chosen random. Since c^* is always outside of $\mathsf{Enc}(ek, \mathcal{M})$, $\mathcal{A}$ cannot obtain any information about $\mathsf{H}_q(c^*)$. Hence, the two games Game_2 and Game_3 are equivalent and we have

$$\Pr[S_2] = \Pr[S_3].$$

Game_4*:* This game is the same as Game_3 except that $\mathsf{H}_q(\cdot)$ is replaced by $\mathsf{H}(s, \cdot)$. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_3] - \Pr[S_4]| \leq 2(q_{\mathsf{H}} + q_{\mathrm{Dec}})/\sqrt{|\mathcal{M}|}.$$

Game_5*:* This game is the same as Game_4 except that $K^* := \overline{\mathsf{Dec}}(dk, c^*)$ instead of $\mathsf{H}(s, c^*)$. Recall that c^* is always in *outside* of $\mathsf{Enc}(ek, \mathcal{M})$. Thus, we always have $\mathsf{Dec}(c^*) = \bot$ or $\mathsf{Enc}(ek, \mathsf{Dec}(c^*)) \neq c^*$ and, thus, $K^* = \mathsf{H}(s, c^*)$. Hence, the two games are equivalent and we have

$$\Pr[S_4] = \Pr[S_5].$$

Game_6*:* We finally replace how to compute c^*. In this game, the ciphertext is chosen by $\mathcal{S}(1^\kappa)$ as in Game_0. The difference between two games Game_5 and Game_6 is bounded by statistical disjointness.

$$|\Pr[S_5] - \Pr[S_6]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Moreover, this game Game_6 is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa)$ with $b = 1$.

$$\Pr[S_6] = \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain Theorem F.3:

$$\begin{aligned}\mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{S},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_6]| \\ &\leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}} + q_{\mathrm{Dec}})/\sqrt{|\mathcal{M}|}.\end{aligned}$$

F.3 SCFR-CCA Security

Theorem F.4. *If* PKE *is* XCFR*-secure or* SCFR-CCA*-secure, then* $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$ *is* SCFR-CCA*-secure in the QROM.*

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof. Suppose that an adversary outputs a ciphertext c which is decapsulated into $K \neq \bot$ by both $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $\mu_i' = \mathsf{Dec}(dk_i, c)$ for $i \in \{0, 1\}$. We also define $\mu_i := \mu_i'$ if $c = \mathsf{Enc}(ek_i, \mu_i')$ and $\bot$ otherwise.
We have five cases defined as follows:

1. Case 1 ($\mu_0 = \mu_1 \neq \bot$): This violates XCFR-security of SCFR-CCA-security of the underlying PKE and it is easy to make a reduction.
2. Case 2 ($\bot \neq \mu_0 \neq \mu_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0, c) = \mathsf{H}(\mu_1, c)$. Thus, we succeed to find a collision for H, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 3 ($\mu_0 = \bot$ and $\mu_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(s_0, c) = \mathsf{H}(\mu_1, c)$. Notice that we can replace $\mathsf{H}(s_0, \cdot)$ with $\mathsf{H}_q(\cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $(c, (\mu_1, c))$ between H_q and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
4. Case 4 ($\mu_0 \neq \bot$ and $\mu_1 = \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0, c) = \mathsf{H}(s_1, c)$. Again, we can replace $\mathsf{H}(s_1, \cdot)$ with $\mathsf{H}_q(\cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((\mu_0, c), c)$ between H and H_q. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 5 (The other cases): In this case, we find a collision $((s_0, c), (s_1, c))$ of H, which is indeed collision if $s_0 \neq s_1$ which occurs with probability at lease $1 - 1/2^\ell$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. ◻

Table 9. Summary of Games for the Proof of Theorem G.1. We define $g(\mu) = \mathsf{Enc}(ek, \mu) = \mathsf{Enc}_0(ek, \mu; \mathsf{G}(\mu))$.

Game	H	F	G	c_0^*	c_1^*	K^*	Decryption K	condition	justification
Game_0	H	F	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}(\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	
$\mathsf{Game}_{0.1}$	H	F	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}(\mu^*)$	$\mathsf{H}(\mu^*)$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	Lemma 2.1 + correctness
Game_1	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}_q(c_0^*)$	$\mathsf{H}_q(c_0^*)$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	if key is not bad
Game_2	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}_q(c_0^*)$	$\mathsf{H}_q(c_0^*)$	$\mathsf{H}_q(c_0)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}_q(c_0)$	if key is not bad
Game_3	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}_q(c_0^*)$	$\mathsf{H}_q(c_0^*)$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	statistical
$\mathsf{Game}_{3.1}$	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathsf{Enc}(ek,\mu^*)$	$\mathsf{F}_q(c_0^*)$	$\mathsf{H}_q(c_0^*)$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	Lemma 2.1 + correctness
Game_4	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$\mathsf{F}_q(c_0^*)$	$\mathsf{H}_q(c_0^*)$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	DS-IND
Game_5	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$\mathsf{F}_q(c_0*)$	$U(\mathcal{K})$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	statistical disjointness
$\mathsf{Game}_{5.1}$	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	statistical disjointness
$\mathsf{Game}_{5.2}$	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}_q(c_0)$	if $c_1 = \mathsf{F}_q(c_0)$	Lemma 2.1 + correctness
Game_6	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}_q(c_0)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}_q(c_0)$	statistical
Game_7	$\mathsf{H}_q \circ g$	$\mathsf{F}_q \circ g$	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	if key is not bad
$\mathsf{Game}_{7.1}$	H	F	$\mathcal{F}_{\mathrm{good}}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	if key is not bad
Game_8	H	F	$\mathcal{F}(\mathcal{M},\mathcal{R})$	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	if $c_0 = \mathsf{Enc}(ek,\mu)$ and $c_1 = \mathsf{F}(\mu)$	Lemma 2.1 + correctness

G Property of $\mathsf{HU}_m^{\perp}$

Let us consider $\mathsf{HU}_m^{\perp}$ [JZM19]: Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let $\mathcal{C}$ and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$ and $\mathsf{F}\colon \mathcal{M} \to \mathcal{H}$ be hash functions modeled by random oracles. Let $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}_m^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ is a KEM scheme obtained by using $\mathsf{HU}_m^{\perp}$.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$m \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c_0)$
$\overline{dk} := (dk, ek)$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \perp$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
return $(ek, \overline{dk})$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \perp$
	$K := \mathsf{H}(\mu)$	**else return** $K := \mathsf{H}(\mu')$
	return $((c_0, c_1), K)$	

G.1 SPR-CCA security:

Theorem G.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. *Suppose that a ciphertext space* $\mathcal{C}$ *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable with simulator* $\mathcal{S}$ *and* δ*-correct with negligible* δ*, then* $\mathsf{KEM} = \mathsf{HU}_m^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *is* SPR-CCA*-secure, where we use a new simulator* $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.
Formally speaking, for any $\mathcal{A}$ *against the* SPR-CCA *security of* KEM *issuing at most* q_{DEC} *queries to the decapsulation oracle and* q_{F}, q_{G}*, and* q_{H} *queries to* F, G*, and* H*, respectively, there exist* $\mathcal{A}_{34}$ *against ciphertext-indistinguishability of* PKE *such that*

$$
\begin{aligned}
\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{S}',\mathcal{A}}(\kappa) \leq{}& \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 16(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta + 4\delta \\
&+ 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}})^2\delta + 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}} + q_{\mathrm{DEC}} + 1)^2\delta \\
&+ (2q_{\mathrm{DEC}} + 1)/|\mathcal{H}| + q_{\mathrm{DEC}}/(|\mathcal{H}| - 1)
\end{aligned}
$$

Theorem G.2. *Suppose that a ciphertext space* $\mathcal{C}$ *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable and* δ*-correct with negligible* δ*, then* $\mathsf{KEM} = \mathsf{HU}_m^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *is* SPR-CCA*-secure.*

Proof Sketch of Theorem G.1: We use the game-hopping proof. We consider Game_i for $i = 0, \ldots, 8$. We summarize the games in Table 9. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i.
We mainly follow the security proof in [JZM19, XY19, LW21], while we use a new simulator $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$ instead of $\mathcal{S}' = \mathsf{Enc}(ek, \mathcal{M}) \times U(\mathcal{H})$.

Game_0*:* This game is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$\mu^* \leftarrow \mathcal{M}; c_0^* := \mathsf{Enc}(ek, \mu^*; \mathsf{G}(\mu^*)); c_1^* := \mathsf{F}(\mu^*); K^* := \mathsf{H}(\mu^*).$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

$\mathsf{Game}_{0.1}$*:* This game is the same as Game_0 except that the random oracle G is chosen from $\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$ instead of $\mathcal{F}(\mathcal{M}, \mathcal{R})$. As in the proof of Theorem 5.1, we have

$$|\Pr[S_0] - \Pr[S_{0.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2 \delta.$$

In addition, we have $\Pr[\mathsf{Bad}] \leq \delta$ and $|\Pr[S_{0.1}] - p| \leq |\Pr[S_{0.1} \wedge \neg\mathsf{Bad}] - p| + \delta$ for any $p \in [0, 1]$.

Game_1*:* This game is the same as $\mathsf{Game}_{0.1}$ except that the random oracles H and F are simulated by $\mathsf{H}_q \circ g$ and $\mathsf{F}_q \circ g$, respectively, where $\mathsf{H}_q \colon \mathcal{C} \to \mathcal{K}$ and $\mathsf{F}_q \colon \mathcal{C} \to \mathcal{H}$ are random oracles and $g(\mu) := \mathsf{Enc}(ek, \mu)$. If key is not bad, then those games are equivalent and we have

$$\Pr[S_{0.1} \wedge \neg\mathsf{Bad}] = \Pr[S_1 \wedge \neg\mathsf{Bad}].$$

Game_2*:* This game is the same as Game_1 except that the decapsulation oracle internally computes K as $\mathsf{H}_q(c_0)$ and c_1 as $\mathsf{F}_q(c_0)$. If key is not bad, then those games are equivalent and we have

$$\Pr[S_1 \wedge \neg\mathsf{Bad}] = \Pr[S_2 \wedge \neg\mathsf{Bad}].$$

Game_3*:* In this game the decapsulation oracle ignores whether $c_0 = \mathsf{Enc}(ek, \mu)$ or not. That is, when $(c_0, c_1) \neq (c_0^*, c_1^*)$, the oracle returns $K = \mathsf{H}_q(c_0)$ if $c_1 = \mathsf{F}_q(c_0)$.
Let us consider the following cases:

- If $c_0 = \mathsf{Enc}(ek, \mu)$ for some μ, then the results are equal.
- If $c_0 \notin \mathsf{Enc}(ek, \mathcal{M})$ and $c_1 \neq \mathsf{F}_q(c_0)$, then the results are equal.
- If $c_0 \notin \mathsf{Enc}(ek, \mathcal{M})$ and $c_1 = \mathsf{F}_q(c_0)$, then the results differ ($\perp$ in Game_2 but $K = \mathsf{H}_q(c)$ in Game_3).

The difference occurs when c_0 is outside of $\mathsf{Enc}(ek, \mathcal{M})$ and $c_1 = \mathsf{F}_q(c_0)$. Notice that the adversary cannot access such hash values directly, since it is given F instead of F_q. Therefore, any c_1 hits the value $\mathsf{F}_q(c_0)$ with probability at most $1/|\mathcal{H}|$ and we obtain the bound $q_{\mathrm{DEC}}/|\mathcal{H}|$. (If a decapsulation query is quantum, we will get another bound $2q_{\mathrm{DEC}}(|\mathcal{H}|)^{-1/2}$.) We have

$$|\Pr[S_2 \wedge \neg\mathsf{Bad}] - \Pr[S_3 \wedge \neg\mathsf{Bad}]| \leq q_{\mathrm{DEC}}/|\mathcal{H}|.$$

We also have for any p,

$$|\Pr[S_3 \wedge \neg\mathsf{Bad}] - p| \leq |\Pr[S_3] - p| + \delta.$$

$\mathsf{Game}_{3.1}$*:* This game is the same as Game_3 except that G is chosen from $\mathcal{F}(\mathcal{M}, \mathcal{R})$. We have

$$|\Pr[S_3] - \Pr[S_{3.1}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}} + q_{\mathrm{DEC}} + 1)^2 \delta.$$

(We note that H, F, DEC, and the challenge ciphertext also query to G internally.)

Game_4*:* We replace $c_0^* := \mathsf{Enc}(ek, \mu^*; \mathsf{G}(\mu^*))$ with $c_0^* \leftarrow \mathcal{S}(1^\kappa)$. The difference is bounded by the advantage of ciphertext indistinguishability. We have

$$|\Pr[S_{3.1}] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa).$$

Game_5*:* This game is the same as Game_4 except that $K^* \leftarrow \mathcal{K}$ instead of $K^* \leftarrow \mathsf{H}_q(c_0^*)$.
Suppose that c_0^* is outside of $\mathsf{Enc}(ek, \mathcal{M})$ in both games: If so, the adversary cannot access to $K^* = \mathsf{H}_q(c_0^*)$ via H. Suppose that the adversary queries (c_0, c_1) to DEC. If $c_0 = c_0^*$ and $c_1 = c_1^*$, then it receives $\perp$ in both games. If $c_0 = c_0^*$ and $c_1 \neq c_1^*$, then $c_1 \neq \mathsf{F}_q(c_0^*) = c_1^*$ holds and it receives $\perp$ in both games. Thus, the two games are equal if c_0^* is outside of $\mathsf{Enc}(ek, \mathcal{M})$.
Hence, the difference is bounded by the statistical disjointness in disjoint simulatability. We have

$$|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

$\mathsf{Game}_{5.1}$*:* Here, our proof leaves the proof in [JZM19]. This game is the same as $\mathsf{Game}_{5.1}$ except that $c_1^* \leftarrow U(\mathcal{H})$ instead of $c_1^* := \mathsf{F}_q(c_0^*)$.

Recall that the adversary cannot access the real hash value $c_1^* = \mathsf{F}_q(c_0^*)$ directly if c_0^* is the outside of $\mathsf{Enc}(ek, \mathcal{M})$. When the adversary queries (c_0, c_1) for $c_0 \neq c_0^*$, there is no leak on $\mathsf{F}_q(c_0^*)$. Suppose that the adversary queries (c_0^*, c_1) for $\textsc{Dec}$.

- In Game_5, we have $c_1^* = \mathsf{F}_q(c_0^*)$. If $c_1 = c_1^*$, then it receives $\perp$; otherwise, that is, if $c_1 \neq c_1^*$, it also receives $\perp$.
- In $\mathsf{Game}_{5.1}$, we have $c_1^* \leftarrow U(\mathcal{H})$.
  - If $c_1^* = \mathsf{F}_q(c_0^*)$, then this game is the same as Game_5.
  - Suppose that $c_1^* \neq \mathsf{F}_q(c_0^*)$. If $c_1 = c_1^*$, then it receives $\perp$; otherwise, it receives $\perp$ if and only if $c_1 \neq \mathsf{F}_q(c_0^*)$; it receives $K = \mathsf{H}_q(c_0^*)$ if $c_1 = \mathsf{F}_q(c_0^*)$.

Thus, assuming that c_0^* is the outside of $\mathsf{Enc}(ek, \mathcal{M})$ and $c_1^* \neq \mathsf{F}_q(c_0^*)$, a value c_1 hits $\mathsf{F}_q(c_0^*)$ with probability at most $1/(|\mathcal{H}| - 1)$. We have

$$|\Pr[S_5] - \Pr[S_{5.1}]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 1/|\mathcal{H}| + q_{\textsc{Dec}}/(|\mathcal{H}| - 1).$$

$\mathsf{Game}_{5.2}$*:* This game is the same as $\mathsf{Game}_{5.1}$ except that G is chosen from $\mathcal{F}_{\text{good}}(\mathcal{M}, \mathcal{R})$. We have

$$|\Pr[S_{5.1}] - \Pr[S_{5.2}]| \leq 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}})^2\delta.$$

We also have, for any p,

$$|\Pr[S_{5.2}] - p| \leq |\Pr[S_{5.2} \wedge \neg\mathsf{Bad}] - p| + \delta.$$

Game_6*:* This game is the same as $\mathsf{Game}_{5.2}$ except that the decapsulation algorithm checks if $c_0 = \mathsf{Enc}(ek, \mu)$ and $c_1 = \mathsf{F}_q(c_0)$.

Let us consider the following cases for a decapsulation query (c_0, c_1):

- If $c_0 = \mathsf{Enc}(ek, \mu)$ for some μ, then the results are equal since the key is not bad.
- If $c_0 \notin \mathsf{Enc}(ek, \mathcal{M})$ and $c_1 \neq \mathsf{F}_q(c_0)$, then the results are equal.
- If $c_0 \notin \mathsf{Enc}(ek, \mathcal{M})$ and $c_1 = \mathsf{F}_q(c_0)$, then the results differ ($\perp$ in Game_6 but $K = \mathsf{H}_q(c)$ in $\mathsf{Game}_{5.2}$).

The difference occurs when c_0 is outside of $\mathsf{Enc}(ek, \mathcal{M})$ and $c_1 = \mathsf{F}_q(c_0)$. Notice that the adversary cannot access such hash values directly, since it is given F instead of F_q. Therefore, any c_1 hits the value $\mathsf{F}_q(c_0)$ with probability at most $1/|\mathcal{H}|$ and we obtain the bound $q_{\textsc{Dec}}/|\mathcal{H}|$. (If the query is quantum, we will get another bound $2q_{\textsc{Dec}}(|\mathcal{H}|)^{-1/2}$.) We have

$$|\Pr[S_{5.2} \wedge \neg\mathsf{Bad}] - \Pr[S_6 \wedge \neg\mathsf{Bad}]| \leq q_{\textsc{Dec}}/|\mathcal{H}|.$$

Game_7*:* This game is the same as Game_6 except that the decapsulation oracle use H and F instead of H_q and F_q, respectively. If the key is not bad, then this is the conceptual change and we have

$$\Pr[S_6 \wedge \neg\mathsf{Bad}] = \Pr[S_7 \wedge \neg\mathsf{Bad}].$$

$\mathsf{Game}_{7.1}$*:* This game is the same as Game_7 except that H and F are modified as the original. If the key is not bad, then this is the conceptual change and we have

$$\Pr[S_7 \wedge \neg\mathsf{Bad}] = \Pr[S_{7.1} \wedge \neg\mathsf{Bad}].$$

We also have, for any p,

$$|\Pr[S_{7.1} \wedge \neg\mathsf{Bad}] - p| \leq |\Pr[S_{7.1}] - p| + \delta.$$

Game_8*:* This game is the same as $\mathsf{Game}_{7.1}$ except that the random oracle G is chosen from $\mathcal{F}(\mathcal{M}, \mathcal{R})$. We have

$$|\Pr[S_{7.1}] - \Pr[S_8]| \leq 8(q_{\mathsf{G}} + q_{\textsc{Dec}})^2\delta.$$

We note that this game is the original game $\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$. We have

$$\Pr[S_8] = \Pr[\mathsf{Expt}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summary: Summarizing those (in)equalities, we obtain the following bound:

$$\begin{aligned}\mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_8]| \\ &\leq \mathsf{Adv}^{\text{ds-ind}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{S},\mathcal{A}_{34}}(\kappa) + 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 16(q_{\mathsf{G}} + q_{\mathrm{DEC}} + 1)^2\delta + 4\delta \\ &\quad + 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}})^2\delta + 8(q_{\mathsf{G}} + q_{\mathsf{H}} + q_{\mathsf{F}} + q_{\mathrm{DEC}} + 1)^2\delta \\ &\quad + (2q_{\mathrm{DEC}} + 1)/|\mathcal{H}| + q_{\mathrm{DEC}}/(|\mathcal{H}| - 1)\end{aligned}$$

G.2 Sparseness

Theorem G.3. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. Let* $\mathsf{KEM} = \mathsf{HU}^{\perp}_m[\mathsf{PKE},\mathsf{H},\mathsf{F}]$. *Let $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$ be the simulator for* SPR-CCA *security of* KEM*. Then,* KEM *is $1/|\mathcal{H}|$-sparse.*

Proof. Let us consider $(c_0, c_1) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{H})$. If c_0 is decrypted into $\mu' \neq \perp$, then $c_1 = \mathsf{F}(\mu')$ with probability at most $1/|\mathcal{H}|$. Thus, KEM is $1/|\mathcal{H}|$-sparse. ◻

H Property of $\mathsf{HU}^{\perp}$

In this section, we consider a variant of HU with explicit rejection, $\mathsf{HU}^{\perp}$. Let $\mathsf{PKE} = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let C and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $\mathsf{H}\colon \mathcal{M} \times C \times \mathcal{H} \to \mathcal{K}$ and $\mathsf{F}\colon \mathcal{M} \to \mathcal{H}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}^{\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F}]$ is defined as follows:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$m \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c_0)$
$\overline{dk} := (dk, ek)$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \perp$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
return $(ek, \overline{dk})$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \perp$
	$K := \mathsf{H}(\mu, c_0, c_1)$	**else return** $K := \mathsf{H}(\mu', c_0, c_1)$
	return $((c_0, c_1), K)$	

H.1 SPR-CCA security:

Theorem H.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable with simulator $\mathcal{S}$, then* $\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F}]$ *is* SPR-CCA*-secure, where we use the new simulator $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.*

Theorem H.2. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F}]$ *is* SPR-CCA*-secure.*

In order to show those proof, we consider the following theorem for indifferentiable reduction, which is obtained by mimicking that for $\mathsf{U}^{x}_{m} \leftrightarrow \mathsf{U}^{x}$ in [BHH+19, Theorem 5].

Theorem H.3 **($\mathsf{HU}^{\perp}_{m} \leftrightarrow \mathsf{HU}^{\perp}$:).** *Let* PKE *be a deterministic PKE. Let* $\mathsf{KEM}_m = \mathsf{HU}^{\perp}_m[\mathsf{PKE},\mathsf{H}_m,\mathsf{F}]$ *and* $\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F}]$.

1. *If* KEM_m *is* SPR-CCA*-secure, then* KEM *is* SPR-CCA*-secure also.*
2. *If* KEM *is* SPR-CCA*-secure, then* KEM_m *is* SPR-CCA*-secure also.*

Proof (The first part). Suppose that we have an adversary $\mathcal{A}$ against SPR-CCA-security of KEM. We construct an adversary $\mathcal{A}_m$ against SPR-CCA-security of KEM_m with random oracle $\mathsf{H}_m\colon \mathcal{M} \to \mathcal{K}$ as follows: $\mathcal{A}_m$ samples a fresh random oracle $\mathsf{H}' \leftarrow \mathsf{Func}(\mathcal{M} \times C \times \mathcal{H}, \mathcal{K})$ and set

$$\mathsf{H}(\mu, c_0, c_1) = \begin{cases} \mathsf{H}_m(\mu) & \text{if } c_0 = \mathsf{Enc}(ek, \mu) \text{ and } c_1 = \mathsf{F}(\mu) \\ \mathsf{H}'(\mu, c_0, c_1) & \text{otherwise.} \end{cases}$$

The simulation is perfect. ◻

Proof (The second part). Suppose that we have an adversary $\mathcal{A}_m$ against SPR-CCA-security of KEM_m. We construct an adversary $\mathcal{A}$ against SPR-CCA-security of KEM with random oracle $\mathsf{H}\colon \mathcal{M} \times (C \times \mathcal{H}) \to \mathcal{K}$ as follows: $\mathcal{A}$ define

$$\mathsf{H}_m(\mu) := \mathsf{H}\big(\mu, \mathsf{Enc}(ek, \mu), \mathsf{F}(\mu)\big).$$

This simulation is perfect. □

H.2 Sparseness

$\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ is $1/|\mathcal{H}|$-sparse as $\mathsf{HU}^{\perp}_m$.

Theorem H.4. *Suppose that a ciphertext space C of PKE depends on the public parameter only. Let $\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$. Let $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$ be the simulator for SPR-CCA security of KEM. Then, KEM is $1/|\mathcal{H}|$-sparse.*

Proof. Let us consider $(c_0, c_1) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{H})$. If c_0 is decrypted into $\mu' \neq \perp$, then $c_1 = \mathsf{F}(\mu')$ with probability at most $1/|\mathcal{H}|$. Thus, KEM is $1/|\mathcal{H}|$-sparse. □

I Property of $\mathsf{HU}^{\not\perp}_m$

Let us review $\mathsf{HU}^{\not\perp}_m$. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let C and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $\mathsf{H}\colon \mathcal{M} \to \mathcal{K}$, $\mathsf{H}_{\mathsf{prf}}\colon \{0,1\}^{\ell} \times C \times \mathcal{H} \to \mathcal{K}$, and $\mathsf{F}\colon \mathcal{M} \to \mathcal{H}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ is defined as follows:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$m \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c_0)$
$s \leftarrow \{0,1\}^{\ell}$	$c_0 := \mathsf{Enc}(ek, m)$	**if** $\mu' = \perp$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \mathsf{H}_{\mathsf{prf}}(s, (c_0, c_1))$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu)$	**else return** $K := \mathsf{H}(\mu')$
	return $((c_0, c_1), K)$	

I.1 SPR-CCA Security

Bindel et al. showed that if $\mathsf{KEM}^{\perp} = \mathsf{U}^{\perp}_m[\mathsf{PKE}, \mathsf{H}]$ is IND-CCA-secure then $\mathsf{KEM}^{\not\perp} = \mathsf{U}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_{\mathsf{prf}}]$ is also IND-CCA-secure [BHH+19, Theorem 3] by overwriting $\perp$ from the decapsulation query c with the PRF value $\mathsf{H}_{\mathsf{prf}}(s, c)$. The same indifferentiable reduction can be applied to SPR-CCAsecurity and the case for $\mathsf{HU}^{\perp}_m$ and $\mathsf{HU}^{\not\perp}_m$ and obtain the following theorem.

Theorem I.1 **($\mathsf{HU}^{\perp}_m \to \mathsf{HU}^{\not\perp}_m$).** *Let PKE be a deterministic PKE. Let $\mathsf{KEM}^{\perp} = \mathsf{HU}^{\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ and $\mathsf{KEM}^{\not\perp} = \mathsf{HU}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$. If $\mathsf{KEM}^{\perp}$ is SPR-CCA-secure, then $\mathsf{KEM}^{\not\perp}$ is also SPR-CCA-secure.*

Proof. Suppose that we have an adversary $\mathcal{A}$ against SPR-CCA-security of $\mathsf{KEM}^{\not\perp}$. We construct an adversary $\mathcal{A}'$ against SPR-CCA-security of $\mathsf{KEM}^{\perp}$ as follows: Given an encapsulation key ek, a target ciphertext (c_0^*, c_1^*), and a key K_b^*, $\mathcal{A}'$ samples a fresh seed $s \leftarrow \mathcal{M}$. It runs $\mathcal{A}$ on input ek, (c_0^*, c_1^*), and K_b^*. If $\mathcal{A}$ queries a ciphertext (c_0, c_1) to the decapsulation oracle, then $\mathcal{A}'$ queries the ciphertext (c_0, c_1) and receives K. If $K \neq \perp$, then it returns K to $\mathcal{A}$; Otherwise, it queries $(s, (c_0, c_1))$ to the random oracle $\mathsf{H}_{\mathsf{prf}}$, receives $\tilde{K}$, and returns $\tilde{K}$ to $\mathcal{A}$. If $\mathcal{A}$ outputs b' and halts, then $\mathcal{A}'$ also outputs b' and halts.
This simulation is clearly perfect and the theorem follows. □

Apply the above indifferentiable reduction with Theorem I.2 and Theorem I.3, we obtain the following theorems:

Theorem I.2. *Let $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. Suppose that a ciphertext space C of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable with simulator $\mathcal{S}$, then $\mathsf{KEM} = \mathsf{HU}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ is SPR-CCA-secure, where we use the new simulator $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.*

Theorem I.3. *Suppose that a ciphertext space C of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then $\mathsf{KEM} = \mathsf{HU}^{\not\perp}_m[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ is SPR-CCA-secure, where we use the new simulator $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.*

Table 10. Summary of Games for the Proof of Theorem I.4: $\mathsf{Enc}'(ek, \mathcal{M}) = \{(c_0, c_1) = (\mathsf{Enc}(ek, m), \mathsf{F}(\mu) \mid m \in \mathcal{M}\}$. '$\mathcal{S}(1^\kappa) \times U(\mathcal{H}) \setminus \mathsf{Enc}'(ek, \mathcal{M})$' implies that the challenger generates $c_0^* \leftarrow \mathcal{S}(1^\kappa)$, $c_1^* \leftarrow \mathcal{H}$ and returns $\perp$ if $(c_0^*, c_1^*) \in \mathsf{Enc}'(ek, \mathcal{M})$.

Game	H F	c_0^*	c_1^*	K^*	Decryption valid (c_0, c_1)	invalid (c_0, c_1)	justification
Game_0	H F	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c_0, c_1)$	
Game_1	H F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c_0, c_1)$	statistical disjointness
Game_2	H F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c_0, c_1)$	Lemma 2.2
Game_3	H F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\mathsf{H}_q(c_0^*, c_1^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_q(c_0, c_1)$	$\mathsf{H}_q(c_0^*, c_1^*)$ is hidden
Game_4	H F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\mathsf{H}_{\mathrm{prf}}(s, c_0^*, c_1^*)$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c_0, c_1)$	Lemma 2.2
Game_5	H F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c_0, c_1)$	re-encryption check
Game_6	H F	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$	$\mathsf{H}(\mu)$	$\mathsf{H}_{\mathrm{prf}}(s, c_0, c_1)$	statistical disjointness

I.2 SSMT-CCA Security

Theorem I.4. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}_m^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathrm{prf}}]$ *is* SSMT-CCA*-secure.*
Formally speaking, for any $\mathcal{A}$, we have

$$\mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

The security proof is essentially same as that for SXY (Theorem 5.3). Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Game_0*:* This game is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$(c_0^*, c_1^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{H}) \times \mathcal{K}.$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game_1*:* In this game, the ciphertext is set as $\perp$ if c_0^* is in $\mathsf{Enc}(ek, \mathcal{M})$.
The difference between two games Game_0 and Game_1 is bounded by statistical disjointness.

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Game_2*:* This game is the same as Game_1 except that $\mathsf{H}_{\mathrm{prf}}(s, c, d)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c_0, c_1)$ where $\mathsf{H}_q : C \times \mathcal{H} \to \mathcal{K}$ is another random oracle.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_1] - \Pr[S_2]| \leq 2(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2},$$

where $q_{\mathsf{H}_{\mathrm{prf}}}$ denote the number of queries to $\mathsf{H}_{\mathrm{prf}}$ the adversary makes.

Game_3*:* This game is the same as Game_2 except that $K^* := \mathsf{H}_q(c_0^*, c_1^*)$ instead of chosen random. Since c_0^* is always outside of $\mathsf{Enc}(ek, \mathcal{M})$, $\mathcal{A}$ cannot obtain any information about $\mathsf{H}_q(c_0^*, c_1^*)$ via the decapsulation oracle. Hence, the two games Game_2 and Game_3 are equivalent and we have

$$\Pr[S_2] = \Pr[S_3].$$

Game_4*:* This game is the same as Game_3 except that $\mathsf{H}_q(\cdot, \cdot)$ is replaced by $\mathsf{H}_{\mathrm{prf}}(s, \cdot, \cdot)$. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_3] - \Pr[S_4]| \leq 2(q_{\mathsf{H}_{\mathrm{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

Game_5*:* This game is the same as Game_4 except that $K^* := \overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$ instead of $\mathsf{H}_{\mathrm{prf}}(s, c_0^*, c_1^*)$. Recall that c_0^* is always in *outside* of $\mathsf{Enc}(ek, \mathcal{M})$. Thus, we always have $\mathsf{Dec}(c_0^*) = \perp$ or $\mathsf{Enc}(ek, \mathsf{Dec}(c_0^*)) \neq c_0^*$ and, thus, $K^* = \mathsf{H}_{\mathrm{prf}}(s, c_0^*, c_1^*)$. Hence, the two games are equivalent. We have

$$\Pr[S_4] = \Pr[S_5].$$

Game_6: We finally replace how to compute (c_0^*, c_1^*). In this game, the ciphertext is chosen by $\mathcal{S}(1^\kappa) \times U(\mathcal{H})$ as in Game_0.
The difference between two games Game_5 and Game_6 is bounded by statistical disjointness.

$$|\Pr[S_5] - \Pr[S_6]| \leq \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Moreover, this game Game_6 is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$.

$$\Pr[S_6] = \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain Theorem I.4:

$$\begin{aligned}\mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_6]| \\ &\leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\text{DEC}}) \cdot 2^{-\ell/2}.\end{aligned}$$

I.3 SCFR-CCA Security

Theorem I.5. *If* PKE *is* XCFR-*secure or* SCFR-CCA-*secure, then* $\mathsf{KEM} = \mathsf{HU}_m^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \perp$ by $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $\mu_i' = \mathsf{Dec}(dk_i, c_0)$ for $i \in \{0,1\}$. We also define $\mu_i = \mu_i'$ if $c_0 = \mathsf{Enc}(ek_i, \mu_i')$ and $c_1 = \mathsf{F}(\mu_i')$, and $\perp$ otherwise.
We have five cases defined as follows:

1. Case 1 ($\mu_0 = \mu_1 \neq \perp$): This violates XCFR-security or SCFR-CCA-security of the underlying PKE.
2. Case 2 ($\perp \neq \mu_0 \neq \mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0) = \mathsf{H}(\mu_1)$ and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 3 ($\mu_0 = \perp$ and $\mu_1 \neq \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1)$ and $\mathsf{H}(\mu_1)$ and we find a claw $((s_0, c_0, c_1), \mu_1)$ of $\mathsf{H}_{\mathsf{prf}}$ and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
4. Case 4 ($\mu_0 \neq \perp$ and $\mu_1 = \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}(\mu_0) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a claw $(\mu_0, (s_1, c_0, c_1))$ of H and $\mathsf{H}_{\mathsf{prf}}$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of $\mathsf{H}_{\mathsf{prf}}$ if $s_0 \neq s_1$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

If we add ek to F's input, we can reduce the assumption on PKE.

Theorem I.6. *Let* $\mathsf{Col}_{\mathsf{Gen}}$ *be the event that when generating two keys* $(ek_i, dk_i) \leftarrow \mathsf{Gen}(1^\kappa)$ *for* $i \in \{0,1\}$*, they collides, that is,* $ek_0 = ek_1$*. If* $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ *is negligible, then* $\mathsf{KEM} = \mathsf{HU}_m^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *with* $c_1 = \mathsf{F}(\mu, ek)$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \perp$ by $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $\mu_i' = \mathsf{Dec}(dk_i, c_0)$ for $i \in \{0,1\}$. We also define $\mu_i = \mu_i'$ if $c_0 = \mathsf{Enc}(ek_i, \mu_i')$ and $c_1 = \mathsf{F}(\mu_i', ek_i)$, and $\perp$ otherwise.
We consider six cases defined as follows:

1. Case 1-1 ($\mu_0 = \mu_1 \neq \perp$ and $ek_0 = ek_1$): This case rarely occurs since $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ is negligible.
2. Case 1-2 ($\mu_0 = \mu_1 \neq \perp$ and $ek_0 \neq ek_1$): In this case, we have $d = \mathsf{F}(\mu_0', ek_0) = \mathsf{F}(\mu_1', ek_1)$ with $(\mu_0', ek_0) \neq (\mu_1', ek_1)$ and we succeed to find a collision for F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 2 ($\perp \neq \mu_0 \neq \mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0) = \mathsf{H}(\mu_1)$ and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).

4. Case 3 ($\mu_0 = \bot$ and $\mu_1 \neq \bot$): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1)$ and $\mathsf{H}(\mu_1)$ and we find a claw $((s_0, c_0, c_1), \mu_1)$ of $\mathsf{H}_{\mathsf{prf}}$ and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 4 ($\mu_0 \neq \bot$ and $\mu_1 = \bot$): In this case, the decapsulation algorithms output $K = \mathsf{H}(\mu_0) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a claw $(\mu_0, (s_1, c_0, c_1))$ of H and $\mathsf{H}_{\mathsf{prf}}$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
6. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of $\mathsf{H}_{\mathsf{prf}}$ if $s_0 \neq s_1$, which occurs with probability at least $1 - 1/2^{\ell}$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

J Property of $\mathsf{HU}^{\not\perp,\mathsf{prf}}$

Next, we consider a variant of HU with implicit rejection, $\mathsf{HU}^{\not\perp,\mathsf{prf}}$, which is used in Classic McEliece. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let C and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $\mathsf{H}, \mathsf{H}_{\mathsf{prf}} \colon \mathcal{M} \times C \times \mathcal{H} \to \mathcal{K}$ and $\mathsf{F} \colon \mathcal{M} \to \mathcal{H}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ is defined as follows:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c_0)$
$s \leftarrow \mathcal{M}$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \bot$ or $c \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c_0, c_1)$	**else return** $K := \mathsf{H}(\mu', c_0, c_1)$
	return $((c_0, c_1), K)$	

J.1 SPR-CCA Security

Theorem J.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$. *Suppose that a ciphertext space* C *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable with simulator* $\mathcal{S}$, *then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *is* SPR-CCA-*secure, where we use the new simulator* $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.

Theorem J.2. *Suppose that a ciphertext space* C *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *is* SPR-CCA-*secure, where we use the new simulator* $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.

In order to show those theorems, we want to invoke the following theorem for indifferentiable reduction, which is obtained by mimicking that for $\mathsf{U}_m^{\not\perp} \leftrightarrow \mathsf{U}^{\not\perp,\mathsf{prf}}$ in [BHH+19, Theorem 5], and apply it to Theorem I.2 and Theorem I.3.

Theorem J.3 ($\mathsf{HU}_m^{\not\perp} \leftrightarrow \mathsf{HU}^{\not\perp,\mathsf{prf}}$:). *Let* PKE *be a deterministic PKE. Let* $\mathsf{KEM}_m = \mathsf{HU}_m^{\not\perp}[\mathsf{PKE}, \mathsf{H}_m, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *and* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$.

1. *If* KEM_m *is* SPR-CCA-*secure, then* KEM *is* SPR-CCA-*secure also.*
2. *If* KEM *is* SPR-CCA-*secure, then* KEM_m *is* SPR-CCA-*secure also.*

The proof is the same as that of Theorem H.3 and we omit it.

J.2 SSMT-CCA Security

Theorem J.4. *Suppose that a ciphertext space* C *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *is* SSMT-CCA-*secure.*
Formally speaking, for any $\mathcal{A}$, *we have*

$$\mathsf{Adv}^{\mathsf{ssmt\text{-}cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}_{\mathsf{prf}}} + q_{\mathrm{DEC}}) \cdot 2^{-\ell/2}.$$

The security proof is the same as that for $\mathsf{HU}_m^{\not\perp}$ (Theorem I.4) and we omit it.

J.3 SCFR-CCA Security

Theorem J.5. *If* PKE *is* XCFR-*secure or* SCFR-CCA-*secure, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Theorem J.6. *Let* $\mathsf{Col}_{\mathsf{Gen}}$ *be the event that when generating two keys* $(ek_i, dk_i) \leftarrow \mathsf{Gen}(1^\kappa)$ *for* $i \in \{0,1\}$*, they collides, that is,* $ek_0 = ek_1$*. If* $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ *is negligible, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ *with* $c_1 = \mathsf{F}(\mu, ek)$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

The security proofs are the same as those for $\mathsf{HU}^{\not\perp,\mathsf{prf}}$ (Theorem I.5 and Theorem I.6) and we omit them.

K Property of $\mathsf{HU}^{\not\perp}$

Next, we consider another variant of HU with implicit rejection, $\mathsf{HU}^{\not\perp}$. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let $\mathcal{C}$ and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $\mathsf{H}\colon \mathcal{M} \times \mathcal{C} \times \mathcal{H} \to \mathcal{K}$ and $\mathsf{F}\colon \mathcal{M} \to \mathcal{H}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ is defined as follows:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' \leftarrow \mathsf{Dec}(dk, c)$
$s \leftarrow \mathcal{M}$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \perp$ or $c_0 \neq \mathsf{Enc}(ek, \mu')$ or $c_1 \neq \mathsf{F}(\mu'[, ek])$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu[, ek])$	**then return** $K := \mathsf{H}(s, c_0, c_1)$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c_0, c_1)$	**else return** $K := \mathsf{H}(\mu', c_0, c_1)$
	return $((c_0, c_1), K)$	

K.1 SPR-CCA security:

Theorem K.1. *Let* $\mathsf{PKE} = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$*. Suppose that a ciphertext space* $\mathcal{C}$ *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable with simulator* $\mathcal{S}$*, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *is* SPR-CCA-*secure, where we use the new simulator* $\mathcal{S}' = \mathcal{S} \times U(\mathcal{H})$.

Theorem K.2. *Suppose that a ciphertext space* $\mathcal{C}$ *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *is* SPR-CCA-*secure.*

Hence, we use [BHH+19, Theorem 3] here.

Theorem K.3 **($\mathsf{HU}^{\perp} \to \mathsf{HU}^{\not\perp}$).** *Let* PKE *be a deterministic PKE. Let* $\mathsf{KEM}^{\perp} = \mathsf{HU}^{\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *and* $\mathsf{KEM}^{\not\perp} = \mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$*. If* $\mathsf{KEM}^{\perp}$ *is* SPR-CCA-*secure, then* $\mathsf{KEM}^{\not\perp}$ *is also* SPR-CCA-*secure.*

Proof. Suppose that we have an adversary $\mathcal{A}$ against SPR-CCA-security of $\mathsf{KEM}^{\not\perp}$. We construct an adversary $\mathcal{A}'$ against SPR-CCA-security of $\mathsf{KEM}^{\perp}$ as follows: Given an encapsulation key ek, a target ciphertext (c_0^*, c_1^*), and a key K_b^*, $\mathcal{A}'$ samples a fresh seed $s \leftarrow \mathcal{M}$. It runs $\mathcal{A}$ on input ek, (c_0^*, c_1^*), and K_b^*. If $\mathcal{A}$ queries a ciphertext (c_0, c_1) to the decapsulation oracle, then $\mathcal{A}'$ queries the ciphertext (c_0, c_1) and receives K. If $K \neq \perp$, then it returns K to $\mathcal{A}$; Otherwise, it queries (s, c_0, c_1) to the random oracle H, receives $\tilde{K}$, and returns $\tilde{K}$ to $\mathcal{A}$. If $\mathcal{A}$ outputs b' and halts, then $\mathcal{A}'$ also outputs b' and halts.
This simulation is clearly perfect and the theorem follows. □

K.2 SSMT-CCA Security

Theorem K.4. *Suppose that a ciphertext space* $\mathcal{C}$ *of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}]$ *is* SSMT-CCA-*secure.*
Formally speaking, for any $\mathcal{A}$*, we have*

$$\mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) \leq 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}} + q_{\mathrm{Dec}})/\sqrt{|\mathcal{M}|}.$$

The security proof is essentially same as that for SXY (Theorem 5.3). Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Table 11. Summary of Games for the Proof of Theorem K.4: '$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$' implies that the challenger generates $c_0^* \leftarrow \mathcal{S}(1^\kappa)$, $c_1^* \leftarrow \mathcal{H}$ and returns $\perp$ if $c_0^* \in \mathsf{Enc}(ek, \mathcal{M})$.

Game	H	F	c_0^*	c_1^*	K^*	Decryption valid (c_0, c_1)	Decryption invalid (c_0, c_1)	justification
Game$_0$	H	F	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}(s, c_0, c_1)$	
Game$_1$	H	F	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}_q(c_0, c_1)$	Lemma 2.2
Game$_2$	H	F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$U(\mathcal{K})$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}_q(c_0, c_1)$	statistical disjointness
Game$_3$	H	F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\mathsf{H}_q(c_0^*, c_1^*)$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}_q(c_0, c_1)$	$\mathsf{H}_q(c_0^*, c_1^*)$ is hidden
Game$_4$	H	F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\mathsf{H}(s, c_0^*, c_1^*)$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}(s, c_0, c_1)$	Lemma 2.2
Game$_5$	H	F	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$U(\mathcal{H})$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}(s, c_0, c_1)$	re-encryption check
Game$_6$	H	F	$\mathcal{S}(1^\kappa)$	$U(\mathcal{H})$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$	$\mathsf{H}(\mu, c_0, c_1)$	$\mathsf{H}(s, c_0, c_1)$	statistical disjointness

Game$_0$: This game is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$(c_0^*, c_1^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa) \times U(\mathcal{H}) \times \mathcal{K}.$$

We have

$$\Pr[S_0] = 1 - \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 0].$$

Game$_1$: This game is the same as Game$_0$ except that $\mathsf{H}(s, c_0, c_1)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c_0, c_1)$ where $\mathsf{H}_q : \mathcal{C} \times \mathcal{H} \to \mathcal{K}$ is another random oracle. As in[JZC+18, Theorem 1] and [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_1] - \Pr[S_2]| \le 2(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|},$$

where q_{H} denote the number of queries to H the adversary makes.

Game$_2$: In this game, the ciphertext is set as $\perp$ if c_0^* is in $\mathsf{Enc}(ek, \mathcal{M})$.
The difference between two games Game$_1$ and Game$_2$ is bounded by statistical disjointness.

$$|\Pr[S_1] - \Pr[S_2]| \le \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Game$_3$: This game is the same as Game$_2$ except that $K^* := \mathsf{H}_q(c_0^*, c_1^*)$ instead of chosen random. Since c_0^* is always outside of $\mathsf{Enc}(ek, \mathcal{M})$, $\mathcal{A}$ cannot obtain any information about $\mathsf{H}_q(c_0^*, c_1^*)$ via the decapsulation oracle. Hence, the two games Game$_2$ and Game$_3$ are equivalent and we have

$$\Pr[S_2] = \Pr[S_3].$$

Game$_4$: This game is the same as Game$_3$ except that $\mathsf{H}_q(\cdot, \cdot)$ is replaced by $\mathsf{H}_{\mathsf{prf}}(s, \cdot, \cdot)$. As in [JZC+18, Theorem 1] and [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_3] - \Pr[S_4]| \le 2(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|},$$

Game$_5$: This game is the same as Game$_4$ except that $K^* := \overline{\mathsf{Dec}}(\overline{dk}, (c_0^*, c_1^*))$ instead of $\mathsf{H}(s, c_0^*, c_1^*)$. Recall that c_0^* is always in *outside* of $\mathsf{Enc}(ek, \mathcal{M})$. Thus, we always have $\mathsf{Dec}(c_0^*) = \perp$ or $\mathsf{Enc}(ek, \mathsf{Dec}(c_0^*)) \neq c_0^*$ and, thus, $K^* = \mathsf{H}(s, c_0^*, c_1^*)$. Hence, the two games are equivalent. We have

$$\Pr[S_4] = \Pr[S_5].$$

Game$_6$: We finally replace how to compute (c_0^*, c_1^*). In this game, the ciphertext is chosen by $\mathcal{S}(1^\kappa) \times U(\mathcal{H})$ as in Game$_0$.
The difference between two games Game$_5$ and Game$_6$ is bounded by statistical disjointness.

$$|\Pr[S_5] - \Pr[S_6]| \le \mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa).$$

Moreover, this game Game$_6$ is the original game $\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$.

$$\Pr[S_6] = \Pr[\mathsf{Expt}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) = 1 \mid b = 1].$$

Summarizing the (in)equalities, we obtain Theorem K.4:

$$\begin{aligned} \mathsf{Adv}^{\text{ssmt-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa) &= |\Pr[S_0] - \Pr[S_6]| \\ &\le 2\mathsf{Disj}_{\mathsf{PKE},\mathcal{S}}(\kappa) + 4(q_{\mathsf{H}} + q_{\mathrm{DEC}})/\sqrt{|\mathcal{M}|}. \end{aligned}$$

K.3 SCFR-CCA Security

Theorem K.5. *If* PKE *is* XCFR-*secure or* SCFR-CCA-*secure, then* $\mathsf{KEM} = \mathsf{HU}_m^{\not\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F}]$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \perp$ by $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $\mu_i' = \mathsf{Dec}(dk_i, c_0)$ for $i \in \{0,1\}$. We also define $\mu_i = \mu_i'$ if $c_0 = \mathsf{Enc}(ek_i, \mu_i')$ and $c_1 = \mathsf{F}(\mu_i')$, and $\perp$ otherwise.
We have five cases defined as follows:

1. Case 1 ($\mu_0 = \mu_1 \neq \perp$): This violates XCFR-security or SCFR-CCA-security of the underlying PKE.
2. Case 2 ($\perp \neq \mu_0 \neq \mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0, c_0, c_1) = \mathsf{H}(\mu_1, c_0, c_1)$ and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 3 ($\mu_0 = \perp$ and $\mu_1 \neq \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}(s_0, c_0, c_1)$ and $\mathsf{H}(\mu_1, c_0, c_1)$. As in the proof of Theorem F.3, we can replace $\mathsf{H}(s_0, \cdot, \cdot)$ with $\mathsf{H}_q(\cdot, \cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((c_0, c_1), (\mu_1, c_0, c_1))$ between H_q and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
4. Case 4 ($\mu_0 \neq \perp$ and $\mu_1 = \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}(\mu_0, c_0, c_1) = \mathsf{H}(s_1, c_0, c_1)$. This follows as Case 3.
5. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = \mathsf{H}(s_0, c_0, c_1) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of H if $s_0 \neq s_1$, which occurs with overwhelming probability $1 - 1/|\mathcal{M}|$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

If we add ek to F's input, we can reduce the assumption on PKE.

Theorem K.6. *Let* $\mathsf{Col}_{\mathsf{Gen}}$ *be the event that when generating two keys* $(ek_i, dk_i) \leftarrow \mathsf{Gen}(1^\kappa)$ *for* $i \in \{0,1\}$*, they collides, that is,* $ek_0 = ek_1$*. If* $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ *is negligible, then* $\mathsf{KEM} = \mathsf{HU}^{\not\perp}[\mathsf{PKE},\mathsf{H},\mathsf{F},\mathsf{H}_{\mathsf{prf}}]$ *with* $c_1 = \mathsf{F}(\mu, ek)$ *is* SCFR-CCA-*secure in the quantum random oracle model.*

Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by T.

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \perp$ by $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $\mu_i' = \mathsf{Dec}(dk_i, c_0)$ for $i \in \{0,1\}$. We also define $\mu_i = \mu_i'$ if $c_0 = \mathsf{Enc}(ek_i, \mu_i')$ and $c_1 = \mathsf{F}(\mu_i', ek_i)$, and $\perp$ otherwise.
We consider six cases defined as follows:

1. Case 1-1 ($\mu_0 = \mu_1 \neq \perp$ and $ek_0 = ek_1$): This case rarely occurs since $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ is negligible.
2. Case 1-2 ($\mu_0 = \mu_1 \neq \perp$ and $ek_0 \neq ek_1$): In this case, we have $d = \mathsf{F}(\mu_0', ek_0) = \mathsf{F}(\mu_1', ek_1)$ with $(\mu_0', ek_0) \neq (\mu_1', ek_1)$ and we succeed to find a collision for F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 2 ($\perp \neq \mu_0 \neq \mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(\mu_0, c_0, c_1) = \mathsf{H}(\mu_1, c_0, c_1)$ and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).
4. Case 3 ($\mu_0 = \perp$ and $\mu_1 \neq \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}(s_0, c_0, c_1)$ and $\mathsf{H}(\mu_1, c_0, c_1)$. As in the proof of Theorem F.3, we can replace $\mathsf{H}(s_0, \cdot, \cdot)$ with $\mathsf{H}_q(\cdot, \cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((c_0, c_1), (\mu_1, c_0, c_1))$ between H_q and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 4 ($\mu_0 \neq \perp$ and $\mu_1 = \perp$): In this case, the decapsulation algorithms output $K = \mathsf{H}(\mu_0, c_0, c_1) = \mathsf{H}(s_1, c_0, c_1)$. This follows as Case 3.
6. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = \mathsf{H}(s_0, c_0, c_1) = \mathsf{H}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of H if $s_0 \neq s_1$, which occurs with overwhelming probability $1 - 1/|\mathcal{M}|$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

Table 12. Parameter sets of Classic McEliece in Round 3. Note that $q = 2^m$ and $k = n - mt$. (We omit the semi-systematic forms.)

parameter sets	m	n	t	k
`kem/mceliece348864`	12	3488	64	2720
`kem/mceliece460896`	13	4608	96	3360
`kem/mceliece6688128`	13	6688	128	5024
`kem/mceliece6960119`	13	6960	119	5413
`kem/mceliece8192128`	13	8192	128	6528

L Classic McEliece

Review of Classic McEliece: Classic McEliece [ABC+20] is based on the Niederreiter PKE, in which a public key is a scrambled parity-check matrix, a plaintext is an error vector, and a ciphertext is a syndrome. See Table 12 for concrete parameter values (we omit semi-systematic ones).
Define $\mathcal{S} = \{e \in \mathbb{F}_2^n : \mathsf{HW}(e) = t\}$, which is a plaintext space. Let I_{n-k} be an identity matrix of dimension $n-k$. The underlying PKE of Classic McEliece, which we call CM-DPKE, is summarized as follows, where we only consider the systematic form and omit the details for the semi-systematic form:

- $\mathsf{Gen}(1^\kappa)$: Choose a monic irreducible polynomial g in $\mathbb{F}_q[x]$ of degree t and distinct $\alpha_1, \dots, \alpha_n \leftarrow \mathbb{F}_q$. Compute a parity-check matrix $\hat{H} \in \mathbb{F}_2^{n \times k}$ of the Goppa code generated by g and $\alpha_1, \dots, \alpha_n$. Reduce $\hat{H}$ to systematic form $[I_{n-k} \mid T]$. (If this fails, return $\perp$). Output $ek := T \in \mathbb{F}_2^{(n-k)\times k}$ and $dk := (T, \Gamma)$, where $\Gamma := (g, \alpha_1, \dots, \alpha_n)$. We note that using Γ, one can correct an error of the codeword up to t, because the minimum distance of the Goppa code is at least $2t+1$ by design.
- $\mathsf{Enc}(ek, e \in \mathcal{S})$: Define $H := [I_{n-k} \mid T] \in \mathbb{F}_2^{(n-k)\times n}$. Compute $c := He \in \mathbb{F}_2^{n-k}$. Output c.
- $\mathsf{Dec}(dk, c)$: Extend c to $v := (c, 0, \dots, 0) \in \mathbb{F}_2^n$. Find the unique codeword $\tilde{c}$ in the Goppa code defined by Γ that satisfies $\mathsf{HW}(\tilde{c} - v) \le t$. Set $e := v + \tilde{c}$. If $\mathsf{HW}(e) = t$ and $c = He$, then return e. Otherwise, return $\perp$.

Classic McEliece applies $\mathsf{HU}^{\not\perp,\mathsf{prf}}$ to CM-DPKE, where $\mathsf{H}(\mu, c_0, c_1) = \mathsf{SHAKE256}_{256}(\mathtt{0x01}, \mu \| c_0 \| c_1)$ $\mathsf{H}_{\mathsf{prf}}(s, c_0, c_1) = \mathsf{SHAKE256}_{256}(\mathtt{0x00}, s \| c_0 \| c_1)$ $\mathsf{F}(e) = \mathsf{SHAKE256}_{256}(\mathtt{0x02}, e)$:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk} = (dk, s), (c_0, c_1))$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$e \leftarrow \mathsf{FixedWeight}()$	$e := \mathsf{Dec}(dk, c_0)$
$s \leftarrow \mathbb{F}_2^n$	$c_0 := \mathsf{Enc}(ek, e)$	**if** $e = \perp or c_1 \neq \mathsf{F}(e)$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
$ek := T, \overline{dk} := (dk, s)$	$c_1 := \mathsf{F}(e)$	**else return** $K := \mathsf{H}(e, c_0, c_1)$
return $(ek, \overline{dk})$	$K := \mathsf{H}(e, c_0, c_1)$	
	return $((c_0, c_1), K)$	

L.1 Classic McEliece is not collision-free

Let $e_{\mathsf{fixed}} := (1^t, 0^{n-t})$ and $c_{\mathsf{fixed}} := (1^t, 0^{n-k-t})$. We have $t \le mt = n - k$ for all parameter sets of Classic McEliece. Grubbs et al. observed that for any public key T, $c_{\mathsf{fixed}} := (1^t, 0^{n-k-t})$ is a valid ciphertext of plaintext $e_{\mathsf{fixed}} := (1^t, 0^{n-t})$, because $H \cdot e_{\mathsf{fixed}} = [I_{n-k} \mid T] \cdot e_{\mathsf{fixed}} = e_{\mathsf{fixed}} = c_{\mathsf{fixed}}$. Hence, Classic McEliece is not collision free.

Salvaging Collision-Freeness of Classic McEliece: Let us consider Grubbs et al. [GMP21a, Section 5.1] suggested a variant of HU with implicit rejection, in which F takes μ and ek as input, but they did not recommend it since $ek = T$ of Classic McEliece is relatively large. (We can show its security as Theorem K.6.) Instead, we can use a variant of HU with implicit rejection, in which F takes μ and $\mathsf{Hash}(ek)$ as input. We can show its strong collision-freeness assuming that the probability that two independent encryption keys collide is negligible.

Theorem L.1 **(SCFR-CCA-security of modified Classic McEliece).** *The modified Classic McEliece is* SCFR-CCA-*secure in the QROM.*

Theorem L.2. *Let $\mathsf{Col}_{\mathsf{Gen}}$ be the event that when generating two keys $(ek_i, dk_i) \leftarrow \mathsf{Gen}(1^\kappa)$ for $i \in \{0,1\}$, they collides, that is, $ek_0 = ek_1$. If $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ is negligible, then the modified Classic McEliece is* SCFR-CCA*-secure in the QROM.*

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \bot$ by $\overline{dk}_0$ and $\overline{dk}_1$, that is, $\overline{\mathsf{Dec}}(\overline{dk}_0, c) = \overline{\mathsf{Dec}}(\overline{dk}_1, c)$. Let us define $e'_i = \mathsf{Dec}(dk_i, c_0)$ for $i \in \{0,1\}$. We also define $e_i = e'_i$ if $c_0 = \mathsf{Enc}(ek_i, e'_i)$ and $c_1 = \mathsf{F}(e'_i, \mathsf{Hash}(ek_i))$, and $\bot$ otherwise.
We consider seven cases defined as follows:

1. Case 1-1 ($e_0 = e_1 \neq \bot$ and $ek_0 = ek_1$): This case rarely occurs since $\Pr[\mathsf{Col}_{\mathsf{Gen}}]$ is negligible.
2. Case 1-2 ($e_0 = e_1 \neq \bot$, $ek_0 \neq ek_1$, and $\mathsf{Hash}(ek_0) = \mathsf{Hash}(ek_1)$): In this case, we have $\mathsf{Hash}(ek_0) = \mathsf{Hash}(ek_1)$ with $ek_0 \neq ek_1$ and we succeed to find a collision for Hash, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 1-3 ($e_0 = e_1 \neq \bot$, $ek_0 \neq ek_1$, and $\mathsf{Hash}(ek_0) \neq \mathsf{Hash}(ek_1)$): In this case, we have $d = \mathsf{F}(e_0, \mathsf{Hash}(ek_0)) = \mathsf{F}(e_1, \mathsf{Hash}(ek_1))$ with $(e_0, \mathsf{Hash}(ek_0)) \neq (e_1, \mathsf{Hash}(ek_1))$ and we succeed to find a collision for F, which is negligible for any QPT adversary (Lemma 2.3).
4. Case 2 ($\bot \neq e_0 \neq e_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = \mathsf{H}(e_0) = \mathsf{H}(e_1)$ and we succeed to find a collision for H or F, which is negligible for any QPT adversary (Lemma 2.3).
5. Case 3 ($e_0 = \bot$ and $e_1 \neq \bot$): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1)$ and $\mathsf{H}(e_1, c_0, c_1)$ and we find a claw $((s_0, c_0, c_1), (e_1, c_0, c_1))$ of $\mathsf{H}_{\mathsf{prf}}$ and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
6. Case 4 ($e_0 \neq \bot$ and $e_1 = \bot$): In this case, the decapsulation algorithms output $K = \mathsf{H}(e_0, c_0, c_1) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a claw $((e_0, c_0, c_1), (s_1, c_0, c_1))$ of H and $\mathsf{H}_{\mathsf{prf}}$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
7. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = \mathsf{H}_{\mathsf{prf}}(s_0, c_0, c_1) = \mathsf{H}_{\mathsf{prf}}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of $\mathsf{H}_{\mathsf{prf}}$ if $s_0 \neq s_1$, which occurs with probability at least $1 - 1/2^n$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. □

L.2 Security

Assumption:

Definition L.1. *Fix the parameter set. We define a random key-generation algorithm* $\mathsf{RandGen}(pp)$ *as follows: Choose $\hat{H} \leftarrow U(\mathbb{F}_2^{n \times k})$, reduce $\hat{H}$ to systematic form $[I_{n-k} \mid T]$ (if this fails, resample), outputs $\hat{T} \in \mathbb{F}_2^{(n-k)\times k}$*

- *The modified PR-Key assumption: It is hard to distinguish T and $\hat{T}$, where $(T, sk) \leftarrow \mathsf{Gen}(1^\kappa)$ and $\hat{T} \leftarrow \mathsf{RandGen}(pp)$.*
- *The modified Decisional Syndrome Decoding assumption: It is hard to distinguish $(\hat{T}, [I_{n-k} \mid \hat{T}] \cdot e)$ from $(\hat{T}, u)$ with $\hat{T} \leftarrow \mathsf{RandGen}(pp)$, $e \leftarrow \mathsf{FixedWeight}()$, and $u \leftarrow U(\mathbb{F}_2^{n-k})$.*

Security: Assuming the modified PR-Key assumption and the modified Decisional Syndrome Decoding assumption, it is easy to show that CM-DPKE is ciphertext-indistinguishable in the sense of disjoint simulatability. Since $2^n = |\mathbb{F}_2^n| \gg \binom{n}{t} = |\mathcal{S}| \geq |\mathsf{Enc}(ek, \mathcal{M})|$, it has statistical disjointness. Thus, CM-DPKE is strongly disjoint-simulatable. Applying our theorem for $\mathsf{HU}^{\not\perp,\mathsf{prf}}$, we can conclude that Classic McEliece is strongly pseudorandom and smooth under those assumptions.

L.3 Summary:

We show that

- CM-DPKE is strongly disjoint-simulatable under the modified PR-Key assumption and the modified Decisional Syndrome Decoding assumption.
- Thus, Classic McEliece is SPR-CCA-secure and SSMT-CCA-secure in the QROM. (Theorem K.2, Theorem K.4)
- Classic McEliece is ANON-CCA-secure. (Theorem 3.1)
- Classic McEliece leads to ANON-CCA-secure hybrid PKE. (Theorem 4.2, Theorem 3.1)

If we use modified Classic McEliece, then it is SCFR-CCA-secure in the QROM. This implies that the modified Classic McEliece leads to SROB-CCA-secure PKE (Theorem 2.2).
Grubbs et al. [GMP21a] discussed the barrier to show anonymity of hybrid encryption based on Classic McEliece, since Classic McEliece is not collision free. We avoid this barrier by using SPR-CCAsecurity.

M Kyber

Review of Kyber in Round 3: Kyber [SAB+20] is a KEM scheme based on the Module LWE problem. We briefly review the underlying PKE scheme of Kyber. See Table 13 for concrete parameter sets.

Table 13. Parameter sets of Kyber in Round 3.

parameter sets	n	k	q	η_1	η_2	d_U	d_V
Kyber512	256	2	3329	3	2	10	4
Kyber768	256	3	3329	2	2	10	4
Kyber1024	256	4	3329	2	2	11	5

Define $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$ and $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n+1)$. For a positive integer η, we define a central-binomial distribution Ψ_η as $(a_1, b_1, \dots, a_\eta, b_\eta) \leftarrow \{0,1\}^{2\eta}$ and return $\sum_{i=1}^{\eta}(a_i - b_i)$. For a polynomial $P \in \mathcal{R}$, $P \leftarrow \Psi_\eta$ implies each coefficient of the polynomial is chosen from Ψ_η independently.
For $x \in \mathbb{Z}$, we define two functions: $\mathsf{comp}_q(x,d) := \lceil (2^d/q)\cdot x \rfloor \bmod^{\pm} 2^d$ and $\mathsf{decomp}_q(x,d) := \lceil (q/2^d)\cdot x \rfloor$. For $x = (x_1,\dots,x_k) \in \mathbb{Z}^k$ with some k, we define $\mathsf{comp}_q(x,d) := (\mathsf{comp}_q(x_1,d),\dots,\mathsf{comp}_q(x_k,d))$ and $\mathsf{decomp}_q(x,d) := (\mathsf{decomp}_q(x_1,d),\dots,\mathsf{decomp}_q(x_k,d))$
We have $\left|(x - \mathsf{decomp}_q(\mathsf{comp}_q(x,d),d)) \bmod^{\pm} q\right| \le \lceil q/2^{d+1} \rfloor$. We also note that $\mathsf{comp}_q(x,1) = 1$ if $|x \bmod^{\pm} q| \le \lceil q/4 \rfloor$ and 0 otherwise.
The underlying PKE scheme of Kyber, Kyber-PKE, is summarized as follows:

- $\mathsf{Gen}(pp)$: $A \leftarrow \mathcal{R}_q^{k\times k}$ and $(dk, d) \leftarrow (\Psi_{\eta_1}^k)^2$. Compute $B := A \cdot dk + d$. Output $ek := (A, B)$ and dk.
- $\mathsf{Enc}(ek,\mu)$: Sample $t \leftarrow \Psi_{\eta_1}^k$, $e \leftarrow \Psi_{\eta_2}^k$, and $f \leftarrow \Psi_{\eta_2}$. Compute $(U,V) := (tA+e, tB+f+\lceil q/2 \rfloor \cdot \mu) \in \mathcal{R}_q^k \times \mathcal{R}_q$. Output $(U',V') := (\mathsf{comp}_q(U,d_U), \mathsf{comp}_q(V,d_V))$.
- $\mathsf{Dec}(dk,(U',V'))$: Compute $(U,V) := (\mathsf{decomp}_q(U',d_U), \mathsf{decomp}_q(V',d_V))$. Output $\mu' := \mathsf{comp}_q(V - U\cdot dk, 1)$.

We next consider an intermediate PKE scheme $\mathsf{PKE}_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call Kyber-PKE-PRG:

- $\mathsf{Gen}_0(pp) = \mathsf{Gen}(pp)$:
- $\mathsf{Enc}_0(ek,\mu;r)$: Use $\rho_i = \mathsf{SHAKE256}_X(r,i)$ for $i = 0,1,\dots,$ to sample $t \leftarrow \Psi_{\eta_1}^k$, $e \leftarrow \Psi_{\eta_2}^k$, and $f \leftarrow \Psi_{\eta_2}$, where $X = 2\eta_1$ or $2\eta_2$. Compute $(U,V) := (tA+e, tB+f+\lceil q/2 \rfloor\cdot\mu) \in \mathcal{R}_q^k \times \mathcal{R}_q$. Output $(U',V') := (\mathsf{comp}_q(U,d_U), \mathsf{comp}_q(V,d_V))$.
- $\mathsf{Dec}_0(dk,(U,V)) = \mathsf{Dec}(dk,(U,V))$:

Kyber applies $\mathsf{FO}^{\not\perp\prime}$ to Kyber-PKE-PRG, where $\mathsf{H}' = $ SHA3-256, $\mathsf{G}(\mu,h) = $ SHA3-512, and $\mathsf{H} = \mathsf{SHAKE256}_X$ with unspecified output bits X:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk},c)$, where $\overline{dk} = (dk, ek, h, s)$
$(ek,dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}_0(dk,c)$
$h \leftarrow \mathsf{H}'(ek)$	$\mu := \mathsf{H}'(\mu)$	$(\bar{K}', r') := \mathsf{G}(\mu', h)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$(\bar{K}, r) := \mathsf{G}(\mu, \mathsf{H}'(ek))$	$c' := \mathsf{Enc}_0(ek,\mu';r')$
$\overline{dk} := (dk, ek, h, s)$	$c := \mathsf{Enc}_0(ek,\mu;r)$	**if** $c \ne c'$, **then return** $K := \mathsf{H}(s, \mathsf{H}'(c))$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\bar{K}, \mathsf{H}'(c))$	**else return** $K := \mathsf{H}(\bar{K}', \mathsf{H}'(c))$
	return (c, K)	

Security: Grubbs et al. [GMP21a] pointed out there are technical barriers. At first, a pre-key $\bar{K}$ and a randomness r is generated by $\mathsf{G}(\mu, \mathsf{H}'(ek))$. We can treat is as $\bar{K} = \mathsf{G}_0(\mu, \mathsf{H}'(ek))$ and $r = \mathsf{G}_1(\mu, \mathsf{H}'(ek))$, where $\mathsf{G}_0(x)$ and $\mathsf{G}_1(x)$ are defined as the first and last 256-bits of GSHA3-512. Using this notion, we compute $K = \mathsf{H}(\mathsf{G}_0(\mu, \mathsf{H}'(ek)), \mathsf{H}'(c))$. See Table 6. Grubbs et al. solved the problem on nested random oracles on μ by letting $\mathsf{G}_r(\mu) := \mathsf{G}_0(\mu, \mathsf{H}'(ek)) : \{0,1\}^{256} \to \{0,1\}^{256}$ and simulating G_r by a random polynomial over $\mathrm{GF}(2^{256})$ of degree $2q_{\mathsf{G}}+1$ as in [TU16, HHK17]. Grubbs et al. succeeded to show its IND-CCA-security if K was computed as $\mathsf{H}(\mathsf{G}_r(\mu), c)$ as in $\mathsf{FO}^{\not\perp\prime\prime}$. Unfortunately, they left showing $\mathsf{FO}^{\not\perp\prime}$'s IND-CCA-security as open problem. We also left it here. It might be interesting to study anonymity and robustness in the ROM.

N Saber

Review of Saber: Saber [DKR+20] is a KEM scheme based on the Module LWR problem. Saber has three parameter sets LightSaber (lv.1), Saber (lv.3), and FireSaber (lv.5). See Table 14 for concrete parameter values.

Table 14. Parameter sets of Saber in Round 3.

parameter sets	n	k	q	p	T	μ
`LightSaber`	256	2	8192	1024	8	10
`Saber`	256	3	8192	1024	16	8
`FireSaber`	256	4	8192	1024	64	6

Define $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$ and $\mathcal{R}_a = \mathbb{Z}_a[x]/(x^n+1)$ for $a = q, p, T, 2$. Let $\epsilon_q = \lg(q)$, $\epsilon_p = \lg(p)$, and $\epsilon_T = \lg(T)$. For an even positive integer μ, we define a central-binomial distribution β_η as $(a_1, b_1, \dots, a_{\mu/2}, b_{\mu_2}) \leftarrow \{0,1\}^\mu$ and return $\sum_{i=1}^{\mu/2}(a_i - b_i) \in [-\mu/2, \mu/2]$. For a polynomial $P \in \mathcal{R}$, $P \leftarrow \beta_\mu$ implies each coefficient of the polynomial is chosen from β_μ independently. For a positive integer x, we define $\mathsf{shiftright}(x, d)$ as $\lfloor x/2^d \rfloor$, the result of d bit shift of x to right. We define $h_1 := \sum_{i=0}^{n-1} 2^{\epsilon_q - \epsilon_p - 1} x^i \in \mathcal{R}_q$, $h_2 := \sum_{i=0}^{n-1}(2^{\epsilon_p - 2} - 2^{\epsilon_p - \epsilon_T - 1} + 2^{\epsilon_q - \epsilon_p - 1})x^i \in \mathcal{R}_q$, and $h := (h_1, \dots, h_1) \in \mathcal{R}_q^k$.
The underlying PKE scheme of Saber, which we call Saber-PKE, is summarized as follows:

- $\mathsf{Gen}(pp)$: $A \leftarrow \mathcal{R}_q^{k \times k}$ and $dk \leftarrow \beta_\mu^k$. Compute $B := \mathsf{shiftright}(A \cdot dk + h, \epsilon_q - \epsilon_p)$ Output $ek := (A, B)$ and dk.
- $\mathsf{Enc}(ek, \mu)$: Sample $t \leftarrow \beta_\mu^k$. Output $(U, V) := \big(\mathsf{shiftright}(tA + h, \epsilon_q - \epsilon_p), \mathsf{shiftright}(t \cdot B + h_1 - 2^{\epsilon_p - 1}\mu \bmod p, \epsilon_p - \epsilon_T)\big) \in \mathcal{R}_p^k \times \mathcal{R}_T$.
- $\mathsf{Dec}(dk, (U, V))$: Return $\mu' := \mathsf{shiftright}\big(U \cdot dk - 2^{\epsilon_p - \epsilon_T} \cdot V + h_2 \bmod p, \epsilon_p - 1\big) \in \mathcal{R}_2$.

We next consider an intermediate PKE scheme $\mathsf{PKE}_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call Saber-PKE-PRG:

- $\mathsf{Gen}_0(pp) = \mathsf{Gen}(pp)$:
- $\mathsf{Enc}_0(ek, \mu; r)$: Use $\rho = \mathsf{SHAKE128}_X(r)$ to sample $t \leftarrow \beta_\mu^k$. Output $(U, V) := \big(\mathsf{shiftright}(tA + h, \epsilon_q - \epsilon_p), \mathsf{shiftright}(t \cdot B + h_1 - 2^{\epsilon_p - 1}\mu \bmod p, \epsilon_p - \epsilon_T)\big) \in \mathcal{R}_p^k \times \mathcal{R}_T$.
- $\mathsf{Dec}_0(dk, (U', V')) = \mathsf{Dec}(dk, (U, V))$:

Saber applies $\mathsf{FO}^{\not\perp\prime}$ to Saber-PKE-PRG, where $\mathsf{H}' = \mathsf{SHA3\text{-}256}$, $\mathsf{G}(\mu, h) = \mathsf{SHA3\text{-}512}$, and $\mathsf{H} = \mathsf{SHA3\text{-}256}$.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, h, s)$
$(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}_0(dk, c)$
$h \leftarrow \mathsf{H}'(ek)$	$\mu := \mathsf{H}'(\mu)$	$(\bar{K}', r') := \mathsf{G}(\mu', h)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$(\bar{K}, r) := \mathsf{G}(\mu, \mathsf{H}'(ek))$	$c' := \mathsf{Enc}_0(ek, \mu'; r')$
$\overline{dk} := (dk, ek, h, s)$	$c := \mathsf{Enc}_0(ek, \mu; r)$	**if** $c \neq c'$, **then return** $K := \mathsf{H}(s, \mathsf{H}'(c))$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\bar{K}, \mathsf{H}'(c))$	**else return** $K := \mathsf{H}(\bar{K}', \mathsf{H}'(c))$
	return (c, K)	

Security: Grubbs et al. [GMP21a] wrote Saber uses $\mathsf{FO}^{\not\perp\prime\prime}$ as defined in [DKR+20, Section 2.5]. However, the specification uses $\mathsf{FO}^{\not\perp\prime}$ [DKR+20, Section 8.5]. Thus, Saber lacks the IND-CCA-security proof in the QROM as Kyber. We also left proving the IND-CCA security of Saber in the QROM as an open problem. It might be interesting to study anonymity and robustness in the ROM.

O BIKE

Review of BIKE: BIKE in round 3 [ABB+20] is a KEM scheme based on QC-MDPC [MTSB13], which is a variant of the McEliece PKE upon a code with quasi-cyclic (QC) moderate density parity-check (MDPC) matrix. BIKE can

Table 15. Parameter sets of BIKE in Round 3.

parameter sets	r	w	t
BIKE-1	12,323	142	134
BIKE-3	24,659	206	199
BIKE-5	40,973	274	264

be considered as the Niederreiter PKE scheme upon a code with the QC-MDPC matrix. Let $\mathcal{R} := \mathbb{F}[x]/(x^r - 1)$. Let $\mathcal{H}_w := \{(h_0, h_1) \in \mathcal{R}^2 : \mathsf{HW}(h_0) = \mathsf{HW}(e_1) = w/2\}$. Let $\mathcal{E}_t := \{(e_0, e_1) \in \mathcal{R}^2 : \mathsf{HW}(e_0, e_1) = t\}$. For concrete values of r, w, and t, see Table 15.

The underlying CPA-secure PKE scheme of BIKE, which we call BIKE-PKE, is summarized as follows:

- $\mathsf{Gen}(pp)$: $dk := (h_0, h_1) \leftarrow \mathcal{H}_w$. Output $ek = h := h_1 \cdot h_0^{-1} \in \mathcal{R}$ and dk.
- $\mathsf{Enc}(ek, \mu \in \{0,1\}^{256}; r)$: Sample $(e_0, e_1) \leftarrow \mathcal{E}_t(r)$. Compute $u := e_0 + e_1 h \in \mathcal{R}$ and $v := \mu \oplus \mathsf{L}(e_0, e_1)$ and output $c := (u, v)$.
- $\mathsf{Dec}(dk, (u, v))$: Compute $(e_0, e_1) \leftarrow \mathsf{decode}(u h_0, (h_0, h_1))$. Output $\mu' := v \oplus \mathsf{L}(e_0, e_1)$.

Notice that $u h_0 = e_0 h_0 + e_1 h_1$, which is the syndrome of (e_0, e_1) with the parity-check matrix spanned by h_0 and h_1. They also define $\mathsf{L} = \mathsf{SHA3\text{-}384}_{256}$.

BIKE applies $\mathsf{FO}^{\not\perp}$ to BIKE-PKE PKE, where $\mathsf{G} = \mathsf{SHAKE256}$ and $\mathsf{H} = \mathsf{SHA3\text{-}384}_{256}$.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}(dk, c)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$r := \mathsf{G}(\mu)$	$r' := \mathsf{G}(\mu')$
$\overline{dk} := (dk, ek, s)$	$c := \mathsf{Enc}(ek, \mu; r)$	$c' := \mathsf{Enc}(ek, \mu'; r')$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c)$	**if** $c \neq c'$, **then return** $K := \mathsf{H}(s, c)$**else return** $K := \mathsf{H}(\mu', c)$
	return (c, K)	

O.1 Security

Although we can invoke theorems on $\mathsf{FO}^{\not\perp}$ by Grubbs et al. [GMP21a] to show BIKE's anonymity and collision-freeness, we can show BIKE's anonymity through another pass.

Assumptions: For $b \in \{0,1\}$, define the finite set $\mathcal{F}_b := \{h \in \mathcal{R} : \mathsf{HW}(h) \equiv b \pmod 2\}$, that is, a set of all binary vectors of length r and parity b. We suppose that w is even and $w/2$ is odd, which hold for all parameter sets of BIKE.

Definition O.1 (The 2-Decisional Quasi-Cyclic Code-Finding (2-DQCCF) assumption [ABB+20]). *For any (Q)PPT adversary, it is hard to distinguish the following two distributions:*

- $h := h_1 \cdot h_0^{-1}$, *where* $(h_0, h_1) \leftarrow \mathcal{H}_w$.
- $h \leftarrow \mathcal{F}_1$.

Definition O.2 (The 2-Computational Quasi-Cyclic Syndrome Decoding (2-CQCSD) assumption [ABB+20]). *For any (Q)PPT adversary, given* $(h, u := h e_1 + e_0)$, *where* $h \leftarrow \mathcal{F}_1$ *and* $(e_0, e_1) \leftarrow \mathcal{E}_t$, *it is hard to find* $(e_0', e_1') \in \mathcal{E}_t$ *with* $u = h e_1' + e_0'$.

Definition O.3 (The 2-Decisional Quasi-Cyclic Syndrome Decoding (2-DQCSD) assumption [ABB+20]). *For any (Q)PPT adversary, it is hard to distinguish the following two distributions:*

- $(h, u := h e_1 + e_0)$, *where* $h \leftarrow \mathcal{F}_1$ *and* $(e_0, e_1) \leftarrow \mathcal{E}_t$.
- (h, u), *where* $h \leftarrow \mathcal{F}_1$ *and* $u \leftarrow \mathcal{F}_{t \bmod 2}$.

BIKE-Simple: Before showing the security, we consider the following deterministic PKE scheme, which we call BIKE-Simple:

- $\mathsf{Gen}(pp)$: $dk := (h_0, h_1) \leftarrow \mathcal{H}_w$. Output $ek = h := h_1 \cdot h_0^{-1} \in \mathcal{R}$ and dk.
- $\mathsf{Enc}(ek, (e_0, e_1) \in \mathcal{E}_t)$: Compute $u := e_0 + e_1 h \in \mathcal{R}$ and output u.
- $\mathsf{Dec}(dk, u)$: Output $(e_0, e_1) \leftarrow \mathsf{decode}(uh_0, (h_0, h_1))$.

The proposers showed that this scheme is OW-CPA-secure using appropriate assumptions as follows:

Lemma O.1 ([ABB+20, Theorem 1]). *If the 2-DQCCF and 2-CQCSD assumptions hold, then BIKE-Simple is* OW-CPA-*secure.*

Remark O.1. It is easy to show BIKE-Simple's disjoint simulatability: Let $\mathcal{F}_1$ be a ciphertext space. We define the simulator as sampling $u \leftarrow U(\mathcal{F}_1)$. Statistical disjointness follows from the fact that $|\mathcal{F}_1| \approx 2^r/2 \gg \binom{2r}{t} = |\mathcal{E}_t| \geq |\mathsf{Enc}(ek, \mathcal{E}_t)|$. We can show ciphertext indistinguishability by using the 2-DQCCF and 2-DQCSD assumptions. Applying SXY and assuming δ is negligible, we can obtain a *tightly* CCA-secure KEM scheme with shorter ciphertext, which leads to anonymous, robust hybrid PKE.

Security of BIKE-PKE: We next show that BIKE-PKE is ciphertext-indistinguishable in the QROM with a simulator that outputs $u \leftarrow \mathcal{F}_{t \bmod 2}$ and $v \leftarrow \mathbb{F}_2^{256}$.

Lemma O.2. *Assume that the 2-DQCCF and 2-DQCSD assumptions hold. Then, BIKE-PKE* PKE *is ciphertext-indistinguishable in the QROM.*

Proof (Proof Sketch). We consider four games Game_0, Game_1, Game_2, and Game_3:

- Game_0: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h := h_1 \cdot h_0^{-1}$.
  - Encryption: $\mu \leftarrow \mathbb{F}_2^{256}$, $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$and $v := \mu \oplus \mathsf{L}(e_0, e_1)$; return $c = (u, v)$.
- Game_1: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h := h_1 \cdot h_0^{-1}$.
  - Encryption: $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$; $v \leftarrow \mathbb{F}_2^{256}$; return $c = (u, v)$.
- Game_2: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_1$.
  - Encryption: $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$; $v \leftarrow \mathbb{F}_2^{256}$; return $c = (u, v)$.
- Game_3: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_1$.
  - Encryption: $u \leftarrow \mathcal{F}_{t \bmod 2}$; $v \leftarrow \mathbb{F}_2^{256}$; return $c = (u, v)$.
- Game_4: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h := h_1 \cdot h_0^{-1}$.
  - Encryption: $u \leftarrow \mathcal{F}_{t \bmod 2}$; $v \leftarrow \mathbb{F}_2^{256}$; return $c = (u, v)$.

Game_0 and Game_1 are equivalent, since μ in Game_0 and v in Game_1 is chosen uniformly at random. Game_1 and Game_2 are computationally indistinguishable because of the 2-DQCCF assumption. Game_2 and Game_3 are computationally indistinguishable because of the 2-DQCSD assumption. Game_3 and Game_4 are computationally indistinguishable because of the 2-DQCCF assumption. ◻

We next consider BIKE-PKE is IND-CPA-secure in the QROM. The proposers showed the security in the ROM as follows:

Lemma O.3 ([ABB+20, Theorem 2]). *If the 2-DQCCF and 2-CQCSD assumptions hold, then BIKE-PKE is* IND-CPA-*secure in the ROM.*

Unfortunately, applying their idea directly to the QROM setting, the security proof becomes loose since it will involve the O2H lemma (Lemma A.2). We here show the IND-CPA security of BIKE-PKE in the QROM *tightly* using the idea of [SXY18].

Lemma O.4. *Assume that the 2-DQCCF and 2-DQCSD assumptions hold and* PKE *is δ-correct. Then, BIKE-PKE* PKE *is* IND-CPA-*secure (and* OW-CPA-*secure) in the QROM.*

Proof (Proof Sketch). We consider $\mathsf{Game}_{i,b}$ for $b \in \{0, 1\}$ and $i = 0, \ldots, 4$ defined as follows:

- $\mathsf{Game}_{0,b}$: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h := h_1 \cdot h_0^{-1}$.

- Encryption given μ_0 and μ_1: $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$, $k := \mathsf{L}(e_0, e_1)$, and $v := \mu_b \oplus k$; return $c = (u, v)$.

– $\mathsf{Game}_{1,b}$: In this game, we use $\mathsf{L}_q : \mathcal{R} \to \{0,1\}^{256}$ and define $\mathsf{L}(e_0, e_1) = \mathsf{L}_q(he_0 + e_1)$. a public key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h := h_1 \cdot h_0^{-1}$.
  - Encryption given μ_0 and μ_1: $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$, $k := \mathsf{L}_q(u)$, and $v := \mu_b \oplus k$; return $c = (u, v)$.

– $\mathsf{Game}_{2,b}$: In this game, we use random h. A public key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_1$.
  - Encryption given μ_0 and μ_1: $(e_0, e_1) \leftarrow \mathcal{E}_t$; compute $u := e_0 + he_1$, $k := \mathsf{L}_q(u)$, and $v := \mu_b \oplus k$; return $c = (u, v)$.

– $\mathsf{Game}_{3,b}$: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_1$.
  - Encryption given μ_0 and μ_1: $u \leftarrow \mathcal{F}_{t \bmod 2}$; compute $k := \mathsf{L}_q(u)$, and $v := \mu_b \oplus k$; return $c = (u, v)$.

– $\mathsf{Game}_{4,b}$: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_1$.
  - Encryption given μ_0 and μ_1: $u \leftarrow \mathcal{F}_{t \bmod 2}$, $k \leftarrow \{0,1\}^{256}$; compute $v := \mu_b \oplus k$; return $c = (u, v)$.

$\mathsf{Game}_{0,b}$ and $\mathsf{Game}_{1,b}$ are equivalent if the mapping $(e_0, e_1) \mapsto he_0 + e_1$ is injective, which is satisfied if a key pair of PKE is correct. $\mathsf{Game}_{1,b}$ and $\mathsf{Game}_{2,b}$ are computationally indistinguishable because of the 2-DQCCF assumption. $\mathsf{Game}_{2,b}$ and $\mathsf{Game}_{3,b}$ are computationally indistinguishable because of the 2-DQCSD assumption. $\mathsf{Game}_{3,b}$ and $\mathsf{Game}_{4,b}$ are equivalent if u is in outside of the image of the mapping $(e_0, e_1) \mapsto e_0 + e_1 h$, which occurs with overwhelming probability. $\mathsf{Game}_{4,0}$ and $\mathsf{Game}_{4,1}$ are equivalent since k is uniformly at random. □

Remark O.2. We can replace the term δ with the probability that the mapping $(e_0, e_1) \mapsto e_0 + e_1 h$ is injective for random $h \leftarrow \mathcal{F}_1$.

Security of BIKE-DPKE: We then consider $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, which we call BIKE-DPKE.

Lemma O.5. *Assume that the 2-DQCCF and 2-DQCSD assumptions hold. Then, BIKE-DPKE* $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ *is disjointly-simulatable.*

Proof. Statistical disjointess follows from the fact that $|\mathcal{S}(1^\kappa)| \approx 2^r/2 \cdot 2^{n_1 n_2}$ and $|\mathsf{Enc}'(ek, \mathcal{M})| \leq 2^k$. Ciphertext indistinguishability follows from Theorem E.1 that states that T preserves the ciphertext indistinguishability (Lemma O.2) and onewayness (Lemma O.4) loosely. □

We next consider BIKE-DPKE's XCFR-security:

Lemma O.6. *Let ϵ_u be a probability that $h_0 - h_1 \notin \mathcal{R}^*$ holds for two randomly generated keys h_0 and h_1. Let ϵ_0 be a probability that an efficient adversary finds μ such that $e_1 = 0$ where $(e_0, e_1) := \mathcal{E}_t(\mathsf{G}(\mu))$. Suppose that and $\epsilon := \epsilon_u + \epsilon_0$ is negligible. Then,* $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$ *is* XCFR*-secure.*

Proof (Proof sketch:). Let us consider $ek_i = h_i$ and $dk_i = (h_0, h_1)$ for $i \in zo$. If the adversary outputs $c = (u, v)$, it should be decrypted into μ by using dk_0 and dk_1, respectively. Let $(e_0, e_1) = \mathcal{E}_t(\mathsf{G}(\mu))$. We have $u = e_0 + e_1 h_0 = e_0 + e_1 h_1$ in the re-encryption check. This implies $(h_0 - h_1) \cdot e_1 = 0 \in \mathcal{R}$. If $e_1 \neq 0$ and $h_0 - h_1 \in \mathcal{R}^*$, then this leads a contradiction. Thus, the lemma holds. □

Summary: Recall that $\mathsf{FO}^{\not\perp}$ is $\mathsf{U}^{\not\perp} \circ \mathsf{T}$. Applying $\mathsf{U}^{\not\perp}$ to $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, we obtain $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}, \mathsf{H}]$. After applying our theorems, we summarize the security properties of BIKE as follows:

– BIKE-DPKE PKE' is strongly disjointly-simulatable if the 2-DQCCF and 2-DQCSD assumptions hold (Lemma O.5). It is XCFR-secure if ϵ is negligible (Lemma O.6).
– Thus, BIKE is SPR-CCA-secure, SSMT-CCA-secure, and SCFR-CCA-secure in the QROM.
– BIKE is ANON-CCA-secure.
– BIKE leads to ANON-CCA-secure, SROB-CCA-secure hybrid PKE.

Table 16. Parameter sets of FrodoKEM in Round 3.

parameter sets	n	q	σ	s	B	$\bar{m}$	$\bar{n}$
Frodo-640	640	2^{15}	2.8	12	2	8	8
Frodo-976	976	2^{16}	2.3	10	3	8	8
Frodo-1344	1344	2^{16}	1.4	6	4	8	8

P FrodoKEM

Review of FrodoKEM: FrodoKEM [NAB+20] is an LWE-based KEM scheme in the alternates,
Let $q = 2^D$ for some $D \le 16$. For a positive integer $B < D$, $\bar{m}$, and $\bar{n}$, they use encode: $\{0,1\}^{B\bar{m}\bar{n}} \to \mathbb{Z}_q^{\bar{m}\times\bar{n}}$ and decode: $\{0,1\}^{B\bar{m}\bar{n}} \to \{0,1\}^{B\bar{m}\bar{n}}$. (Roughly speaking, they compute ec: $k \in [0, 2^B) \mapsto k \cdot q/2^B \in \mathbb{Z}_q$ and dc: $K \in \mathbb{Z}_q \mapsto \lceil K2^B/q \rfloor \bmod 2^B$ and arrange the result.) Let $\ell = B\bar{m}\bar{n}$ be a message length. They use a distribution χ_S that is a centered symmetric distribution whose support is $\{-s, -(s-1), \ldots, s-1, s\}$. (See [NAB+20, Sect.2.2.4 and Table 3] for the concrete distribution.) For concrete values, see Table 16.
The underlying PKE scheme of FrodoKEM, which we call FrodoKEM-PKE, is summarized as follows:

- $\mathsf{Gen}(pp)$: Choose $A \leftarrow \mathbb{Z}_q^{n\times n}$, $S \leftarrow \chi^{n\times\bar{n}}$ and $E \leftarrow \chi^{n\times\bar{n}}$. Compute $B := AS + E$. Output $ek := (A, B)$ and $dk := S$.
- $\mathsf{Enc}(ek, \mu)$: Choose $S', E' \leftarrow \chi^{\bar{m}\times n}$ and $E'' \leftarrow \chi^{\bar{m}\times\bar{n}}$. Output $c = (U, V) := (S'A+E', S'B+E''+\mathsf{encode}(\mu))$.
- $\mathsf{Dec}(dk = S, (U, V))$: Compute $M := V - U \cdot S$ and output $\mu' := \mathsf{decode}(M)$.

We next consider an intermediate PKE scheme $\mathsf{PKE}_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call FrodoKEM-PKE-PRG:

- $\mathsf{Gen}_0(pp) = \mathsf{Gen}(pp)$:
- $\mathsf{Enc}_0(ek, \mu; r)$: Use $\rho = \mathsf{SHAKE128}_X(\mathtt{0x96}\|r)$ to sample S', E', E''. Output $c = (U, V) := (S'A + E', S'B + E'' + \mathsf{encode}(\mu))$.
- $\mathsf{Dec}_0(dk, (U, V)) = \mathsf{Dec}(dk, (U, V))$:

FrodoKEM applies $\mathsf{FO}^{\not\perp\prime\prime}$ to ForodoKEM-PKE-PRG, where $\mathsf{H}' = \mathsf{SHAKE}$, $\mathsf{G} = \mathsf{SHAKE}$, and $\mathsf{H} = \mathsf{SHAKE}$: We can treat them as different random oracles because their input length differ.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, h, s)$
$(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}_0(dk, c)$
$h \leftarrow \mathsf{H}'(ek)$	$(\bar{K}, r) := \mathsf{G}(\mu, \mathsf{H}'(ek))$	$(\bar{K}', r') := \mathsf{G}(\mu', h)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$c := \mathsf{Enc}_0(ek, \mu; r)$	$c' := \mathsf{Enc}_0(ek, \mu'; r')$
$\overline{dk} := (dk, ek, h, s)$	$K := \mathsf{H}(\bar{K}, c)$	**if** $c \ne c'$, **then return** $K := \mathsf{H}(s, c)$
return $(ek, \overline{dk})$	**return** (c, K)	**else return** $K := \mathsf{H}(\bar{K}', c)$

Security: Grubbs et al. [GMP21a] fortunately show the security of $\mathsf{FO}^{\not\perp\prime\prime}$. Thus, we can apply their result to FrodoKEM.

Q HQC

Review of HQC: HQC [AAB+20] is another code-based KEM scheme in the alternate candidates.
Let $\mathcal{R} := \mathbb{F}_2[x]/(x^r - 1)$. Let C be a decodable $[n_1n_2, k]$ code generated by $G \in \mathbb{F}_2^{k\times n_1n_2}$, where $n_1n_2 \le r$. Let decode be a decoder algorithm which corrects an error up to δ. Let $\mathcal{S}_w := \{x \in \mathcal{R} \mid \mathsf{HW}(x) = w\}$. For a polynomial $A = \sum_i a_i x^i \in \mathcal{R}$, we define $\mathsf{trunc}(A, l) = (a_0, \ldots, a_{l-1}) \in \mathbb{F}_2^l$. For concrete values, see Table 17.
The underlying PKE scheme of HQC, which we call HQC-PKE, is summarized as follows:

- $\mathsf{Gen}(pp)$: $h_0 \leftarrow \mathcal{R}$. $(x, y) \leftarrow \mathcal{S}_w^2$. Compute $h_1 := x + h_0 y$. Output $dk := (x, y)$ and $ek := (h_0, h_1)$.
- $\mathsf{Enc}(ek, \mu \in \mathbb{F}_2^k; (e, f, t) \in \mathcal{S}_{w_e} \times \mathcal{S}_{w_r} \times \mathcal{S}_{w_r})$: Output

$$c = (u, v) := (h_0 t + f, \mathsf{trunc}(h_1 t + e, n_1 n_2) \oplus \mu G) \in \mathcal{R} \times \mathbb{F}_2^{n_1 n_2}.$$

Table 17. Parameter sets of HQC in Round 3.

parameter sets	r	n_1	k_1	d_1	n_2	k_2	d_2	w	w_e	w_r
hqc-128	17,669	46	16	31	384	8	192	66	75	75
hqc-192	35,851	56	24	32	640	8	320	100	114	114
hqc-256	57,637	90	32	59	640	8	320	131	149	149

– $\mathsf{Dec}(dk, (u, v))$: Compute $a := v \oplus \mathsf{trunc}(uy, n_1 n_2) \in \mathbb{F}_2^{n_1 n_2}$ and output $\mathsf{decode}(a)$.

We next consider an intermediate PKE scheme $\mathsf{PKE}_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call HQC-PKE-PRG:

– $\mathsf{Gen}_0(pp) = \mathsf{Gen}(pp)$:
– $\mathsf{Enc}_0(ek, \mu; r)$: Use $\rho = \mathsf{SHAKE256}(r, \mathtt{0x02})$ to sample $(e, f, t) \in \mathcal{S}_{w_e} \times \mathcal{S}_{w_r} \times \mathcal{S}_{w_r}$. Output $(u, v) := \mathsf{Enc}(ek, \mu; (e, f, t))$.
– $\mathsf{Dec}_0(dk, (u, v)) = \mathsf{Dec}(dk, (u, v))$:

HQC applies $\mathsf{HFO}^{\perp}$ to HQC-PKE-PRG PKE_0, where $\mathsf{G}(\mu) = \mathsf{SHAKE256}_{512}(\mu, \mathtt{0x03})$, $\mathsf{F}(\mu) = \mathsf{SHAKE256}_{512}(\mu, \mathtt{0x04})$. and $\mathsf{H}(\mu, (c_0, c_1)) = \mathsf{SHAKE256}_{512}(\mu, \mathtt{0x05})$. We can treat them as different random oracles because their input length differ.

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek)$
$(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}_0(dk, c_0)$
$\overline{dk} := (dk, ek)$	$r := \mathsf{G}(\mu)$	**if** $\mu' = \perp$, **then return** $K := \perp$
return $(ek, \overline{dk})$	$c_0 := \mathsf{Enc}_0(ek, \mu; r)$	$r' := \mathsf{G}(\mu')$
	$c_1 := \mathsf{F}(\mu)$	$c'_0 := \mathsf{Enc}_0(ek, \mu'; r')$
	$K := \mathsf{H}(\mu, c_0, c_1)$	$c'_1 := \mathsf{F}(\mu')$
	return $((c_0, c_1), K)$	**if** $(c_0, c_1) \neq (c'_0, c'_1)$, **then return** $K := \perp$
		else return $K := \mathsf{H}(\mu', c_0, c_1)$

Q.1 Security

Grubbs et al. [GMP21a] show properties of a variant of $\mathsf{HFO}^{\perp}$, in which $c_1 = \mathsf{F}(\mu, c_0)$ instead of $c_1 = \mathsf{F}(\mu)$. We here show HQC's anonymity directly by using properties of $\mathsf{HFO}^{\perp} = \mathsf{HU}^{\perp} \circ \mathsf{T}$.

Assumptions: For $b \in \{0,1\}$, define the finite set $\mathcal{F}_b := \{h \in \mathcal{R} : h(1) \equiv b \pmod 2\}$, that is, a set of all binary vectors of length r and parity b. Similarly, for $b, b_0, b_1 \in \{0,1\}$, we define the sets

$$\mathcal{F}_b^{1,2} := \{H = [1, h] \in \mathcal{R}^2 : h \in \mathcal{F}_b\}$$
$$\mathcal{F}_{b_0,b_1}^{2,3} := \left\{H = \begin{bmatrix} 1 & 0 & h_0 \\ 0 & 1 & h_1 \end{bmatrix} \in \mathcal{R}^{2\times 3} : h_0 \in \mathcal{F}_{b_0} \wedge h_1 \in \mathcal{F}_{b_1}\right\}.$$

Definition Q.1 (The 2-Decisional Quasi-Cyclic Syndrome Decoding (2-DQCSD) assumption [AAB+20]). *Fix $b \in \{0,1\}$, w, and $b' := w + bw \bmod 2$. For any (Q)PPT adversary, it is hard to distinguish the following two distributions:*
– *$(H, H \cdot (x, y))$, where $H \leftarrow \mathcal{F}_b^{1,2}$ and $(x_1, x_2) \leftarrow \mathcal{S}_w^2$.*
– *(H, z), where $H \leftarrow \mathcal{F}_b^{1,2}$ and $y \leftarrow \mathcal{F}_{b'}$.*

Definition Q.2 (The 3-Decisional Quasi-Cyclic Syndrome Decoding (3-DQCSD) assumption [AAB+20]). *Fix $b_0, b_1 \in \{0,1\}$, w. Let $b'_0 := w + b_0 w \bmod 2$ and $b'_1 := w + b_1 w \bmod 2$. For any (Q)PPT adversary, it is hard to distinguish the following two distributions:*
– *$(H, H \cdot (x_0, x_1, x_2))$, where $H \leftarrow \mathcal{F}_{b_0,b_1}^{2,3}$ and $(x_0, x_1, x_2) \leftarrow \mathcal{S}_w^3$.*
– *$(H, (z_0, z_1))$, where $H \leftarrow \mathcal{F}_{b_0,b_1}^{2,3}$, $z_0 \leftarrow \mathcal{F}_{b'_0}$, and $z_1 \leftarrow \mathcal{F}_{b'_1}$.*

For collision-freeness, we define the following new assumption:

Definition Q.3 (The 3-Computational Quasi-Cyclic Codeword Finding (3-CQCCF) assumption). *For any (Q)PPT adversary, given $(1, h, h')$ where $h, h' \leftarrow \mathcal{R}$, it is hard to find a non-zero codeword (f, t, t') whose Hamming weight is at most $4w_r$.*

Security of HQC-PKE: Using those assumptions, the proposers show the IND-CPA security of HQC-PKE:

Lemma Q.1 ([AAB+20, Theorem 5.1], adapted). *Assume that the 2-DQCSD and 3-DQCSD assumptions hold. Then, the HQC-PKE* PKE *is* IND-CPA*-secure (and* OW-CPA*-secure).*

By mimicking their proof, we can show that it is ciphertext-indistinguishable with a simulator that outputs $u \leftarrow \mathcal{F}_{b_0}$ and $v \leftarrow \mathbb{F}_2^{n_1 n_2}$, where $b_0 := (1 + h_0(1))w_r \bmod 2$.

Lemma Q.2. *Assume that the 2-DQCSD and 3-DQCSD assumptions hold. Then, the underlying PKE* PKE *is ciphertext-indistinguishable.*

Proof (Proof Sketch). We consider four games Game_0, Game_1, Game_2, and Game_3:
In what follows, we define the parity of h_1 as $b := (1+h_0(1))w \bmod 2$, the parity of u as $b_0 := (1+h_0(1))w_r \bmod 2$, and the parity of $\tilde{v}$ as $b_1 := w_e + bw_r \bmod 2$:

- Game_0: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}$, $x, y \leftarrow \mathcal{S}_w$, and $h_1 := x + h_0 y$.
  - Encryption: $\mu \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathcal{S}_{w_e}$, $t, f \leftarrow \mathcal{S}_{w_r}$, and compute $u := h_0 t + f$ and $v := \mathsf{trunc}(h_1 t + e, n_1 n_2) \oplus \mu G$.
- Game_1: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}$, $h_1^+ \leftarrow \mathcal{F}_b$.
  - Encryption: $\mu \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathcal{S}_{w_e}$, $t, f \leftarrow \mathcal{S}_{w_r}$, and compute $u := h_0 t + f$ and $v := \mathsf{trunc}(h_1^+ t + e, n_1 n_2) \oplus \mu G$.
- Game_2: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}$, $h_1^+ \leftarrow \mathcal{F}_b$.
  - Encryption: $\mu \leftarrow \mathbb{F}_2^k$, $e \leftarrow \mathcal{F}_{w_e}$, $t, f \leftarrow \mathcal{F}_{w_r}$, and compute $u := h_0 t + f$ and $v := \mathsf{trunc}(h_1^+ t + e, n_1 n_2) \oplus \mu G$.
- Game_3: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}$, $h_1^+ \leftarrow \mathcal{F}_{2,b}$.
  - Encryption: $u \leftarrow \mathcal{F}_{b_0}$ and $v \leftarrow \mathbb{F}_2^{n_1 n_2}$.
- Game_4: In this game, a public key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}$, $x, y \leftarrow \mathcal{S}_w$, and $h_1 := x + h_0 y$.
  - Encryption: $u \leftarrow \mathcal{F}_{b_0}$ and $v \leftarrow \mathbb{F}_2^{n_1 n_2}$.

Game_0 and Game_1 are computationally indistinguishable because of the 2-DQCSD assumption. Game_1 and Game_2 are computationally indistinguishable because of the 3-DQCSD assumption. Game_2 and Game_3 are statistically indistinguishable, because trunc truncates $r - n_1 n_2$ bits of $\tilde{y} := h_1^+ t + e$ in Game_2 and thus, $\mathsf{trunc}(\tilde{y}, n_1 n_2)$'s distribution is statistically close to the uniform distribution over $\mathbb{F}_2^{n_1 n_2}$. Game_3 and Game_4 are computationally indistinguishable because of the 2-DQCSD assumption. ◻

Let us compute the parity of h_1, $b := (1 + h_0(1))w \bmod 2$ and the parity of u, $b_0 := (1 + h_0(1))w_r \bmod 2$. According to Table 17, we obtain that the parity b of h_1 is 0, 0, $1 - h_0(1)$ and the parity b_0 of u is 1, 0, $h_0(1)$, for `HQC-128/192/256`, respectively. We can say that `HQC-128` and `HQC-192` are SPR-CPA secure, while `HQC-256` is not strong. Indeed, the parity of u leaks the information of h_0 of the encryption key.

Security of HQC-PKE-PRG and HQC-DPKE: We next consider HQC-PKE-PRG PKE_0, whose encryption algorithm uses a PRG $\mathsf{SHAKE256}(\cdot, \mathtt{0x02})$ instead of true randomness. The IND-CPA security and ciphertext indistinguishability of PKE_0 follows from PRG's quantum security tightly.

Lemma Q.3. *Assume that the 2-DQCSD and 3-DQCSD assumptions hold and* $\mathsf{SHAKE256}(\cdot, \mathtt{0x02})$ *is quantumly-secure PRG. Then, HQC-PKE-PRG* PKE_0 *is ciphertext-indistinguishable and* IND-CPA*-secure (and* OW-CPA*-secure).*

We then consider $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$, which we call HQC-DPKE.

Lemma Q.4. *Assume that the 2-DQCSD and 3-DQCSD assumptions hold and* $\mathsf{SHAKE256}(\cdot, \mathtt{0x02})$ *is quantumly-secure PRG. Then,* $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$ *is disjointly-simulatable.*

Proof. Statistical disjointess follows from the fact that $|\mathcal{S}(1^\kappa)| \approx 2^r/2 \cdot 2^{n_1 n_2}$ and $|\mathsf{Enc}'(ek, \mathcal{M})| \leq 2^k$. Ciphertext indistinguishability follows from Theorem E.1 that states that T preserves ciphertext indistinguishability and onewayness of PKE_0 (Lemma Q.3). ◻

Robustness of HQC: We finally consider HQC's SROB-CCA-security:

Lemma Q.5. *Suppose that the 3-CQCCF assumption holds. Then, HQC is* SROB-CCA*-secure.*

Proof (Proof sketch:). Given $(1, h_{0,0}, h_{1,0})$ with $h_{0,0}, h_{1,0} \leftarrow \mathcal{R}$, we generate decryption keys and encryption keys $ek_i = (h_{i,0}, h_{i,1})$ and $dk_i = (x_i, y_i)$ for $i \in zo$. We give them to an adversary against SROB-CCA security of KEM. Suppose that the adversary outputs $c = (u, v)$ and the adversary wins. If so, it should be decapsulated into $K_0 \neq \bot$ and $K_1 \neq \bot$. Thus, c should be decrypted into μ_0 and μ_1 by using dk_0 and dk_1, respectively. In re-encryption check, we have $(e_0, f_0, t_0) := \mathsf{SHAKE256}(\mathsf{G}(\mu_0), \mathtt{0x02})$ and $(e_1, f_1, t_1) := \mathsf{SHAKE256}(\mathsf{G}(\mu_1), \mathtt{0x02})$, and $u = h_{0,0}t_0 + f_0 = h_{1,0}t_1 + f_1$. This implies $(1, h_{0,0}, h_{1,0}) \cdot (f_0 + f_1, t_0, t_1) = 0$ and $(f_0 + f_1, t_0, t_1)$ is the solution of the 3-CQCCF problem. □

Summary: Recall that $\mathsf{HFO}^{\bot}$ is $\mathsf{HU}^{\bot} \circ \mathsf{T}$. Applying $\mathsf{HU}^{\bot}$ to $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$, we obtain $\mathsf{KEM} = \mathsf{HU}^{\bot}[\mathsf{PKE}_0, \mathsf{H}]$. After applying our theorems, we summarize the security properties of HQC as follows:
- HQC-DPKE PKE' is disjointly-simulatable if the 2-DQCSD and 3-DQCSD assumptions hold (Lemma Q.4). HQC-DPKE for `HQC-128` and `HQC-192` are *strongly* disjointly-simulatable.
- Thus, `HQC-128` and `HQC-192` are SPR-CCA-secure and $1/2^{512}$-sparse in the QROM.
- HQC is SROB-CCA-secure if the 3-CQCCF assumption holds.
- `HQC-128` and `HQC-192` are ANON-CCA-secure.
- `HQC-128` and `HQC-192` lead to ANON-CCA-secure, SROB-CCA-secure hybrid PKE.

R Streamlined NTRU Prime

Streamlined NTRU Prime is one of two KEMs in NTRU Prime [BBC+20].

Review of Streamlined NTRU Prime: Streamlined NTRU Prime (sntrupr) has parameter sets p, q, and w. p and q are prime numbers and w is a positive integer. We note that $2p \geq 3w$ and $q \geq 16w + 1$. They choose $q = 6q' + 1$ for some q'. For concrete values, see Table 18.

Table 18. Parameter sets of sntrupr of NTRU Prime

parameter sets	p	q	w
`sntrupr653`	653	4621	288
`sntrupr761`	761	4591	286
`sntrupr857`	857	5167	322
`sntrupr953`	953	6343	396
`sntrupr1013`	1013	7177	448
`sntrupr1277`	1277	7879	492

Let $\mathcal{R} := \mathbb{Z}[x]/(x^p - x - 1)$ and $\mathcal{R}_a := (\mathbb{Z}/a)[x]/(x^p - x - 1)$ for $a = 3, q$. Let $\mathcal{S} := \{a = \sum_{i=0}^{p-1} a_i x^i \in \mathcal{R} \mid a_i \in \{-1, 0, +1\}\}$, a set of ternary polynomials. Let $\mathcal{S} := \{a = \sum_{i=0}^{p-1} a_i x^i \in \mathcal{R} \mid a_i \in \{-1, 0, +1\}, \mathsf{HW}(a) = w\}$, a set of "short" polynomials. For $a \in [-(q-1)/2, (q-1)/2]$, define $\mathsf{Round}(a) = 3 \cdot \lfloor a/3 \rceil$.[5]
The underlying CPA-secure PKE scheme[6] works as follows:
- $\mathsf{Gen}(pp)$: Choose $g \leftarrow \mathcal{S}$ that satisfies $g \in \mathcal{R}_3^\times$ at random. Compute $1/g \in \mathcal{R}_3$. Choose $f \leftarrow \mathcal{S}$. Compute $h := g/(3f) \in \mathcal{R}_q$. Output $ek := h$ and $dk := (f, 1/g)$.
- $\mathsf{Enc}(ek, r \in \mathcal{S})$: Compute $hr \in \mathcal{R}_q$ and output $c := \mathsf{Round}(hr \bmod^{\pm} q)$.
- $\mathsf{Dec}(dk = (f, v), c)$: Compute $e := (3fc \bmod^{\pm} q) \bmod^{\pm} 3$. Compute $r' := ev \bmod^{\pm} 3$. Output r' if $\mathsf{HW}(r') = w$. Otherwise, output $r'_{\mathsf{invalid}} := (1, 1, \dots, 1, 0, \dots, 0)$ with $\mathsf{HW}(r'_{\mathsf{invalid}}) = w$.

Due to rounding, we have a 'short' error m such that $c = hr + m$.
Streamlined NTRU Prime [BBC+20] used $\mathsf{HU}^{\not\bot,\mathsf{prf}}$, where $\mathsf{H}(\mu, c) = \mathsf{SHA512}_{256}(\mathtt{0x01}, \mathsf{SHA512}_{256}(\mathtt{0x03}, \mu), c)$ $\mathsf{H}_{\mathsf{prf}}(s, c) = \mathsf{SHA512}_{256}(\mathtt{0x00}, \mathsf{SHA512}_{256}(\mathtt{0x03}, s), c)$ $\mathsf{F}(\mu, ek) = \mathsf{SHA512}_{256}(\mathtt{0x02}, \mathsf{SHA512}_{256}(\mathtt{0x03}, \mu), \mathsf{SHA512}_{256}(\mathtt{0x04}, ek))$.

[5] When $q = 6q' + 1$, $\mathsf{Round}([-(q-1)/2, (q-1)/2]) \in [-(q-1)/2, (q-1)/2]$.
[6] 'Streamlined NTRU Prime Core' in the specification.

$\mathsf{Gen}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' := \mathsf{Dec}(dk, c_0)$
$s \leftarrow \{0,1\}^l$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \bot$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{F}(\mu, ek)$	$c_0' := \mathsf{Enc}(ek, \mu')$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c_0, c_1)$	$c_1' := \mathsf{F}(\mu', ek)$
	return $((c_0, c_1), K)$	**if** $(c_0, c_1) = (c_0', c_1')$, **then return** $K := \mathsf{H}(\mu', c_0, c_1)$
		else return $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$

Security: We found that Streamlined NTRU Prime has a problem. For simplicity, let $\mathsf{H}_i(x) = \mathsf{SHA512}_{256}(\mathtt{0x0i}\|x)$ as in [BBC+20]. Using this notation, we have

- $\mathsf{H}(\mu, c) = \mathsf{H}_1(\mathsf{H}_3(\mu)\|c)$
- $\mathsf{H}_{\mathsf{prf}}(s, c) = \mathsf{H}_0(\mathsf{H}_3(s)\|c)$
- $\mathsf{F}(\mu, ek) = \mathsf{H}_2(\mathsf{H}_3(\mu)\|\mathsf{H}_4(ek))$.

Formally speaking, the conversion of Streamlined NTRU Prime is summarized as follows Figure 19:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \mathcal{M}$	$\mu' := \mathsf{Dec}(dk, c_0)$
$s \leftarrow \{0,1\}^\ell$	$c_0 := \mathsf{Enc}(ek, \mu)$	**if** $\mu' = \bot$, **then return** $K := \mathsf{H}_0(\mathsf{H}_3(s), c_0, c_1)$
$\overline{dk} := (dk, ek, s)$	$c_1 := \mathsf{H}_2(\mathsf{H}_3(\mu), \mathsf{H}_4(ek))$	$c_0' := \mathsf{Enc}(ek, \mu')$
return $(ek, \overline{dk})$	$K := \mathsf{H}_1(\mathsf{H}_3(\mu), c_0, c_1)$	$c_1' := \mathsf{H}_2(\mathsf{H}_3(\mu'), \mathsf{H}_4(ek))$
	return $((c_0, c_1), K)$	**if** $(c_0, c_1) \neq (c_0', c_1')$, **then return** $K := \mathsf{H}_0(\mathsf{H}_3(s), c_0, c_1)$
		else return $K := \mathsf{H}_1(\mathsf{H}_3(\mu'), c_0, c_1)$

Fig. 19. $\mathsf{KEM} = \mathsf{HU}^{\not\bot,\mathsf{prf},\prime}[\mathsf{PKE}, \mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4]$.

We can assume H_i as random oracles. The IND-CCA security proof in the ROM is straightforward because, intuitively speaking, the adversary cannot distinguish real K with random one unless it asks μ of c_0 to H_3 and the simulation of decapsulation is done by the list of queries to the random oracles. Unfortunately, we have a technical obstacle for the IND-CCA security in the QROM.

We have tried to show its security via an intermediate transformation $\mathsf{HU}^{\bot,\prime}[\mathsf{PKE}, \mathsf{H}_2, \mathsf{H}_3, \mathsf{H}_4]$, in which $K = \mathsf{H}_3(\mu)$ and the decapsulation algorithm returns $\bot$ for a invalid ciphertext. If this was secure, then we can convert the security proof of $\mathsf{HU}^{\bot,\prime}$ into that of $\mathsf{HU}^{\not\bot,\mathsf{prf},\prime}$ using a simple reduction. Unfortunately, $\mathsf{HU}^{\bot,\prime}$ is not IND-CPA-secure because $c_1 = \mathsf{H}_2(\mathsf{H}_3(\mu), \mathsf{H}_4(ek)) = \mathsf{H}_2(K, \mathsf{H}_4(ek))$ and we can check if K is real by checking if $c_1 = \mathsf{H}_2(K, \mathsf{H}_4(ek))$ or not.

If H_3 is length-preserving, we could use the technique by Grubbs et al. [GMP21a] for QROM security proof. Unfortunately, μ is longer than 256-bits and this is not length-preserving.

If F is not nested on μ, we can prove the security as follows: We first consider $\mathsf{HU}_m^{\bot}[\mathsf{PKE}, \mathsf{H}_3, \mathsf{F}]$, which is SPR-CCA-secure if PKE is strongly disjoint-simulatable. We then consider an indifferentiable reduction defined as follows: if $K \neq \bot$, then we rewrite the decapsulation result as $\mathsf{H}_1(K\|c)$; if $K = \bot$, then we rewrite the decapsulation result as $\mathsf{H}_0(\mathsf{H}_3(s)\|c)$. It is easy to see $\mathsf{HU}^{\not\bot,\mathsf{prf}}[\mathsf{PKE}, \mathsf{H}, \mathsf{F}, \mathsf{H}_{\mathsf{prf}}]$ is SPR-CCA-secure if $\mathsf{HU}_m^{bot}[\mathsf{PKE}, \mathsf{H}_3, \mathsf{F}]$ is SPR-CCA-secure.

Bernstein [Ber21] suggests to use the domain extension of quantum random oracles in [Zha19, Section 5], which is shown *quantumly indifferentiable.* Let $C^{\mathsf{H}_1,\mathsf{H}_2}(x, y) = \mathsf{H}_1(\mathsf{H}_2(x), y)$. Roughly speaking, we say $C^{\mathsf{H}_1,\mathsf{H}_2}$ is indifferentiable if any efficient adversary cannot distinguish oracles $\mathsf{H}_1, \mathsf{H}_2, C^{\mathsf{H}_1,\mathsf{H}_2}$ from $\mathsf{Sim}^{\mathsf{H}}, \mathsf{H}$, where Sim queries to H and simulates H_1 and H_2. We did not check the detail and leave to show the IND-CCA security (and anoymity) in the QROM as an open problem.

It might be interesting to study anonymity and robustness in the ROM.

S NTRU LPRime

NTRU LPRime is the other KEM in NTRU Prime [BBC+20].

Review of NTRU LPRime: NTRU LPRime has parameter sets $p, q, w, \delta, \tau_0, \tau_1, \tau_2$, and τ_3. We note that $q = 6q' + 1$ for some q' and $q \geq 16w + 2\delta + 3$. For concrete values, see Table 19.

Table 19. Parameter sets of ntrulpr of NTRU Prime

parameter sets	p	q	w	δ	τ_0	τ_1	τ_2	τ_3
`ntrulpr653`	653	4621	252	289	2175	113	2031	290
`ntrulpr761`	761	4591	250	292	2156	114	2007	287
`ntrulpr857`	857	5167	281	329	2433	101	2265	324
`ntrulpr953`	953	6343	345	404	2997	82	2798	400
`ntrulpr1013`	1013	7177	392	450	3367	73	3143	449
`ntrulpr1277`	1277	7879	429	502	3724	66	3469	496

Let $\mathcal{R} := \mathbb{Z}[x]/(x^p - x - 1)$ and $\mathcal{R}_q := \mathbb{Z}_q[x]/(x^p - x - 1)$. Let $\mathcal{S} := \{a = \sum_{i=0}^{p-1} a_i x^i \in \mathcal{R} \mid a_i \in \{-1, 0, +1\}, \mathsf{HW}(a) = w\}$, a set of "short" polynomials.
For $a \in [-(q-1)/2, (q-1)/2]$, define $\mathsf{Round}(a) = 3 \cdot \lceil a/3 \rfloor$.[7] For a polynomial $A = \sum_i a_i x^i \in \mathcal{R}_q$, we define $\mathsf{trunc}(A, l) = (a_0, \ldots, a_{l-1}) \in \mathbb{Z}_q^l$. For $C \in [0, q)$, define $\mathsf{Top}(C) = \lfloor (\tau_1 (C + \tau_0) + 2^{14})/2^{15} \rfloor$. For $T \in [0, 16)$, define $\mathsf{Right}(T) = \tau_3 T - \tau_2 \in \mathbb{Z}_q$. For $a \in \mathbb{Z}$, define $\mathsf{Sign}(a) = 1$ if $a < 0$, 0 otherwise.
The underlying CPA-secure PKE scheme[8] PKE works as follows:

- $\mathsf{Gen}(pp)$: Generate $A \leftarrow \mathcal{R}_q$ and $dk \leftarrow \mathcal{S}$. Compute $B := \mathsf{Round}(A \cdot dk)$. Output $ek := (A, B)$ and dk.
- $\mathsf{Enc}(ek, \mu \in \{0,1\}^{256})$: Choose $t \leftarrow \mathcal{S}$ and output

$$(U, V) := \big(\mathsf{Round}(t \cdot A), \mathsf{Top}(\mathsf{trunc}(t \cdot B, 256) + \mu(q-1)/2)\big).$$

- $\mathsf{Dec}(dk, (U, V))$: Compute $r := \mathsf{Right}(V) - \mathsf{trunc}(dk \cdot U, 256) + (4w + 1) \cdot \mathbf{1}_{256} \in \mathbb{Z}^{256}$ and outputs $\mu := \mathsf{Sign}(r \bmod^{\pm} q)$.

We next consider an intermediate PKE scheme $\mathsf{PKE}_0 = (\mathsf{Gen}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which is called as "NTRU LPRime Expand":

- $\mathsf{Gen}_0(pp) = \mathsf{Gen}(pp)$:
- $\mathsf{Enc}_0(ek, \mu; r)$: Use $\rho = \mathsf{AES256\text{-}CTR}(r)$ to sample $t \leftarrow \mathcal{S}$. Output $(U, V) := \mathsf{Enc}(ek, \mu; t)$.
- $\mathsf{Dec}_0(dk, (U, V)) = \mathsf{Dec}(dk, (U, V))$:

NTRU LPRime applies $\mathsf{HFO}_{\not\perp,\mathsf{prf}}$ to NTRU LPRime Expand PKE_0, where $\mathsf{G}(\mu) = \mathsf{SHA512}_{256}(\mathtt{0x05}, \mu)$, $\mathsf{H}(\mu, c) = \mathsf{SHA512}_{256}(\mathtt{0x01}, \mu, c)$, $\mathsf{H}_{\mathsf{prf}}(s, c) = \mathsf{SHA512}_{256}(\mathtt{0x00}, s, c)$, $\mathsf{F}(\mu, ek) = \mathsf{SHA512}_{256}\big(\mathtt{0x02}, \mu, \mathsf{SHA512}_{256}(\mathtt{0x04}, ek)\big)$:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}_0(1^\kappa)$	$\mu \leftarrow \{0,1\}^{\ell(\kappa)}$	$\mu' := \mathsf{Dec}_0(dk, c_0)$
$s \leftarrow \{0,1\}^{\ell(\kappa)}$	$r := \mathsf{G}(\mu)$	$r' := \mathsf{G}(\mu')$
$\overline{dk} := (dk, ek, s)$	$c_0 := \mathsf{Enc}_0(ek, \mu; r)$	$c_0' := \mathsf{Enc}_0(ek, \mu'; r')$
return $(ek, \overline{dk})$	$c_1 := \mathsf{F}(\mu, ek)$	$c_1' := \mathsf{F}(\mu', ek)$
	$K := \mathsf{H}(\mu, c_0, c_1)$	**if** $(c_0, c_1) \neq (c_0', c_1')$, **then return** $K := \mathsf{H}_{\mathsf{prf}}(s, c_0, c_1)$
	return $((c_0, c_1), K)$	**else return** $K := \mathsf{H}(\mu', c_0, c_1)$

[7] When $q = 6q' + 1$, $\mathsf{Round}([-(q-1)/2, (q-1)/2]) \in [-(q-1)/2, (q-1)/2]$.
[8] 'NTRU LPRime Core' in the specification.

S.1 Security

We directly assume that $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$ is strongly disjoint-simulatable. Recall that $\mathsf{HFO}_{\not\perp,\mathsf{prf}}$ is $\mathsf{HU}^{\not\perp,\mathsf{prf}} \circ \mathsf{T}$. Applying $\mathsf{HU}^{\not\perp,\mathsf{prf}}$ to $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}_0, \mathsf{G}]$, we obtain $\mathsf{KEM} = \mathsf{HU}^{\not\perp,\mathsf{prf}}[\mathsf{PKE}', \mathsf{H}, \mathsf{F}]$. After applying our theorems, we summarize the security properties of NTRU LPRime as follows:

- Assume that the underlying DPKE of NTRU LPRime PKE' is strongly disjointly-simulatable with simulator that samples $a \leftarrow \mathcal{R}$, computes $U := \mathsf{Round}(a)$, samples $V \leftarrow (\mathbb{Z}/16\mathbb{Z})^{256}$, and outputs (U, V).
- Then, NTRU LPRime is SPR-CCA-secure and SSMT-CCA-secure in the QROM.
- NTRU LPRime is SCFR-CCA-secure if the colliding probability of ek is negligible since F takes μ and ek as input.
- NTRU LPRime is ANON-CCA-secure.
- NTRU LPRime leads to ANON-CCA-secure, SROB-CCA-secure hybrid PKE.

T SIKE

Brief Review of SIKE: SIKE [JAC+20] is KEM scheme based on SIDH [JD11, ?]. For a survey of isogeny-based cryptography, we recommend reading [?].
Let $p = 2^{e_2}3^{e_3} - 1$. Let E be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Let $P_2, Q_2 \in E[2^{e_2}]$ and $P_3, Q_3 \in E[3^{e_3}]$ linearly independent points of order 2^{e_2} and 3^{e_3} respectively. Let $\{0,1\}^n$ be a message space and let $\mathsf{L} : \mathbb{F}_{p^2} \to \{0,1\}^n$ be a random oracle, instantiated by $\mathsf{SHAKE256}_n(\cdot)$.
Roughly speaking, the underlying PKE scheme [JAC+20, Algorithm 1], which we call SIKE-PKE, is summarized as follows (for the details, see the specification):

- $\mathsf{isogen}_\ell(dk_\ell)$ with $(m, \ell) = (2,3)$ or $(3,2)$: On input $dk_\ell \in [0, \ell^{e_\ell})$, compute $S := P_\ell + [dk_\ell]Q_\ell$, compute isogeny $\phi_\ell \colon E \to E/\langle S\rangle$, and compute $E'_m := E/\langle S\rangle = \phi_\ell(E)$. Compute $P'_m := \phi_\ell(P_m)$ and $Q'_m := \phi_\ell(Q_m)$. Output $ek_\ell := (E'_m, P'_m, Q'_m)$.[9]
- $\mathsf{isoex}_\ell(ek_m, dk_\ell)$ with $(m, \ell) = (2,3)$ or $(3,2)$: On input $ek_m = (E'_\ell, P'_\ell, Q'_\ell)$ and $dk_\ell \in [0, \ell^{e_\ell})$, compute $S := P'_\ell + [dk_\ell]Q'_\ell$ and compute $E''_\ell := E'_\ell/\langle S\rangle = E'_\ell/\langle \phi_m(P_\ell + [dk_\ell]Q_\ell)\rangle$. Compute j_ℓ as the j-invariant of E''_ℓ.
- $\mathsf{Gen}(pp)$: Choose $dk_3 \leftarrow [0, 3^{e_3})$ and $ek_3 := \mathsf{isogen}_3(dk_3)$. Output ek_3 and dk_3.
- $\mathsf{Enc}(ek_3, \mu)$: Choose $dk_2 \leftarrow [0, 2^{e_2})$ and $c_2 := \mathsf{isogen}_2(dk_2)$. Compute $j := \mathsf{isoex}_2(ek_3, dk_2)$. Compute $z := \mathsf{L}(j) \oplus \mu$. Output (c_2, z).
- $\mathsf{Dec}(dk_3, (c_2, z))$: Compute $j' := \mathsf{isoex}_3(c_2, dk_3)$ and output $\mu' := z \oplus \mathsf{L}(j')$.

SIKE uses $\mathsf{FO}^{\not\perp}$ for IND-CCA-secure KEM, where $\mathsf{G} = \mathsf{SHAKE256}_{e_2}$ and $\mathsf{H} = \mathsf{SHAKE256}_k$:

$\overline{\mathsf{Gen}}(1^\kappa)$	$\overline{\mathsf{Enc}}(ek)$	$\overline{\mathsf{Dec}}(\overline{dk}, (c_2, z))$, where $\overline{dk} = (dk, ek, s)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$	$\mu \leftarrow \{0,1\}^n$	$\mu' := \mathsf{Dec}(dk, (c_2, z))$
$s \leftarrow \{0,1\}^n$	$r := \mathsf{G}(\mu, ek)$	$r' := \mathsf{G}(\mu', ek)$
$\overline{dk} := (dk, ek, s)$	$(c_2, z) := \mathsf{Enc}(ek, \mu; r)$	$c'_2 := \mathsf{isogen}_2(r')$
return $(ek, \overline{dk})$	$K := \mathsf{H}(\mu, c_2, z)$	**if** $c_2 \neq c'_2$, **then return** $K := \mathsf{H}(s, c_2, z)$
	return $((c_2, z), K)$	**else return** $K := \mathsf{H}(\mu', c_2, z)$

Remark T.1. SIKE's $\overline{\mathsf{Dec}}$ performs the test $c_2 = c'_2$ but omits the test $z = z'$. Since Dec retrieves $\mu' := z \oplus k$ *deterministically*, we do not need to check the equality of z and z'.

T.1 Security

Assumptions:

[9] Correctly speaking, this algorithm outputs $(P'_m, Q'_m, R'_m := P'_m - Q'_m)$ and omits E'_m. We can reconstruct E'_m from P'_m, Q'_m, and R'_m.

Definition T.1 (Supersingular Computational Diffie-Hellman (SSCDH) Assumption [JD11], adapted). *Let $\phi_3 \colon E \to E_2'$ be an isogeny whose kernel is equal to $\langle P_3 + [dk_3]Q_3 \rangle$, where $dk_3 \leftarrow [0, 3^{e_3})$. Let $\phi_2 \colon E \to E_3'$ be an isogeny whose kernel is equal to $\langle P_2 + [dk_2]Q_2 \rangle$, where $dk_2 \leftarrow [0, 2^{e_2})$.*
For any QPT adversary, given the curves E_2' and E_3' and the points $\phi_3(P_2)$, $\phi_3(Q_2)$, $\phi_2(P_3)$, and $\phi_2(Q_3)$, finding the j-invariant of $E/\langle P_3 + [dk_3]Q_3, P_2 + [dk_2]Q_2 \rangle$ is hard.

Definition T.2 (Supersingular Decisional Diffie-Hellman (SSDDH) Assumption [JD11], adapted). *For any QPT adversary, given a tuple, it is hard to determine which distribution of the following two distributions generates the tuple:*

- *$(E_2', \phi_3(P_2), \phi_3(Q_2), E_3', \phi_2(P_3), \phi_2(Q_3), E_{23})$, where $E_2', \phi_3(P_2), \phi_3(Q_2), E_3', \phi_2(P_3), \phi_2(Q_3)$ are as in the SSCDH assumption and*
$$E_{23} \simeq E/\langle P_3 + [dk_3]Q_3, P_2 + [dk_2]Q_2 \rangle.$$
- *$(E_2', \phi_3(P_2), \phi_3(Q_2), E_3', \phi_2(P_3), \phi_2(Q_3), E_c)$, where $E_2', \phi_3(P_2), \phi_3(Q_2), E_3', \phi_2(P_3), \phi_2(Q_3)$ are as in the SSCDH assumption and*
$$E_c \simeq E/\langle P_3 + [dk_3']Q_3, P_2 + [dk_2']Q_2 \rangle,$$
where $dk_3' \leftarrow [0, 3^{e_3})$ and $dk_2' \leftarrow [0, 2^{e_2})$.

Security of SIKE-PKE: One can show the IND-CPA security of the underlying PKE of SIKE by assuming the SSDDH assumption and the entropy-smoothing property of L [10] as that in [JD11].

Lemma T.1. *Assume that the SSDDH assumption holds and L is entropy-smoothing. Then, SIKE-PKE* PKE *is* IND-CPA-*secure (and* OW-CPA-*secure).*

For ciphertext indistinguishablity, we construct a simulator $\mathcal{S}$ as follows: 1) sample $dk_2 \leftarrow [0, 2^{e_2})$ and compute $c_2 = (E_3', P_3', Q_3') := \mathsf{isogen}_2(dk_2)$; 2) sample $z \leftarrow \{0,1\}^n$; 3) output (c_2, z).

Lemma T.2. *SIKE-PKE* PKE *is ciphertext indistinguishable.*

Notice that we can remove the assumption on L's property.

Proof (Proof Sketch). We consider two games Game_0 and Game_1.

- Game_0: In this game the challenge ciphertext is computes as
$$\mu \leftarrow \{0,1\}^{256}; dk_2 \leftarrow [0, 2^{e_2}); c_2 := \mathsf{isogen}_2(dk_2); j \leftarrow \mathsf{isoex}_2(ek_3, dk_2); z := \mathsf{L}(j) \oplus \mu;\ \text{return } (c_2, z).$$
- Game_1: In this game the challenge ciphertext is computes as
$$dk_2 \leftarrow [0, 2^{e_2}); c_2 := \mathsf{isogen}_2(dk_2); z \leftarrow \{0,1\}^{256};\ \text{return } (c_2, z).$$

Game_0 and Game_1 are equivalent since μ in Game_0 and z in Game_1 are uniformly at random. □

Security of SIKE-DPKE: We next consider $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, which we call SIKE-DPKE.

Lemma T.3. *Assume that the SSDDH assumption holds and L is entropy-smoothing. Then, $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is disjointly-simulatable.*

Proof (Proof sketch:). Statistical disjointess follows from the fact that $|\mathcal{S}(1^\kappa)| \approx 2^{e_2} \cdot 2^n$ and $|\mathsf{Enc}'(ek, \mathcal{M})| \le 2^n$. Ciphertext indistinguishability follows from Theorem E.1 that states that T preserves SIKE-PKE's ciphertext indistinguishability (Lemma T.2) and its OW-CPA security (Lemma T.1). □

We next consider SIKE-DPKE's collision-freeness. If we consider XCFR-security, the adversary, given two encryption keys ek_3^0 and ek_3^1 with their decryption keys dk_3^0 and dk_3^1, should find μ such that $dk_2^0 = \mathsf{G}(\mu, ek_3^0)$, $dk_2^1 = \mathsf{G}(\mu, ek_3^1)$, and $z = \mu \oplus \mathsf{L}(j^0) = \mu \oplus \mathsf{L}(j^1)$, where $j^i \leftarrow \mathsf{isoex}_2(ek_3^i, dk_2^i)$. If $j^0 \neq j^1$, then it finds the collision for L, which should be hard (Lemma 2.3). For $j^0 = j^1$, it seems hard to find dk_2^0 and dk_2^1 such that $\mathsf{isoex}_2(ek_3^0, dk_2^0) = \mathsf{isoex}_2(ek_3^1, dk_2^1)$. Thus, we just assume the XCFR-security of SIKE-DPKE.

Summary: Recall that $\mathsf{FO}^{\not\perp}$ is $\mathsf{U}^{\not\perp} \circ \mathsf{T}$. Applying $\mathsf{U}^{\not\perp}$ to $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$, we obtain $\mathsf{KEM} = \mathsf{U}^{\not\perp}[\mathsf{PKE}', \mathsf{H}]$. After applying our theorems, we summarize the security properties of SIKE as follows:

- SIKE-DPKE PKE' is strongly disjointly-simulatable if the SSDDH assumption holds and L is entropy-smoothing.
- Thus, SIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM.
- SIKE is SCFR-CCA-secure if the underlying PKE $\mathsf{PKE}' = \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$ is SCFR-CCA-secure or XCFR-secure.
- SIKE is ANON-CCA-secure.
- SIKE leads to ANON-CCA-secure, SROB-CCA-secure hybrid PKE.

[10] We borrow the notation from [FNP14]. We say a family of hash functions $\mathfrak{H} = \{H \colon X \to Y\}$ is *entropy smoothing* [IZ89] if for any (Q)PPT adversary, it is hard to distinguish $(H, H(x))$ with (H, y), where $H \leftarrow \mathfrak{H}$, $x \leftarrow X$, and $y \leftarrow Y$.