NTRU leads to Anonymous, Robust Public-Key Encryption

Keita Xagawa[1]

NTT Secure Platform Laboratories, keita.xagawa.zv@hco.ntt.co.jp

Abstract. This short note shows that NTRU in NIST PQC Round 3 finalist is anonymous in the QROM if the underlying NTRU PKE is strongly disjoint-simulatable and a hybrid PKE scheme constructed from NTRU as KEM and appropriate DEM is anonymous and robust.
This solves the the open problem to investigate anonymity and robustness of NTRU posed by Grubbs, Maram, and Paterson (Cryptography ePrint Archive 2021/708).

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

1 Introduction

Roughly speaking, PKE is *anonymous* [BBDP01] if a ciphertext hides the receiver's information. Intuitively speaking, PKE is *robust* [ABN10] if only the intended receiver can obtain a meaningful plaintext from a ciphertext. Grubbs, Maram, and Paterson [GMP21] discussed anonymity and robustness of post-quantum KEM schemes in NIST PQC Standardization finalists, which is an extended version of Mohassel [Moh10]. Grubbs et al. left several open problems. One of them is the case of NTRU and they wrote in [GMP21, Section 6]:

> Important questions remain about the anonymity and robustness of the NIST finalists and alternate candidates. For example, the status of NTRU is open, and it is plausible that the anonymity of CM could be proven by a direct approach.

Our contribution: In this short note, we solve this open problem: We show that NTRU is anonymous in the QROM starting from NTRU's pseudorandomness and the hybrid PKE using NTRU is strongly robust by showing NTRU is strongly collision free in the QROM.

2 Definitions

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

Quantum Random Oracle Model:

Lemma 2.1. *Let ℓ be a positive integer. Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $\mathsf{H}_0 \colon \{0,1\}^\ell \times \mathcal{X} \to \mathcal{Y}$ 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*

$$\left|\Pr[s \leftarrow \{0,1\}^\ell : \mathcal{A}^{\mathsf{H}_0, \mathsf{H}_0(s,\cdot)}() \to 1] - \Pr[\mathcal{A}^{\mathsf{H}_0, \mathsf{H}_q}() \to 1]\right| \leq Q \cdot 2^{-(\ell-1)/2},$$

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

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

Lemma 2.2 (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*

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

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

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

Lemma 2.3 (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

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

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

The following proof is due to Hosoyamada [Hos20].

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 $\mathsf{H}_0\colon \mathcal{X}_1 \to \mathcal{Y}$ 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 2.3, we obtain the bound as we wanted. □

The best upper bound for the claw-finding problem is given by Tani [Tan09]. His algorithm runs in $O\big((N_0N_1)^{1/3}\big)$ if $N_0 \leq N_1 < N_0^2$ and $O\big(N_1^{1/2}\big)$ if $N_1 \geq N_0^2$, which match the lower bound by Buhrman et al. [BDH+05] and Zhang [Zha05]. While there may be a gap, the above upper bound of the success probability is enough for cryptography.

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* (Gen, Enc, 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.*
- $\mathsf{Enc}(ek, m; r_e) \to c$*: an encryption algorithm that takes as input encryption key ek, message $m \in \mathcal{M}$, and randomness $r_e \in \mathcal{R}_{\mathsf{Enc}}$, and outputs ciphertext $c \in C$.*
- $\mathsf{Dec}(dk, c) \to m/\bot$*: a decryption algorithm that takes as input decryption key dk and ciphertext c and outputs message $m \in \mathcal{M}$ 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-correctness *if*

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

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 $m \in \mathcal{M}$,*

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

Remark 2.2. Xagawa and Yamakawa [XY19] observed that if PKE is deterministic, then δ-correctness implies that

$$\mathrm{Exp}_{(ek,dk)\leftarrow\mathsf{Gen}(1^\kappa)}[(ek, dk) \text{ is inaccurate}] \leq \delta.$$

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 PRCCA 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].

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-cpa, ind-cca, pr-cca, anon-cca}*, we define its* goal-atk *advantage against* PKE *as follows:*

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

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

For any $\mathcal{A}$ *and* goal-atk $\in$ {ow-cpa, srob-cca, scfr-cca, wrob-cca, wcfr-cca}*, 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-CPA, IND-CPA, IND-CCA, PR-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA}*, we say that* PKE *is* GOAL-ATK*-secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{PKE}[,\mathcal{D}_{\mathcal{M}}],\mathcal{A}}(\kappa)$ *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.*

Disjoint simulatability:

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* $\mathcal{A}$*,*

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

Liu and Wang gave a slightly modified version of DS in [LW21]. As they noted, this definition 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 2.3. We note that a deterministic PKE scheme produced by TPunc [SXY18] and Punc [HKSU20] is not *special*, because their simulator will output a random ciphertext with special plaintext, $\mathsf{Enc}(ek, \hat{m})$.

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* $\delta = 0$.

```
$\mathsf{Expt}^{\text{ow-cpa}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}},\mathcal{A}}(\kappa)$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$m^* \leftarrow \mathcal{D}_{\mathcal{M}}$
$c^* \leftarrow \mathsf{Enc}(ek, m^*)$
$m' \leftarrow \mathcal{A}(ek, c^*)$
return $\mathsf{boole}(m' \overset{?}{=} \mathsf{Dec}(dk, c^*))$
```

```
$\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$
$b \leftarrow \{0, 1\}$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1(ek)$
$c^* \leftarrow \mathsf{Enc}(ek, m_b)$
$b' \leftarrow \mathcal{A}_2(c^*, state)$
return $\mathsf{boole}(b = b')$
```

```
$\textsc{Dec}_a(c)$
if $c = a$, return $\perp$
$m := \mathsf{Dec}(dk, c)$
return $m$
```

```
$\textsc{Dec}_a(\mathsf{id}, c)$
if $c = a$, return $\perp$
$m := \mathsf{Dec}(dk_{id}, c)$
return $m$
```

```
$\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$
$b \leftarrow \{0, 1\}$
$(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
$(m_0, m_1, state) \leftarrow \mathcal{A}_1^{\textsc{Dec}_\perp(\cdot)}(ek)$
$c^* \leftarrow \mathsf{Enc}(ek, m_b)$
$b' \leftarrow \mathcal{A}_2^{\textsc{Dec}_{c^*}(\cdot)}(c^*, state)$
return $\mathsf{boole}(b = b')$
```

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

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

```
$\mathsf{Expt}^{\text{srob-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$
$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$m_0 \leftarrow \mathsf{Dec}(dk_0, c)$
$m_1 \leftarrow \mathsf{Dec}(dk_1, c)$
return $\mathsf{boole}(m_0 \neq \perp \wedge m_1 \neq \perp)$
```

```
$\mathsf{Expt}^{\text{scfr-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa)$
$(ek_0, dk_0) \leftarrow \mathsf{Gen}(1^\kappa)$
$(ek_1, dk_1) \leftarrow \mathsf{Gen}(1^\kappa)$
$c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
$m_0 \leftarrow \mathsf{Dec}(dk_0, c)$
$m_1 \leftarrow \mathsf{Dec}(dk_1, c)$
return $\mathsf{boole}(m_0 = m_1 \neq \perp)$
```

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

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

Fig. 1. Games for PKE schemes

Security: 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 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) [GMP21].

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-cpa, ind-cca, pr-cca, pr2-cca, anon-cca, srob-cca, scfr-cca}*, we define its* goal-atk *advantage against* KEM *as follows:*

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

where $\mathsf{Expt}^{\text{goal-atk}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ *is an experiment described in Figure 1.*
For any $\mathcal{A}$ *and* goal-atk $\in$ {srob-cca, scfr-ccawrob-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-CPA, IND-CCA, PR-CCA, PR2-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA}*, we say that* KEM *is* GOAL-ATK*-secure if* $\mathsf{Adv}^{\text{goal-atk}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ *is negligible for any QPT adversary* $\mathcal{A}$*. We also say that* KEM *is* SPR-CCA*-secure (or* SPR2-CCA*-secure) if it is* PR-CCA*-secure (or* PR2-CCA*-secure) and its simulator ignores ek, respectively.*

2.3 Data Encapsulation

The model for DEM schemes is summarized as follows:

Definition 2.10. *A DEM scheme* DEM *consists of the following triple of polynomial-time algorithms* (E, D) *with key space* $\mathcal{K}$ *and message space* $\mathcal{M}$*:*

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

Definition 2.11 (Correctness). *We say* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *has* perfect correctness *if for any* $K \in \mathcal{K}$ *and any* $m \in \mathcal{M}$*, we have*

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

Robustness of DEM (FROB and XROB) are taken from Farshim, Orlandi, and Roşi [FOR17].

Definition 2.12 (Security notions for DEM). *Let* $\mathsf{DEM} = (\mathsf{E}, \mathsf{D})$ *be a DEM scheme whose key space is* $\mathcal{K}$*.*
For any $\mathcal{A}$ *and* goal-atk $\in$ {ind-cca, pr-cca, pr-otcca}*, we define its* goal-atk *advantage against* DEM *as follows:*

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

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* $\mathcal{A}$*.*

```
$\mathsf{Expt}^{\text{ind-cpa}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $b \leftarrow \{0,1\}$
    $(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(c^*, K_0^*) \leftarrow \overline{\mathsf{Enc}}(ek);$
    $K_1^* \leftarrow \mathcal{K}$
    $b' \leftarrow \mathcal{A}(ek, c^*, K_b^*)$
    return boole$(b = b')$
```

```
$\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $b \leftarrow \{0,1\}$
    $(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(c^*, K_0^*) \leftarrow \overline{\mathsf{Enc}}(ek);$
    $K_1^* \leftarrow \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(ek, c^*, K_b^*)$
    return boole$(b = b')$
```

```
$\textsc{Dec}_a(c)$
    if $c = a$, return $\perp$
    $K := \overline{\mathsf{Dec}}(dk, c)$
    return $K$
```

```
$\textsc{Dec}_a(\mathsf{id}, c)$
    if $c = a$, return $\perp$
    $K := \overline{\mathsf{Dec}}(dk_{\mathsf{id}}, c)$
    return $K$
```

```
$\mathsf{Expt}^{\text{pr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $b \leftarrow \{0,1\}$
    $(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(c_0^*, K_0^*) \leftarrow \overline{\mathsf{Enc}}(ek);$
    $(c_1^*, K_1^*) \leftarrow \mathcal{S}(1^\kappa, ek) \times \mathcal{K}$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}(\cdot)}(ek, c_b^*, K_b^*)$
    return boole$(b = b')$
```

```
$\mathsf{Expt}^{\text{pr2-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $b \leftarrow \{0,1\}$
    $(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(c^*, K_0^*) \leftarrow \mathcal{S}(1^\kappa, ek) \times \mathcal{K}$
    $K_1^* \leftarrow \overline{\mathsf{Dec}}(dk, c^*)$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot)}(ek, c^*, K_b^*)$
    return boole$(b = b')$
```

```
$\mathsf{Expt}^{\text{anon-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $b \leftarrow \{0,1\}$
    $(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(c^*, K^*) \leftarrow \overline{\mathsf{Enc}}(ek);$
    $b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c^*}(\cdot,\cdot)}(ek, c^*, K^*)$
    return boole$(b = b')$
```

```
$\mathsf{Expt}^{\text{srob-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
    $K_0 \leftarrow \overline{\mathsf{Dec}}(dk_0, c)$
    $K_1 \leftarrow \overline{\mathsf{Dec}}(dk_1, c)$
    return boole$(K_0 \neq \perp \wedge K_1 \neq \perp)$
```

```
$\mathsf{Expt}^{\text{scfr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $c \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
    $K_0 \leftarrow \overline{\mathsf{Dec}}(dk_0, c)$
    $K_1 \leftarrow \overline{\mathsf{Dec}}(dk_1, c)$
    return boole$(K_0 = K_1 \neq \perp)$
```

```
$\mathsf{Expt}^{\text{wrob-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $b \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
    $(c, K_b) \leftarrow \overline{\mathsf{Dec}}(ek_b)$
    $K_{1-b} \leftarrow \overline{\mathsf{Dec}}(dk_{1-b}, c)$
    return boole$(K_{1-b} \neq \perp)$
```

```
$\mathsf{Expt}^{\text{wcfr-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$
    $(ek_0, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $(ek_1, dk_1) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
    $b \leftarrow \mathcal{A}^{\textsc{Dec}_\perp(\cdot,\cdot)}(ek_0, ek_1)$
    $(c, K_b) \leftarrow \overline{\mathsf{Dec}}(ek_b)$
    $K_{1-b} \leftarrow \overline{\mathsf{Dec}}(dk_{1-b}, c)$
    return boole$(K_0 = K_1 \neq \perp)$
```

Fig. 2. Games for KEM schemes

```
$\mathsf{Expt}^{\text{ind-cca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$
$b \leftarrow \{0,1\}$
$K \leftarrow \mathcal{K}$
$(m_0, m_1, state) \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{\perp}(\cdot)}(1^\kappa)$
$c^* \leftarrow \mathsf{E}(K, m_b)$
$b' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{c^*}(\cdot)}(c^*, state)$
return $\mathsf{boole}(b = b')$
```

```
$\textsc{Enc}(m)$
$c \leftarrow \mathsf{E}(k, m)$
return $c$
```

```
$\textsc{Dec}_a(c)$
if $c = a$, return $\perp$
$m \leftarrow \mathsf{D}(k, c)$
return $m$
```

```
$\mathsf{Expt}^{\text{pr-cca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$
$b \leftarrow \{0,1\}$
$K \leftarrow \mathcal{K}$
$(m, state) \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{\perp}(\cdot)}(1^\kappa)$
$c_0^* \leftarrow \mathsf{E}(K, m)$
$c_1^* \leftarrow C_{|m|}$
$b' \leftarrow \mathcal{A}^{\textsc{Enc}(\cdot),\textsc{Dec}_{c_b^*}(\cdot)}(c_b^*, state)$
return $\mathsf{boole}(b = b')$
```

```
$\mathsf{Expt}^{\text{pr-otcca}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$
$b \leftarrow \{0,1\}$
$K \leftarrow \mathcal{K}$
$(m, state) \leftarrow \mathcal{A}(1^\kappa)$
$c_0^* \leftarrow \mathsf{E}(K, m)$
$c_1^* \leftarrow C_{|m|}$
$b' \leftarrow \mathcal{A}^{\textsc{Dec}_{c_b^*}(\cdot)}(c_b^*, state)$
return $\mathsf{boole}(b = b')$
```

```
$\mathsf{Expt}^{\text{frob}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$
$(c, k_0, k_1) \leftarrow \mathcal{A}(1^\kappa)$
$m_0 \leftarrow \mathsf{D}(k_0, c)$
$m_1 \leftarrow \mathsf{D}(k_1, c)$
$b \leftarrow \mathsf{boole}(m_0 \neq \perp \wedge m_1 \neq \perp)$
$b_k \leftarrow \mathsf{boole}(k_0 \neq k_1)$
return $\mathsf{boole}(b \wedge b_k)$
```

```
$\mathsf{Expt}^{\text{xrob}}_{\mathsf{DEM},\mathcal{A}}(\kappa)$
$(m_0, k_0, R_0, k_1, c_1) \leftarrow \mathcal{A}(1^\kappa)$
$c_0 \leftarrow \mathsf{E}(k_0, m_0; R_0)$
$m_1 \leftarrow \mathsf{D}(k_1, c_1)$
$b \leftarrow \mathsf{boole}(m_0 \neq \perp \wedge m_1 \neq \perp)$
$b_k \leftarrow \mathsf{boole}(k_0 \neq k_1)$
$b_c \leftarrow \mathsf{boole}(c_0 = c_1 \neq \perp)$
return $\mathsf{boole}(b \wedge b_k \wedge b_c)$
```

Fig. 3. Games for DEM schemes

2.4 Review of Grubbs et al. [GMP21]

Grubbs et al. studied KEM's anonymity and hybrid PKE's anonymity and robustness, which is an extension of Mohassel [Moh10]. The main difference of Grubbs et al. [GMP21] form Mohassel [Moh10] is they treat KEM with implicit rejection, which are used in all NIST PQC Round 3 KEM candidates except HQC.
Roughly speaking, they showed that

Theorem 2.1 ([GMP21, Theorem 2]). *If* KEM *is* SCFR-CCA*-secure and* WCFR-CCA*-secure and* DEM *is* FROB*-secure and* XROB*-secure, then a hybrid PKE scheme* PKE *obtained by composing* KEM *and* DEM *is* SROB-CCA*-secure and* WROB-CCA*-secure, respectively.*

They also showed that

Theorem 2.2 ([GMP21, Theorem 7]). *If* KEM *is obtained by* $\mathsf{FO}^{\not\perp}$ *with* PKE_1, KEM *is* ANON-CCA*-secure and* IND-CCA*-secure,* PKE_1 *is* WCFR-CPA*-secure, δ-correct, and γ-spreading,* DEM *is* INT-CTXT*-secure, then a hybrid PKE scheme* PKE *obtained by composing* KEM *and* DEM *is* ANON-CCA*-secure.*

3 Strong Pseudorandomness implies Anonymity

We observe that strong pseudorandomness immediately implies anonymity, which may be a folklore. For completeness, we include the proof for PKE.

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.*

Proof: 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 $\mathcal{S}(1^\kappa) \times C_{\mathsf{DEM},|m|}$.

Table 1. Summary of Games for the Proof of Theorem 4.1

Game	c^* and K^*	d^*	Decryption oracle	justification
Game_0	$\overline{\mathsf{Enc}}(ek)$	$\mathsf{E}(K^*, m^*)$	reject if $(c, d) = (c^*, d^*)$	
Game_1	$\overline{\mathsf{Enc}}(ek)$ at first	$\mathsf{E}(K^*, m^*)$	reject if $(c, d) = (c^*, d^*)$	conceptual change
Game_2	$\overline{\mathsf{Enc}}(ek)$ at first	$\mathsf{E}(K^*, m^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	δ-correctness
Game_3	$\mathcal{S}(1^\kappa) \times \mathcal{K}$ at first	$\mathsf{E}(K^*, m^*)$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-CCA security of KEM
Game_4	$\mathcal{S}(1^\kappa) \times \mathcal{K}$ at first	$U(C_{\mathsf{DEM},\lvert m^*\rvert})$	reject if $(c, d) = (c^*, d^*)$; use K^* if $c = c^*$	SPR-oTCCA security of DEM
Game_5	$\mathcal{S}(1^\kappa) \times \mathcal{K}$ at first	$U(C_{\mathsf{DEM},\lvert m^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	SPR2-CCA security of KEM
Game_6	$\mathcal{S}(1^\kappa) \times \mathcal{K}$	$U(C_{\mathsf{DEM},\lvert m^*\rvert})$	reject if $(c, d) = (c^*, d^*)$	conceptual change

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

$$\frac{1}{2}\left|\Pr[S_{0,b}] - \Pr[S_{1,b}]\right| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{A}_{1b}}(\kappa) \text{ and } \Pr[S_{1,0}] = \Pr[S_{1,1}].$$

Hence, we have

$$\mathsf{Adv}^{\text{anon-cca}}_{\mathsf{PKE},\mathcal{A}}(\kappa) = \frac{1}{2}\left|\Pr[S_{0,0}] - \Pr[S_{0,1}]\right| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{A}_{10}}(\kappa) + \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{PKE},\mathcal{A}_{11}}(\kappa).$$

This completes the proof. □

4 Strong Pseudorandomness of Hybrid PKE

The hybrid PKE PKE = (Gen, Enc, Dec) constructed from KEM = $(\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ and DEM = (E, D) is summarized as follows:

$\mathsf{Gen}(1^\kappa)$	$\mathsf{Enc}(ek, m)$	$\mathsf{Dec}(dk, (c, d))$

$(ek, dk) \leftarrow \overline{\mathsf{Gen}}(1^\kappa)$
return (ek, dk)
	$(c, K) \leftarrow \overline{\mathsf{Enc}}(ek)$
	$d \leftarrow \mathsf{E}(K, m)$
	return (c, d)
		$K' \leftarrow \overline{\mathsf{Dec}}(dk, c)$
		if $K' = \bot$ **then return** $\bot$
		$m' \leftarrow \mathsf{D}(K', d)$
		if $m' = \bot$ **then return** $\bot$
		return m'

Theorem 4.1. *Let* PKE = (Gen, Enc, Dec) *be a hybrid encryption scheme obtained by composing a KEM scheme* KEM = $(\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ *and a DEM scheme* DEM = (E, D) *that share key space* $\mathcal{K}$. *If* KEM *is* SPR-CCA*-secure,* SPR2-CCA*-secure, and* δ*-correct with negligible* δ *and* DEM *is* PR-oTCCA*-secure, then* PKE *is* SPR-CCA*-secure.*

The security proof is similar to the security proof of IND-CCA-security of KEM/DEM [CS03]. However, we need to take care about pseudorandom ciphertexts.

Proof: In the following, we consider Game_i for $i = 0, \ldots, 6$. We summarize the games in Table 1. Let S_i denote the event that the adversary outputs $b' = 1$ in Game_i.

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

$$(c^*_0, K^*_0) \leftarrow \overline{\mathsf{Enc}}(ek); d^*_0 \leftarrow \mathsf{E}(K^*_0, m^*); \text{ return } ct^* = (c^*_0, d^*_0).$$

We have

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

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

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

Game_2*:* In this game, the decryption oracle uses K^* is $c = c^*$ instead of $K = \mathsf{Dec}(sk, c^*)$. Game$_1$ and Game$_2$ differ if correctly generated ciphertext c^* with K^* is decapsulated into different $K \neq K^*$ or $\perp$, which 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:

$$(c_1^*, K_1^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}; d^+ \leftarrow \mathsf{E}(K_1^*, m^*); \text{ return } ct^* = (c_1^*, d^+).$$

Apparently, the difference is bounded by SPR-CCA security of KEM: There is an adversary $\mathcal{A}_{23}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$\frac{1}{2}|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}^{\text{spr-cca}}_{\mathsf{KEM},\mathcal{A}_{23}}(\kappa).$$

(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:

$$(c_1^*, K_1^*) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}; d_1^* \leftarrow \mathcal{C}_{\mathsf{DEM},|m|}; \text{ return } ct^* = (c_1^*, d_1^*).$$

Apparently, 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

$$\frac{1}{2}|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\text{spr-otcca}}_{\mathsf{DEM},\mathcal{A}_{34}}(\kappa).$$

(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 $K = \mathsf{Dec}(sk, c^*)$ instead of K^*.
The difference is bounded by SPR2-CCA security of KEM: There is an adversary $\mathcal{A}_{45}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying

$$\frac{1}{2}|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{Adv}^{\text{spr2-cca}}_{\mathsf{DEM},\mathcal{A}_{45}}(\kappa).$$

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

Game_6*:* We change the timing of the generation of (c_1^*, K_1^*). This is just conceptual change and we have

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

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

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

Summarizing the (in)equalities, we obtain the bound in the statement. □

Table 2. Summary of Games for the Proof of Theorem 5.1

				Decryption		
Game	H	c^*	K^*	valid c	invalid c	justification
Game_0	$\mathsf{H}(\cdot)$	$\mathsf{Enc}(ek, m^*)$	$\mathsf{H}(m^*)$	$\mathsf{H}(m)$	$\mathsf{H}_0(s, c)$	
Game_1	$\mathsf{H}(\cdot)$	$\mathsf{Enc}(ek, m^*)$	$\mathsf{H}(m^*)$	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	Lemma 2.1
$\mathsf{Game}_{1.5}$	$\mathsf{H}'_q(\mathsf{Enc}(ek, \cdot))$	$\mathsf{Enc}(ek, m^*)$	$\mathsf{H}(m^*)$	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_2	$\mathsf{H}_q(\mathsf{Enc}(ek, \cdot))$	$\mathsf{Enc}(ek, m^*)$	$\mathsf{H}(m^*)$	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_3	$\mathsf{H}_q(\mathsf{Enc}(ek, \cdot))$	$\mathsf{Enc}(ek, m^*)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	if key is accurate
Game_4	$\mathsf{H}_q(\mathsf{Enc}(ek, \cdot))$	$\mathcal{S}(1^\kappa)$	$\mathsf{H}_q(c^*)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	DS-IND
Game_5	$\mathsf{H}_q(\mathsf{Enc}(ek, \cdot))$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c)$	statistical disjointness
Game_6	$\mathsf{H}_q(\mathsf{Enc}(ek, \cdot))$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
$\mathsf{Game}_{6.5}$	$\mathsf{H}'_q(\mathsf{Enc}(ek, \cdot))$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_7	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	if key is accurate
Game_8	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(m)$	$\mathsf{H}_0(s, c)$	Lemma 2.1

5 SXY may be Strongly Pseudorandom in the QROM

Let us review SXY [SXY18] as known as $\mathsf{U}_m^{\not\perp}$ [HHK17]. (We note that SXY requires the re-encryption check but $\mathsf{U}_m^{\perp}$ does not.)
Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme whose plaintext space is $\mathcal{M}$. Let $\mathcal{C}$ and $\mathcal{K}$ be a ciphertext and key space. Let $\mathsf{H} \colon \mathcal{M} \to \mathcal{K}$ and $\mathsf{H}' \colon \{0,1\}^\ell \times \mathcal{C} \to \mathcal{K}$ be hash functions modeled by random oracles. $\mathsf{KEM} = (\overline{\mathsf{Gen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}}) = \mathsf{SXY}[\mathsf{PKE}, \mathsf{H}, \mathsf{H}_0]$ is defined as follows:

```
$\overline{\mathsf{Gen}}(1^\kappa)$
    $(ek, dk) \leftarrow \mathsf{Gen}(1^\kappa)$
    $s \leftarrow \{0,1\}^\ell$
    $\overline{dk} := (dk, ek, s)$
    return $(ek, \overline{dk})$

$\overline{\mathsf{Enc}}(ek)$
    $m \leftarrow \mathcal{M}$
    $c := \mathsf{Enc}(ek, m)$
    $K := \mathsf{H}(m)$
    return $(c, K)$

$\overline{\mathsf{Dec}}(\overline{dk}, c)$, where $\overline{dk} = (dk, ek, s)$
    $m' \leftarrow \mathsf{Dec}(dk, c)$
    if $m' = \perp$ then return $K := \mathsf{H}_0(s, c)$
    if $c \neq \mathsf{Enc}(pk, m')$return $K := \mathsf{H}_0(s, c)$
    else return $K := \mathsf{H}(m')$
```

SPR-CCA security:

Theorem 5.1. *Suppose that a ciphertext space $\mathcal{C}$ of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* PKE *is SPR-CCA-secure.*

Proof Sketch: We use the game-hopping proof. We consider Game_i for $i = 0, \ldots, 8$. We summarize the games in Table 2. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i. Let Acc and $\overline{\text{Acc}}$ denote the event that the key pair (ek, dk) is accurate and inaccurate, respectively.
We mainly follow the security proof in [XY19].

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 $\mathsf{H}_0(s, c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q \colon \mathcal{C} \to \mathcal{K}$ 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 2.1 we have the bound

$$|\Pr[S_0] - \Pr[S_1]| \leq q_{\mathsf{H}_0} \cdot 2^{-(\ell-1)/2},$$

where q_{H_0} denote the number of queries to H_0 the adversary makes.

$\mathsf{Game}_{1.5}$*:* This game is the same as Game_1 except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}'_q(\mathsf{Enc}(ek,\cdot))$ where $\mathsf{H}'_q\colon C \to \mathcal{K}$ 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 a key pair is accurate, the two games Game_1 and $\mathsf{Game}_{1.5}$ are equivalent because $\mathsf{Enc}(ek,\cdot)$ is *injective*. See [XY19, Lemma 4.3] for the detail.

Game_2*:* This game is the same as Game_1 except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}_q(\mathsf{Enc}(ek,\cdot))$ instead of $\mathsf{H}_q(\mathsf{Enc}(ek,\cdot))$.
If a key pair is accurate, the two games $\mathsf{Game}_{1.5}$ and Game_2 are equivalent as in the proof of [XY19, Lemma 4.4].

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 a key pair is accurate, the two games Game_2 and Game_3 are equivalent as in the proof of [XY19, Lemma 4.5].

Game_4*:* This game is the same as Game_3 except that c^* is generated by $\mathcal{S}(1^\kappa)$.
Apparently, 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].

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].

Game_6*:* This game is the same as Game_5 except that the decapsulation oracle is reset as $\textsc{Dec}$.
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].

$\mathsf{Game}_{6.5}$*:* This game is the same as Game_6 except that the random oracle $\mathsf{H}(\cdot)$ is simulated by $\mathsf{H}'_q(\mathsf{Enc}(ek,\cdot))$ where $\mathsf{H}'_q\colon C \to \mathcal{K}$ is yet another random oracle as in $\mathsf{Game}_{1.5}$.
If a key pair is accurate, the two games Game_6 and $\mathsf{Game}_{6.5}$ are equivalent as in the proof of [XY19, Lemma 4.4].

Game_7*:* This game is the same as $\mathsf{Game}_{6.5}$ except that the random oracle H is chosen from $\{\mathsf{H}\colon \mathcal{M} \to \mathcal{K}\}$.
If a key pair is accurate, the two games $\mathsf{Game}_{6.5}$ and Game_7 are equivalent because $\mathsf{Enc}(ek,\cdot)$ is *injective*. See [XY19, Lemma 4.3] for the detail.

Game_8*:* This game is the same as Game_7 except that $\mathsf{H}_q(c)$ in the decapsulation is replaced by $\mathsf{H}_0(s,c)$.
As in [XY19, Lemmas 4.1], from Lemma 2.1 we have the bound

$$|\Pr[S_7] - \Pr[S_8]| \leq q_{\mathsf{H}_0} \cdot 2^{-(\ell-1)/2}.$$

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

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

SPR2-CCA security:

Theorem 5.2. *Suppose that a ciphertext space C of* PKE *depends on the public parameter only. If* PKE *is strongly disjoint-simulatable, then* PKE *is* SPR2-CCA*-secure.*

Proof Sketch: We use the game-hopping proof. We consider Game_i for $i = 0, \ldots, 6$. We summarize the games in Table 3. Let S_i denote the event that the adversary outputs $b' = 1$ in game Game_i. Let Acc and $\overline{\text{Acc}}$ denote the event that the key pair (ek, dk) is accurate and inaccurate, respectively.

Game_0*:* This game is the original game $\mathsf{Expt}^{\text{spr2-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as

$$(c^*, K^*_0) \leftarrow \mathcal{S}(1^\kappa) \times \mathcal{K}.$$

We have

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

Table 3. Summary of Games for the Proof of Theorem 5.1: '$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$' 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	 invalid c	justification
Game_0	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa)$	random	$\mathsf{H}(m)$	$\mathsf{H}_0(s,c)$	
Game_1	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(m)$	$\mathsf{H}_0(s,c)$	statistical disjointness
Game_2	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	random	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	Lemma 2.1
Game_3	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}_q(c^*)$	$\mathsf{H}(m)$	$\mathsf{H}_q(c)$	$\mathsf{H}_q(c^*)$ is hidden
Game_4	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{H}_0(s,c^*)$	$\mathsf{H}(m)$	$\mathsf{H}_0(s,c)$	Lemma 2.1
Game_5	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa) \setminus \mathsf{Enc}(ek, \mathcal{M})$	$\mathsf{Dec}(dk,c^*)$	$\mathsf{H}(m)$	$\mathsf{H}_0(s,c)$	re-encryption check and key's accuracy
Game_6	$\mathsf{H}(\cdot)$	$\mathcal{S}(1^\kappa)$	$\mathsf{Dec}(dk,c^*)$	$\mathsf{H}(m)$	$\mathsf{H}_0(s,c)$	statistical disjointness

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.

Game_2*:* This game is the same as Game_1 except that $\mathsf{H}_0(s,c)$ in the decapsulation oracle is replace with $\mathsf{H}_q(c)$ where $\mathsf{H}_q : \mathcal{C} \to \mathcal{K}$ is another random oracle.
As in [XY19, Lemmas 4.1], from Lemma 2.1 we have the bound

$$|\Pr[S_1] - \Pr[S_2]| \leq q_{\mathsf{H}_0} \cdot 2^{-(\ell-1)/2},$$

where q_{H_0} denote the number of queries to H_0 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.

Game_4*:* This game is the same as Game_3 except that $\mathsf{H}_q(\cdot)$ is replaced by $\mathsf{H}_0(s,\cdot)$. As in [XY19, Lemmas 4.1], from Lemma 2.1 we have the bound

$$|\Pr[S_3] - \Pr[S_4]| \leq q_{\mathsf{H}_0} \cdot 2^{-(\ell-1)/2},$$

where q_{H_0} denote the number of queries to H_0 the adversary makes.

Game_5*:* This game is the same as Game_4 except that $K^* := \mathsf{Dec}(dk, c^*)$ instead of $\mathsf{H}_0(s, c^*)$. Recall that c^* is always outside of $\mathsf{Enc}(ek, \mathcal{M})$. If a key pair is accurate, then $\mathsf{Enc}(ek, \mathsf{Dec}(c^*)) \neq c^*$ and $K^* = \mathsf{H}_0(s, c^*)$. Hence, the two games are equivalent if a key pair is accurate.

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.
Moreover, this game Game_6 is the original game $\mathsf{Expt}^{\text{spr2-cca}}_{\mathsf{KEM},\mathcal{A}}(\kappa)$ with $b = 1$.
Summarizing the (in)equalities, we obtain Theorem 5.2

6 Review of NTRU

Let us briefly review NTRU [CDH+20]. 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 say a polynomial *ternary* if its coefficients are in $\{-1, 0, +1\}$.
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.. $\underline{\mathrm{S3}}(a)$ 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, $\mathcal{T} = \{a = \sum_{i=0}^{n-2} a_i x^i \mid a \neq 0 \wedge a_i \in \{-1, 0, +1\}\}$. 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, $\{a \in \mathcal{T} \mid \#\{a_i : a_i = 1\} = \#\{a_i : a_i = -1\} = d/2\}$.
The following lemma is due to Schanck [Sch20]. (See e.g. for [CDH+20, p.22] 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 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. □

```
Gen(1^κ)
    $(f, g) \leftarrow \mathsf{Sample\_fg}()$
    $f_q \leftarrow (1/f) \bmod (q, \Phi_n)$
    $h \leftarrow (3 \cdot g \cdot f_q) \bmod (q, \Phi_1\Phi_n)$
    $h_q \leftarrow (1/h) \bmod (q, \Phi_n)$
    $f_p \leftarrow (1/f) \bmod (3, \Phi_n)$
    $ek := h, dk := (f, f_p, h_q)$
    return $(ek, dk)$
```

```
Enc(h, (r, m))
    $m' \leftarrow \mathsf{Lift}(m)$
    $c \leftarrow (h \cdot r + m') \bmod (q, \Phi_1\Phi_n)$
    return $c$
```

```
Dec((f, f_p, h_q), c)
    if $c \not\equiv 0 \bmod (q, \Phi_1)$ then return $(0, 0, 1)$
    $a \leftarrow (c \cdot f) \bmod (q, \Phi_1\Phi_n)$
    $m \leftarrow (a \cdot f_p) \bmod (3, \Phi_n)$
    $m' \leftarrow \mathsf{Lift}(m)$
    $r \leftarrow ((c - m') \cdot h_q) \bmod (q, \Phi_n)$
    if $(r, m) \in \mathcal{L}_r \times \mathcal{L}_m$ then return $(r, m, 0)$
    else return $(0, 0, 1)$
```

Fig. 4. The DPKE for NTRU

NTRU-HPS:

$$\mathcal{L}_f = \mathcal{T}, \mathcal{L}_g = \mathcal{T}(q/8 - 2), \mathcal{L}_r = \mathcal{T}, \mathcal{L}_m = \mathcal{T}(q/8 - 2),$$

and $\mathsf{Lift}(m) = m$. 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-HRSS-KEM: The adversary who is given an honestly generated h cannot distinguish an honestly-generated ciphertext from an element of $\{v \in R/q \mid v \equiv 0 \bmod (q, \Phi_1)\}$ drawn uniformly at random.

$$\mathcal{L}_f = \mathcal{T}_+, \mathcal{L}_g = \{\Phi_1 \cdot v \mid v \in \mathcal{T}_+\}, \mathcal{L}_r = \mathcal{T}, \mathcal{L}_m = \mathcal{T},$$

and $\mathsf{Lift}(m) = \Phi_1 \cdot \underline{S3}(m/\Phi_1)$. 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].)

Rigidity: Notice that we implicitly check $hr + \mathsf{Lift}(m) = c$ by checking if $(r, m) \in \mathcal{L}_r \times \mathcal{L}_r$. See [CDH+20] for the details.

7 NTRU is SPR-CCA and SPR2-CCA in the QROM

We have known that the NTRU PKE is disjointly simulatable ([SXY18]) if the decisional small polynomial ratio (DSPR) assumption [LTV12] and the polynomial learning with errors (PLWE) assumption [] hold. See [SXY18, Section 3.3 of the ePrint version.]. Adapting their argument to NTRU in Round 3, the simulator $\mathcal{S}$ will output a random polynomial $c \leftarrow R/q$ such that $c \equiv 0 \pmod{(q, \Phi_1)}$.
Combining this property with previous theorems, we conclude that NTRU-HPS and NTRU-HRSS are SPR-CCA-secure and SPR2-CCA-secure using appropriate assumptions.

8 NTRU is Strongly Collision-Free

In order to show strong robustness of the hybrid PKE, we use Theorem 2.1 ([GMP21, Theorem 2]). We show NTRU's SCFR-CCA-security by using the collision-resistant property of H_0 and H and the claw-free property of H_0 and H.

Theorem 8.1 (SCFR-CCA-**security of NTRU**). *NTRU is*SCFR-CCA-*secure in the QROM.*

Proof: Suppose that an adversary outputs a ciphertext c which is decapsulated into $k \neq \perp$ by dk_0 and dk_1, that is, $\overline{\mathsf{Dec}}(dk_0, c) = \overline{\mathsf{Dec}}(dk_1, c)$. Let us define $m_0 = \mathsf{Dec}(dk_0, c)$ and $m_1 = \mathsf{PKE}(dk_1, c)$. We have four cases defined as follows:

1. Case 1 ($m_0 \neq \perp \wedge m_1 \neq \perp$): We have two sub-cases:
   - $m_0 = m_1$: Let $m_0 = m_1 = (r, m) \in \mathcal{L}_r \times \mathcal{L}_m$. We have $h_0 \cdot r + \mathsf{Lift}(m) \equiv h_1 \cdot r + \mathsf{Lift}(m) \pmod{q}$. Thus, we have $r(h_0 - h_1) \equiv 0 \pmod{(q, \Phi_n)}$. However, for any $r \in \mathcal{L}_r = \mathcal{T}$, we have $r \neq 0 \in S/q$ (Lemma 6.1). In addition, we have $h_0 \equiv h_1 \in S/q$ with negligible probability. Thus, the probability that the adversary wins as this case is negligible.
   - $m_0 \neq m_1$: In this case, we succeed to find a collision for H, which is negligible for any QPT adversary (Lemma 2.2).
2. $m_0 = \perp \wedge m_1 \neq \perp$: In this case, we find a claw $((s_0, c), m_1)$ of H_0 and H_1. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.3).
3. $m_0 \neq \perp \wedge m_1 = \perp$: In this case, we find a claw $(m_0, (s_1, c))$ of H_0 and H_1. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.3).
4. $m_0 = m_1 = \perp$: In this case, we find a collision $((s_0, c), (s_1, c))$ of H_0, which is a collision if $s_0 \neq s_1$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.2).

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

9 Conclusion

We have shown that NTRU in NIST PQC Round 3 finalist is anonymous in the QROM if the underlying NTRU PKE is strongly disjoint-simulatable and a hybrid PKE scheme constructed from NTRU as KEM and appropriate DEM is anonymous and robust.

We show that

- SPR-CCA-secure KEM and PKE is ANON-CCA-secure (section 3).
- SPR-CCA-secure and SPR2-CCA-secure KEM and SPR-otCCA-secure DEM lead to SPR-CCA-secure PKE (section 4).
- KEM obtained by the SXY transformation is SPR-CCA-secure and SPR2-CCA-secure if the underlying PKE is strongly disjoint-simulatable in the QROM.(section 5).
- NTRU is SPR-CCA-secure and SPR2-CCA-secure if the underlying NTRU OWF is strongly disjoint-simulatable(section 6 and section 7).
- NTRU is also SCFR-CCA-secure (section 8).
- Hence, NTRU leads to ANON-CCA-secure hybrid PKE and SROB-CCA-secure hybrid PKE.

Grubbs et al. [GMP21] discussed the barrier to show anonymity of NTRU, which stems from the design choice $K = \mathsf{H}(m)$ instead of $K = \mathsf{H}(m, c)$. The former choice make their simulation difficult. We avoid this technical barrier by using SPR-CCAsecurity.

References

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. 1

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. 1, 3

BDH+05. Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM J. Comput.*, 34(6):1324–1330, 2005. 2

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. 3, 5

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. 12, 13

CFA05. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. 2005. 13

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. 8, 9

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. 5

GMP21. 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. 1, 5, 7, 13, 14

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. 2, 10

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. 3

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. 3, 5

Hos20. Akinori Hosoyamada. personal communication, June 2020. 2

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. 1

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. 13

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. 3

Moh10. Payman Mohassel. A closer look at anonymity and robustness in encryption schemes. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 501–518. Springer, Heidelberg, December 2010. 1, 3, 7

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. 3, 5

Sch20. John Schanck. personal communication, June 2020. 12

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. 1, 3, 10, 13

Tan09. Seiichiro Tani. Claw finding algorithms using quantum walk. *Theor. Comput. Sci.*, 410(50):5285–5297, 2009. 2

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. 3, 5

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. 2, 10, 11, 12

Zha05. Shengyu Zhang. Promised and distributed quantum search. In Lusheng Wang, editor, *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16-29, 2005, Proceedings*, volume 3595 of *Lecture Notes in Computer Science*, pages 430–439. Springer, 2005. 2

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7–8):557–567, May 2015. 2