NTRU leads to Anonymous, Robust Public-Key Encryption

Keita Xagawa

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 open problem to investigate the anonymity and robustness of NTRU posed by Grubbs, Maram, and Paterson (Cryptography ePrint Archive 2021/708).

Keywords: anonymity, robustness, post-quantum cryptography, 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 $\kappa$. 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 $f(\kappa) = \kappa^{-o(1)}$. We denote a set of negligible functions by $\negl(\kappa)$. For a distribution $\chi$, we often write “$x \leftarrow \chi$,” which indicates that we take a sample $x$ according to $\chi$. 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 $A$, $A(S)$ denotes the set $\{A(x) \mid x \in S\}$. If $\text{inp}$ is a string, then “$\text{out} \leftarrow A(\text{inp})$” denotes the output of algorithm $A$ when run on input $\text{inp}$. If $A$ is deterministic, then $\text{out}$ is a fixed value and we write “$\text{out} : = A(\text{inp})$.” We also use the notation “$\text{out} : = A(\text{inp}; r)$” to make the randomness $r$ explicit.

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

Quantum Random Oracle Model:

Lemma 2.1. Let $\ell$ be a positive integer. Let $X$ and $Y$ be finite sets. Let $H_0: \{0, 1\}^\ell \times X \rightarrow Y$ and $H_q : X \rightarrow Y$ be two independent random oracles. If an unbounded time quantum adversary $A$ makes a query to $H$ at most $Q$ times, then we have

$$\Pr[s \leftarrow \{0, 1\}^\ell : A^{H_0, H_q(s, \cdot)}() \rightarrow 1] - \Pr[A^{H_0, H_q}() \rightarrow 1] \leq Q \cdot 2^{-\ell/2},$$

where all oracle accesses of $A$ 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 \( X \) and \( Y \) be finite sets. Let \( H: X \to Y \) be a random oracle. If an unbounded time quantum adversary \( \mathcal{A} \) makes a query to \( H \) at most \( Q \) times, then we have
\[
\Pr_{H, \mathcal{A}}[(x, x') \leftarrow \mathcal{A}^H : x \neq x' \land H(x) = H(x')] \leq C(Q + 1)^3/|Y|,
\]
where all oracle accesses of \( \mathcal{A} \) can be quantum.

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

Lemma 2.3 (QRO is claw-free). There is a universal constant C such that the following holds: Let \( X_0 \) and \( X_1 \) and \( Y \) be finite sets. Let \( N_0 = |X_0| \) and \( N_1 = |X_1| \). Without loss of generality, we assume \( N_0 \leq N_1 \). Let \( H_0: X_0 \to Y \) and \( H_1: X_1 \to Y \) 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
\[
\Pr[(x_0, x_1) \leftarrow \mathcal{A}^{H_0, H_1} : H_0(x_0) = H_1(x_1)] \leq C(Q_0 + Q_1 + 1)^3/|Y|,
\]
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 \( X_0 \) and \( X_1 \) are enumerable. Given \( H: [N_0 + N_1] \to Y \), we define \( H_0: X_0 \to Y \) and \( H_1: X_1 \to Y \) by \( H_0(x) = H(\text{index}_0(x)) \) and \( H_1(x) = H(\text{index}_1(x) + N_0) \), where \( \text{index}_x: X_x \to [N_x] \) is an index function which returns the index of \( x \) in \( X_x \). \( H_0 \) 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 \((\text{index}_0(x_0), \text{index}_1(x_1) + N_0)\) for \( H \) with \( Q_0 + Q_1 \) queries. Using Lemma 2.3, we obtain the bound as we wanted. \( \square \)

The best upper bound for the claw-finding problem is given by Tani [Tan09]. His algorithm runs in \( O((N_0N_1)^{1/3}) \) if \( N_0 \leq N_1 < N_0^2 \) and \( O(N_1^{1/2}) \) 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 \( \text{PKE} \) consists of the following triple of PPT algorithms (Gen, Enc, Dec).
- Gen\(1^k\) → (ek, dk): a key-generation algorithm that on input \(1^k\), where \( k \) is the security parameter, and randomness \( r \in R_{\text{Gen}} \), outputs a pair of keys (ek, dk). ek and dk are called the encryption key and decryption key, respectively.
- Enc(ek, m, r_e) → c: an encryption algorithm that takes as input encryption key ek, message \( m \in M \), and randomness \( r_e \in R_{\text{Enc}} \), and outputs ciphertext \( c \in C \).
- Dec(\(dk, c\)) → m ∪ ⊥: a decryption algorithm that takes as input decryption key dk and ciphertext \(c\) and outputs message \( m \in M \) or a rejection symbol \( ⊥ \notin M \).

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

Definition 2.2 (\(δ\)-Correctness). Let \( δ = δ(κ) \). We say \( \text{PKE} = (\text{Gen, Enc, Dec}) \) is \( δ \)-correct if
\[
\Pr_{\text{PKE}}[(\text{Enc}(ek, m), c) 
\leq \delta.
\]
In particular, we say that \( \text{PKE} \) is perfectly correct if \( δ = 0 \).

We also define a key pair’s accuracy.

Definition 2.3 (Accuracy [XY19]). We say that a key pair \((ek, \text{dk})\) is accurate if for any \( m \in M \),
\[
\Pr_{c \leftarrow \text{Enc}(ek, m)}[\text{Dec}(\text{dk}, c) = m] = 1.
\]

Remark 2.2. Xagawa and Yamakawa [XY19] observed that if \( \text{PKE} \) is deterministic, then \(δ\)-correctness implies that
\[
\Pr_{\text{PKE}}[(\text{Enc}(ek, m), c) \text{ is inaccurate}] \leq \delta.
\]
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 $S$ 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 $PKE = (Gen, Enc, Dec)$ be a PKE scheme. Let $D_M$ be a distribution over the message space $M$.

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

$$Adv_{PKE, A}^{\text{goal-atk}}(\kappa) = \left| \Pr[Exp_{PKE, A}^{\text{goal-atk}}(\kappa) = 1] - 1/2 \right|,$$

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

For any $A$ and goal-atk $\in \{ow$-cpa, scfr-cca, wcfr-cca, wcf-cca$, we define its goal-atk advantage against PKE as follows:

$$Adv_{PKE, D_M, A}^{\text{goal-atk}}(\kappa) := \Pr[Exp_{PKE, D_M, A}^{\text{goal-atk}}(\kappa) = 1],$$

where $Exp_{PKE, D_M, A}^{\text{goal-atk}}(\kappa)$ is an experiment described in Figure 1.

For $\text{GOAL-ATK} \in \{\text{OW-CPA, IND-CPA, IND-CCA, PR-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA}\}$, we say that PKE is $\text{GOAL-ATK}$-secure if $Adv_{PKE, D_M, A}^{\text{goal-atk}}(\kappa)$ is negligible for any QPT adversary $A$.

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 $D_M$ denote an efficiently sampleable distribution on a set $M$. A deterministic PKE scheme $PKE = (Gen, Enc, Dec)$ with plaintext and ciphertext spaces $M$ and $C$ is disjoint-simulatable if there exists a PPT algorithm $S$ that satisfies the following:

- (Statistical disjointness)
  $$\text{Disj}_{PKE, S}(\kappa) := \max_{(ek, dk) \in \text{Gen}(\kappa)} \Pr[c \leftarrow S(1^\kappa, ek) : c \in \text{Enc}(ek, M)]$$
  is negligible.

- (Ciphertext-indistinguishability) For any QPT adversary $A$,
  $$Adv_{\text{PKE}, D_M, A, S}^{\text{ind}}(\kappa) := \left| \Pr[(ek, dk) \leftarrow \text{Gen}(1^\kappa), m^* \leftarrow D_M, c^* \leftarrow \text{Enc}(ek, m^*) : A(ek, c^*) \rightarrow 1] \right.$$ $$\left. - \Pr[(ek, dk) \leftarrow \text{Gen}(1^\kappa), c \leftarrow S(1^\kappa, ek) : A(ek, c) \rightarrow 1] \right|$$

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

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

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, $\text{Enc}(ek, \hat{m})$.

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 $(\text{Gen}, \text{Enc}, \text{Dec})$:

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

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

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

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

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|}
\hline
Expt_{\text{OW-CPA}}^{\text{PKE}}(\mathcal{A}, \mathcal{M}, \kappa) & Expt_{\text{IND-CPA}}^{\text{PKE}}(\mathcal{A}, \kappa) & Dec_{\text{a}}(c) \\
\hline
(ek, dk) \leftarrow \text{Gen}(1^\kappa) & b \leftarrow \{0, 1\} & \text{if} \ c = a, \text{return } \bot \\
m^* \leftarrow D_M & (ek, dk) \leftarrow \text{Gen}(1^\kappa) & m := \text{Dec}(dk, c) \\
c' \leftarrow \text{Enc}(ek, m^*) & (m_0, m_1, \text{state}) \leftarrow \mathcal{A}_1(ek) & \text{return } m \\
m' \leftarrow \mathcal{A}(ek, c^*) & c^* \leftarrow \text{Enc}(ek, m_b) & m := \text{Dec}(dk_{id}, c) \\
& b' \leftarrow \mathcal{A}_2(c^*, \text{state}) & \text{return } m \\
\text{return } \text{boole}(m' \not= \text{Dec}(d, c^*)) & & \text{boole}(b = b') \\
\hline
Expt_{\text{IND-CCA}}^{\text{PKE}}(\mathcal{A}, \kappa) & Expt_{\text{CPA}}^{\text{PKE}}(\mathcal{A}, \kappa) & Expt_{\text{IND-CCA}}^{\text{PKE}}(\mathcal{A}, \kappa) \\
\hline
b \leftarrow \{0, 1\} & b \leftarrow \{0, 1\} & b \leftarrow \{0, 1\} \\
(ek, dk) \leftarrow \text{Gen}(1^\kappa) & (ek, dk) \leftarrow \text{Gen}(1^\kappa) & (ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa) \\
(m_0, m_1, \text{state}) \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek) & (m, \text{state}) \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek) & (ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa) \\
c^* \leftarrow \text{Enc}(ek, m_b) & c_0^* \leftarrow \text{Enc}(ek, m) & (m, \text{state}) \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek_0, ek_1) \\
b' \leftarrow \mathcal{A}_2^{\text{Dec}(-)}(c^*, \text{state}) & c^* \leftarrow S(1^\kappa, ek) & c^* \leftarrow \text{Enc}(ek_b, m) \\
\text{return } \text{boole}(b = b') & b' \leftarrow \mathcal{A}_2^{\text{Dec}(-)}(c^*, \text{state}) & \text{return } \text{boole}(b = b') \\
& \text{return } \text{boole}(b = b') & \text{boole}(b = b') \\
\hline
Expt_{\text{SCFR-CPA}}^{\text{PKE}}(\mathcal{A}, \kappa) & Expt_{\text{SCFR-CPA}}^{\text{PKE}}(\mathcal{A}, \kappa) & Expt_{\text{SCFR-CPA}}^{\text{PKE}}(\mathcal{A}, \kappa) \\
\hline
(ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa) & (ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa) & (ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa) \\
(ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa) & (ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa) & (ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa) \\
c \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek_0, ek_1) & c \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek_0, ek_1) & (m, \text{state}) \leftarrow \mathcal{A}_1^{\text{Dec}(-)}(ek_0, ek_1) \\
m_0 \leftarrow \text{Dec}(dk_0, c) & m_0 \leftarrow \text{Dec}(dk_0, c) & c \leftarrow \text{Enc}(ek_b, m) \\
m_1 \leftarrow \text{Dec}(dk_1, c) & m_1 \leftarrow \text{Dec}(dk_1, c) & m' \leftarrow \text{Dec}(dk_{1-b}, c) \\
\text{return } \text{boole}(m_0 \neq \bot \land m_1 \neq \bot) & \text{return } \text{boole}(m_0 = m_1 \neq \bot) & \text{return } \text{boole}(m' \neq \bot) \\
\hline
\end{tabular}
\caption{Games for PKE schemes}
\end{table}
Security: We review one-wayness 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 $S$ as a generalization of IND$\text{-}$CCA-security in [vH04, Hop05]. We also review anonymity (ANON-CCA), robustness (WROB$\text{-}$CCA and SROB$\text{-}$CCA), and collision-freeness (WCFR$\text{-}$CCA and SCFR$\text{-}$CCA) [GMP21].

Definition 2.9 (Security notions for KEM). Let KEM = $(Gen, Enc, Dec)$ be a KEM scheme.
For any $A$ and goal-atk $\in \{\text{ind-cca, ind-cca, pr-cca, pr2-cca, anon-cca, srob-cca, scfr-cca}\}$, we define its goal-atk advantage against KEM as follows:

$$Adv_{\text{KEM, } A}^{\text{goal-atk}}(\kappa) := \left| \Pr[Expt_{\text{KEM, } A}^{\text{goal-atk}}(\kappa) = 1] - 1/2 \right|,$$

where $Expt_{\text{KEM, } A}^{\text{goal-atk}}(\kappa)$ is an experiment described in Figure 1.

For any $A$ and goal-atk $\in \{\text{srob-cca, scfr-ccawrob-cca, wcfrc-cca}\}$, we define its goal-atk advantage against KEM as follows:

$$Adv_{\text{KEM, } A}^{\text{goal-atk}}(\kappa) := \Pr[Expt_{\text{KEM, } A}^{\text{goal-atk}}(\kappa) = 1],$$

where $Expt_{\text{KEM, } A}^{\text{goal-atk}}(\kappa)$ is an experiment described in Figure 1.

For $\text{GOAL-ATK} \in \{\text{IND-CPA, IND-CCA, PR-CCA, PR2-CCA, ANON-CCA, SROB-CCA, SCFR-CCA, WROB-CCA, WCFR-CCA}\}$, we say that KEM is $\text{GOAL-ATK}$-secure if $Adv_{\text{KEM, } A}^{\text{goal-atk}}(\kappa)$ is negligible for any QPT adversary $A$. We also say that KEM is $\text{PR-CCA}$-secure (or $\text{PR2-CCA}$-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 $K$ and message space $M$:

- $E(K, m) \rightarrow d$: an encapsulation algorithm that takes as input key $K$ and data $m$ and outputs ciphertext $d$.
- $D(K, d) \rightarrow m/\perp$: a decapsulation algorithm that takes as input key $K$ and ciphertext $c$ and outputs data $m$ or a rejection symbol $\perp \not\in M$.

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

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

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

Definition 2.12 (Security notions for DEM). Let DEM = $(E, D)$ be a DEM scheme whose key space is $K$.

For any $A$ and goal-atk $\in \{\text{ind-cca, pr-cca, pr-otcca}\}$, we define its goal-atk advantage against DEM as follows:

$$Adv_{\text{DEM, } A}^{\text{goal-atk}}(\kappa) := \left| \Pr[Expt_{\text{DEM, } A}^{\text{goal-atk}}(\kappa) = 1] - 1/2 \right|,$$

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

For any $A$ and goal-atk $\in \{\text{frob, xrob}\}$, we define its goal-atk advantage against DEM as follows:

$$Adv_{\text{DEM, } A}^{\text{goal-atk}}(\kappa) := \Pr[Expt_{\text{DEM, } A}^{\text{goal-atk}}(\kappa) = 1],$$

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

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

5
<table>
<thead>
<tr>
<th>Game</th>
<th>Description</th>
<th>Condition</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Expt}^\text{ind-cca}_{\text{KEM},\mathcal{A}}(x)$</td>
<td>$b \leftarrow {0,1}$</td>
<td>$(ek, dk) \leftarrow \text{Gen}(1^k)$</td>
<td>$\text{Dec}_A(c)$ if $c = a$, return $\bot$</td>
</tr>
<tr>
<td>$\text{Expt}^\text{ind-cca}_{\text{KEM},\mathcal{A}}(x)$</td>
<td>$b \leftarrow {0,1}$</td>
<td>$(ek, dk) \leftarrow \text{Gen}(1^k)$</td>
<td>$\text{Dec}_A(id, c)$ if $c = a$, return $\bot$</td>
</tr>
<tr>
<td>$\text{Dec}_A(c)$</td>
<td></td>
<td></td>
<td>$K := \text{Dec}(dk, c)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$K := \text{Dec}(dk_{id}, c)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>return $K$</td>
</tr>
<tr>
<td>$\text{Dec}_A(id, c)$</td>
<td></td>
<td></td>
<td>return $K$</td>
</tr>
<tr>
<td>$b' \leftarrow \mathcal{A}(ek, c^<em>, K_0^</em>)$</td>
<td></td>
<td></td>
<td>return boole($b = b'$)</td>
</tr>
<tr>
<td>$b' \leftarrow \mathcal{A}^{\text{Dec}_c(\cdot)}(ek, c^<em>, K_0^</em>)$</td>
<td></td>
<td></td>
<td>return boole($b = b'$)</td>
</tr>
</tbody>
</table>

Fig. 2. Games for KEM 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] from Mohassel [Moh10] is they treat KEM with implicit rejection, which is 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 FQ with PKE₁, KEM is ANON-CCA-secure and IND-CCA-secure, PKE₁ 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 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 Game₁,b for i, b ∈ {0, 1}. Let S₁,b be the event that the adversary outputs 1 in Game₁,b.

- Game₀,b for b ∈ {0, 1}: This is the original game Exp⁺_{PKE, A}(k) with b = 0 and 1.
- Game₁,b for b ∈ {0, 1}: This game is the same as Game₀,b except that the target ciphertext is randomly taken from \( S(1^k) \times C_{DEM, |m|} \).
Table 1. Summary of Games for the Proof of Theorem 4.1

<table>
<thead>
<tr>
<th>Game</th>
<th>$c^<em>$ and $K^</em>$</th>
<th>$d^*$</th>
<th>Decryption oracle</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Game}_0$</td>
<td>$\text{Enc}(ek)$</td>
<td>$E(K^<em>, m^</em>)$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td></td>
</tr>
<tr>
<td>$\text{Game}_1$</td>
<td>$\text{Enc}(ek)$ at first</td>
<td>$E(K^<em>, m^</em>)$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>$\text{Game}_2$</td>
<td>$\text{Enc}(ek)$ at first</td>
<td>$E(K^<em>, m^</em>)$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td>SPR2/CCA security of KEM</td>
</tr>
<tr>
<td>$\text{Game}_3$</td>
<td>$\mathcal{S}(1^k) \times \mathcal{K}$ at first</td>
<td>$E(K^<em>, m^</em>)$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td>SPR2/CCA security of DEM</td>
</tr>
<tr>
<td>$\text{Game}_4$</td>
<td>$\mathcal{S}(1^k) \times \mathcal{K}$ at first</td>
<td>$U(C_{\text{DEM},[m^*]})$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td>SPR2/CCA security of KEM</td>
</tr>
<tr>
<td>$\text{Game}_5$</td>
<td>$\mathcal{S}(1^k) \times \mathcal{K}$</td>
<td>$U(C_{\text{DEM},[m^*]})$</td>
<td>reject if $(c, d) = (c^<em>, d^</em>)$</td>
<td>conceptual change</td>
</tr>
</tbody>
</table>

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

$$\frac{1}{2} |\Pr[S_{0, b}] - \Pr[S_{1, b}]| \leq \text{Adv}^{\text{spr-cca}}_{\text{PKE}, \mathcal{A}_{10}}(\kappa) \text{ and } \Pr[S_{1, 0}] = \Pr[S_{1, 1}].$$

Hence, we have

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

This completes the proof. $\square$

4 Strong Pseudorandomness of Hybrid PKE

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

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

Proof: In the following, we consider $\text{Game}_i$ 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 $\text{Game}_i$.

$\text{Game}_0$: This is the original game $\text{Exp}^{\text{spr-cca}}_{\text{PKE}, \mathcal{A}}(\kappa)$ with $b = 0$. The target ciphertext is computed as follows:

$$(c_0^*, K_0^*) \leftarrow \text{Enc}(ek); d_0^* \leftarrow E(K_0^*, m^*); \text{return } ct^* = (c_0^*, d_0^*).$$

We have

$$\Pr[S_0] = 1 - \Pr[\text{Exp}^{\text{spr-cca}}_{\text{PKE}, \mathcal{A}}(\kappa) = 1 \mid b = 0].$$
Game 1: In this game, $c_0^*$ and $K_0^*$ are generated before invoking $\mathcal{A}$ with $ek$. This 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 $\text{Dec}(sk, c^*)$. 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 $\delta$. Hence, the difference of Game 1 and Game 2 is bounded by $\delta$ and we have
\[ |Pr[S_1] - Pr[S_2]| \leq \delta. \]
This is corresponding to the event $\text{BadKeyPair}$ 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:
\[ (c_1^*, K_1^*) \leftarrow S(1^k) \times K; d^* \leftarrow E(K_1^*, m^*) \text{; return } ct^* = (c_1^*, d^*). \]
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
\[ \frac{1}{2} |Pr[S_2] - Pr[S_3]| \leq \text{Adv}^{\text{spr-cca}}_{\text{KEM}, \mathcal{A}_{23}}(k). \]
(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:
\[ (c_1^*, K_1^*) \leftarrow S(1^k) \times K; d_1^* \leftarrow C_{\text{DEM}, |m|}; \text{ return } ct^* = (c_1^*, d_1^*). \]
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
\[ \frac{1}{2} |Pr[S_3] - Pr[S_4]| \leq \text{Adv}^{\text{spr-otcca}}_{\text{DEM}, \mathcal{A}_{34}}(k). \]
(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 $K = \text{Dec}(sk, c^*)$ 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
\[ \frac{1}{2} |Pr[S_4] - Pr[S_5]| \leq \text{Adv}^{\text{spr2-cca}}_{\text{DEM}, \mathcal{A}_{45}}(k). \]
(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 $\text{Expt}^{\text{spr-cca}}_{\text{PKE}, \mathcal{A}}(k)$ with $b = 1$, thus, we have
\[ Pr[S_0] = Pr[\text{Expt}^{\text{spr-cca}}_{\text{PKE}, \mathcal{A}}(k) = 1 \mid b = 1] \]
Summarizing the (in)equalities, we obtain the bound in the statement. \(\square\)
As in \[XY19, \text{Lemmas 4.1}\], from Lemma 2.1 we have the bound

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

5 SXY may be Strongly Pseudorandom in the QROM

Let us review SXY [XY18] as known as $U_K^M$ [HHK17]. (We note that SXY requires the re-encryption check but $U_K^M$ does not.) Let $\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ be a PKE scheme whose plaintext space is $\mathcal{M}$. Let $\mathcal{C}$ and $\mathcal{K}$ be a ciphertext and key space. Let $H : \mathcal{M} \to \mathcal{K}$ and $H' : \{0, 1\}^* \times \mathcal{C} \to \mathcal{K}$ be hash functions modeled by random oracles. $\text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{SXY}[\text{PKE}, H, H_0]$ is defined as follows:

<table>
<thead>
<tr>
<th>Game</th>
<th>$H$</th>
<th>$c^*$</th>
<th>$K^*$</th>
<th>Decryption valid $c$ invalid $c$ justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game$_0$</td>
<td>$H(\cdot)$</td>
<td>$\text{Enc}(ek, m^*)$</td>
<td>$H(m^*)$</td>
<td>$H_0(s, c)$</td>
</tr>
<tr>
<td>Game$_1$</td>
<td>$H(\cdot)$</td>
<td>$\text{Enc}(ek, m^*)$</td>
<td>$H(m^*)$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_{1,5}$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{Enc}(ek, m^*)$</td>
<td>$H(m^*)$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_2$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{Enc}(ek, m^*)$</td>
<td>$H(m^*)$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_3$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{S}(1^*)$</td>
<td>$H_q(c^*)$</td>
<td>$H_q(c)$</td>
</tr>
<tr>
<td>Game$_4$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{S}(1^*)$</td>
<td>$H_q(c^*)$</td>
<td>$H_q(c)$</td>
</tr>
<tr>
<td>Game$_5$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{S}(1^*)$</td>
<td>$\text{random}$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_{6,5}$</td>
<td>$H_q(\text{Enc}(ek, \cdot))$</td>
<td>$\text{S}(1^*)$</td>
<td>$\text{random}$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_7$</td>
<td>$H(\cdot)$</td>
<td>$\text{S}(1^*)$</td>
<td>$\text{random}$</td>
<td>$H(m)$</td>
</tr>
<tr>
<td>Game$_{8}$</td>
<td>$H(\cdot)$</td>
<td>$\text{S}(1^*)$</td>
<td>$\text{random}$</td>
<td>$H(m)$</td>
</tr>
</tbody>
</table>

**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 $\text{Acc}$ and $\overline{\text{Acc}}$ denote the event that the key pair $(ek, dk)$ is accurate and inaccurate, respectively.

<table>
<thead>
<tr>
<th>$\text{Gen}(1^k)$</th>
<th>$\text{Enc}(ek)$</th>
<th>$\text{Dec}(dk, c)$, where $\overline{dk} = (dk, ek, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(ek, dk) \leftarrow \text{Gen}(1^k)$</td>
<td>$m \leftarrow M$</td>
<td>$m' \leftarrow \text{Dec}(dk, c)$</td>
</tr>
<tr>
<td>$s \leftarrow {0, 1}^\ell$</td>
<td>$c := \text{Enc}(ek, m)$</td>
<td>if $m' = \perp$, return $K := H_0(s, c)$</td>
</tr>
<tr>
<td>$\overline{dk} := (dk, ek, s)$</td>
<td>$K := H(m)$</td>
<td>if $c \neq \text{Enc}(pk, m')$, return $K := H_0(s, c)$</td>
</tr>
<tr>
<td>return $(ek, \overline{dk})$</td>
<td>return $(c, K)$</td>
<td>else return $K := H(m')$</td>
</tr>
</tbody>
</table>

We mainly follow the security proof in [XY19].

**Game$_0$:** This game is the original game $\text{Expt}_{\text{KEM, } \mathcal{A}}^{\text{spr-cca}}(k)$ with $b = 0$. Thus, we have

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

**Game$_1$:** This game is the same as Game$_0$ except that $H_0(s, c)$ in the decapsulation oracle is replace with $H_q(c)$ where $H_q : \mathcal{C} \to \mathcal{K}$ 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

\[
|\Pr[S_0] - \Pr[S_1]| \leq q_{H_0} \cdot 2^{-(\ell - 1)/2},
\]

where $q_{H_0}$ denote the number of queries to $H_0$ the adversary makes.
**Game_{1,5}:** This game is the same as Game_{1} except that the random oracle \(H(\cdot)\) is simulated by \(H'_q(\text{Enc}(ek, \cdot))\) where \(H'_q: C \rightarrow \mathcal{K}\) is yet another random oracle. We remark that the decapsulation oracle and the generation of \(K^*\) also use \(H'_q(\text{Enc}(ek, \cdot))\) as \(H(\cdot)\).

If a key pair is accurate, the two games Game_{1} and Game_{1,5} are equivalent because \(\text{Enc}(ek, \cdot)\) is injective. See \([XY19, \text{Lemma 4.3}]\) for the detail.

**Game_{2}:** This game is the same as Game_{1} except that the random oracle \(H(\cdot)\) is simulated by \(H_q(\text{Enc}(ek, \cdot))\) instead of \(H'_q(\text{Enc}(ek, \cdot))\).

If a key pair is accurate, the two games Game_{1,5} and Game_{2} are equivalent as in the proof of \([XY19, \text{Lemma 4.4}]\).

**Game_{3}:** This game is the same as Game_{2} except that \(K^*\) is set as \(H_q(c^*)\) and the decapsulation oracle always returns \(H'_q(c)\) as long as \(c \neq c^*\). This decapsulation oracle will denoted by \(\text{Dec}'\).

If a key pair is accurate, the two games Game_{2} and Game_{3} are equivalent as in the proof of \([XY19, \text{Lemma 4.5}]\).

**Game_{4}:** This game is the same as Game_{3} except that \(c^*\) is generated by \(S(1^*)\).

The difference between two games Game_{3} and Game_{4} is bounded by the advantage of ciphertext indistinguishability in disjoint simulatability as in \([XY19, \text{Lemma 4.7}]\).

**Game_{5}:** This game is the same as Game_{4} except that \(K^* \leftarrow \mathcal{K}\) instead of \(K^* \leftarrow H_q(c^*)\).

In Game_{4}, if \(c^* \leftarrow S(1^*)\) is not in \(\text{Enc}(ek, M)\), then the adversary has no information about \(K^* = H_q(c^*)\) 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 \([XY19, \text{Lemma 4.8}]\).

**Game_{6}:** This game is the same as Game_{4} except that the decapsulation oracle is reset as \(\text{Dec}\).

If a key pair is accurate, the two games Game_{5} and Game_{6} are equivalent as in the proof of \([XY19, \text{Lemma 4.5}]\).

**Game_{6,5}:** This game is the same as Game_{6} except that the random oracle \(H(\cdot)\) is simulated by \(H'_q(\text{Enc}(ek, \cdot))\) where \(H'_q: C \rightarrow \mathcal{K}\) is yet another random oracle as in Game_{1,5}.

If a key pair is accurate, the two games Game_{6} and Game_{6,5} are equivalent as in the proof of \([XY19, \text{Lemma 4.4}]\).

**Game_{7}:** This game is the same as Game_{6,5} except that the random oracle \(H\) is chosen from \(\{H: M \rightarrow \mathcal{K}\}\).

If a key pair is accurate, the two games Game_{6,5} and Game_{7} are equivalent because \(\text{Enc}(ek, \cdot)\) is injective. See \([XY19, \text{Lemma 4.3}]\) for the detail.

**Game_{8}:** This game is the same as Game_{7} except that \(H_q(c)\) in the decapsulation is replaced by \(H_0(s, c)\).

As in \([XY19, \text{Lemmas 4.1}]\), from \textbf{Lemma 2.1} we have the bound

\[
\Pr[S_b] - \Pr[S_b^\prime] \leq q_{H_0} \cdot 2^{-(l-1)/2}.
\]

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

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

**SPR2-CCA security:**

\textbf{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.

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

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

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

We have

\[
\Pr[S_0] = 1 - \Pr[\text{Expt}_{\text{KEM, \mathcal{A}}}^{\text{spr-cca}}(\kappa) = 1 | b = 0].
\]
Table 3. Summary of Games for the Proof of Theorem 5.1: \(S(1^k) \setminus \text{Enc}(ek, M)\) implies that the challenger generates \(c^* \leftarrow S(1^k)\) and returns \(\perp\) if \(c^* \in \text{Enc}(ek, M)\).

| Game | \(H\) | \(c^*\) | \(K^*\) | Decryption
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{Game}_0) (H(\cdot)) (S(1^k))</td>
<td>(\text{rand})</td>
<td>(H(m))</td>
<td>(H_0(s,c))</td>
<td>(\text{valid}) (c) (\text{invalid}) (c) (\text{justification})</td>
</tr>
<tr>
<td>(\text{Game}_1) (H(\cdot)) (S(1^k)) \text{Enc}(ek, M)</td>
<td>(\text{rand})</td>
<td>(H(m))</td>
<td>(H_0(s,c))</td>
<td>(\text{statistical disjointness})</td>
</tr>
<tr>
<td>(\text{Game}_2) (H(\cdot)) (S(1^k)) \text{Enc}(ek, M)</td>
<td>(\text{rand})</td>
<td>(H(m))</td>
<td>(H_q(c))</td>
<td>(\text{Lemma 2.1})</td>
</tr>
<tr>
<td>(\text{Game}_3) (H(\cdot)) (S(1^k)) \text{Enc}(ek, M)</td>
<td>(H_0)</td>
<td>(H(m))</td>
<td>(H_q(c)) (H_q(c^*)) is hidden</td>
<td>(\text{Lemma 2.1})</td>
</tr>
<tr>
<td>(\text{Game}_4) (H(\cdot)) (S(1^k)) \text{Enc}(ek, M)</td>
<td>(\text{Pr})</td>
<td>(H(m))</td>
<td>(H_0(s,c))</td>
<td>(\text{re-encryption check and key’s accuracy})</td>
</tr>
<tr>
<td>(\text{Game}_5) (H(\cdot)) (S(1^k))</td>
<td>(\text{Dec}(d_k,c^*))</td>
<td>(H(m))</td>
<td>(H_0(s,c))</td>
<td>(\text{statistical disjointness})</td>
</tr>
</tbody>
</table>

**Game 1:** In this game, the ciphertext is set as \(\perp\) if \(c^*\) is in \(\text{Enc}(ek, M)\). The difference between two games \(\text{Game}_0\) and \(\text{Game}_1\) is bounded by statistical disjointness.

**Game 2:** This game is the same as \(\text{Game}_1\) except that \(H_0(s,c)\) in the decapsulation oracle is replaced with \(H_q(c)\) where \(H_q : C \rightarrow \mathcal{K}\) is another random oracle.

As in [XY19, Lemmas 4.1], from Lemma 2.1 we have the bound

\[
|\Pr[S_1] - \Pr[S_2]| \leq q_{t_{\chi}} \cdot 2^{-(\ell-1)/2},
\]

where \(q_{t_{\chi}}\) denote the number of queries to \(H_0\) the adversary makes.

**Game 3:** This game is the same as \(\text{Game}_2\) except that \(K^* := H_q(c^*)\) instead of chosen random. Since \(c^*\) is always outside of \(\text{Enc}(ek, M)\), \(\mathcal{A}\) cannot obtain any information about \(H_q(c^*)\). Hence, the two games \(\text{Game}_2\) and \(\text{Game}_3\) are equivalent.

**Game 4:** This game is the same as \(\text{Game}_3\) except that \(H_q(\cdot)\) is replaced by \(H_0(s, \cdot)\). As in [XY19, Lemmas 4.1], from Lemma 2.1 we have the bound

\[
|\Pr[S_3] - \Pr[S_4]| \leq q_{t_{\chi}} \cdot 2^{-(\ell-1)/2},
\]

where \(q_{t_{\chi}}\) denote the number of queries to \(H_0\) the adversary makes.

**Game 5:** This game is the same as \(\text{Game}_4\) except that \(K^* := \text{Dec}(d_k,c^*)\) instead of \(H_0(s,c^*)\). Recall that \(c^*\) is always outside of \(\text{Enc}(ek, M)\). If a key pair is accurate, then \(\text{Enc}(ek, \text{Dec}(c^*)) = c^*\) and \(K^* = H_0(s,c^*)\). 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 \(S(1^k)\) as in \(\text{Game}_0\). The difference between two games \(\text{Game}_5\) and \(\text{Game}_6\) is bounded by statistical disjointness.

Moreover, this game \(\text{Game}_6\) is the original game \(\text{Expt}_{\text{NTRU},\mathcal{A}}(\kappa)\) with \(b = 1\).

Summarizing the (in)equalities, we obtain

**Theorem 5.2**

### 6 Review of NTRU

Let us briefly review NTRU [CDH+20]. \(\Phi_1\) denotes the polynomial \(x - 1\) and \(\Phi_n\) 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 \(x^t - 1 = \Phi_1 \Phi_m, R, R/3,\) and \(R/q\) denote \(\mathbb{Z}[x]/(\Phi_1 \Phi_m, R, R/3, \mathbb{Z}[x]/(\Phi_1 \Phi_m, R/3),\) and \(\mathbb{Z}[x]/(q, \Phi_1 \Phi_m),\) respectively. \(S, S/3,\) and \(S/q\) denotes \(\mathbb{Z}[x]/(\Phi_1 \Phi_m, R, R/3, \mathbb{Z}[x]/(3, \Phi_1 \Phi_m),\) and \(\mathbb{Z}[x]/(q, \Phi_1 \Phi_m),\) respectively. \(\Sigma(x)\) returns a canonical \(S/3\)-representative of \(x \in \mathbb{Z}[x]\), that is, \(b \in \mathbb{Z}[x]\) of degree at most \(n - 2\) with ternary coefficients in \(\{-1, 0, +1\}\) such that \(a \equiv b \pmod{(3, \Phi_m)}\). Let \(T\) be a set of non-zero ternary polynomials of degree at most \(n - 2\), that is, \(T = \{a = \sum_{i=0}^{n-2} a_i x^i | a \neq 0 \wedge a_i \in \{-1, 0, +1\}\}\). We say a ternary polynomial \(v = \sum_i v_i x^i\) has the non-negative correlation property if \(\sum_i v_i x^{i+1} \geq 0\). \(T_e\) is a set of non-zero ternary polynomials of degree at most \(n - 2\) with non-negative correlation property. \(T(d)\) is a set of non-zero balanced ternary polynomials of degree at most \(n - 2\) with Hamming weight \(d\), that is, \(\{a \in T | \#(a_i: a_i = 1) = \#(a_i: a_i = -1) = d/2\}\).

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. \(\Phi_n\) 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. \(\square\)

<table>
<thead>
<tr>
<th>Gen(1* )</th>
<th>Enc((h, (r, m) \in \mathcal{L}_r \times \mathcal{L}_m))</th>
<th>Dec((\left((f, f_p, h_q), c\right)))</th>
</tr>
</thead>
<tbody>
<tr>
<td>((f, g) \leftarrow) Sample(_{fg()})</td>
<td>(m' \leftarrow) Lift((m))</td>
<td>if (c \not\equiv 0 \mod (q, \Phi_1)) then return ((0, 0, 1))</td>
</tr>
<tr>
<td>(f_q \leftarrow (1/f) \mod (q, \Phi_n))</td>
<td>(c \leftarrow (h \cdot r + m') \mod (q, \Phi_1\Phi_n))</td>
<td>(a \leftarrow (c \cdot f) \mod (q, \Phi_1\Phi_n))</td>
</tr>
<tr>
<td>(h \leftarrow (3 \cdot g \cdot f_q) \mod (q, \Phi_1\Phi_n))</td>
<td>return (c)</td>
<td>(m \leftarrow (a \cdot f_p) \mod (3, \Phi_n))</td>
</tr>
<tr>
<td>(h_q \leftarrow (1/h) \mod (q, \Phi_n))</td>
<td>(m' \leftarrow) Lift((m))</td>
<td>(m' \leftarrow) Lift((m))</td>
</tr>
<tr>
<td>(f_p \leftarrow (1/f) \mod (3, \Phi_n))</td>
<td>(r \leftarrow ((c - m') \cdot h_q) \mod (q, \Phi_n))</td>
<td>if ((r, m) \in \mathcal{L}_r \times \mathcal{L}_m) then return ((r, m, 0))</td>
</tr>
<tr>
<td>(ek := h, dk := (f, f_p, h_q))</td>
<td></td>
<td>else return ((0, 0, 1))</td>
</tr>
</tbody>
</table>

Fig. 4. The DPKE for NTRU

NTRU-HPS: The parameters are defined as follows:
\[
\mathcal{L}_f = \mathcal{T}, \mathcal{L}_r = \mathcal{T}(q/8 - 2), \mathcal{L}_r = \mathcal{T}, \mathcal{L}_m = \mathcal{T}(q/8 - 2),
\]
and Lift\((m)\) = \(m\). We note that \(h \equiv 0 \mod (q, \Phi_1)\), \(h\) is invertible in \(S/q\), and \(hr + m \equiv 0 \mod (q, \Phi_1)\). (See [CDH\*20, Section2.3].)

NTRU-HRSS-KEM: The parameters are defined as follows:
\[
\mathcal{L}_f = \mathcal{T}_s, \mathcal{L}_r = \mathcal{T}_s, \mathcal{L}_r = \mathcal{T}_s, \mathcal{L}_m = \mathcal{T}_s,
\]
and Lift\((m)\) = \(\Phi_1 \cdot \mathcal{S}_3(m/\Phi_1)\). We note that \(h \equiv 0 \mod (q, \Phi_1)\), \(h\) is invertible in \(S/q\), and \(hr + m \equiv 0 \mod (q, \Phi_1)\). (See [CDH\*20, Section2.3].)

Rigidity: Notice that we implicitly check \(hr + \text{Lift}(m) = c\) 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 \(S\) will output a random polynomial \(c \leftarrow R/q\) such that \(c \equiv 0 \mod (q, \Phi_1)\).

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 the 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, $\mathsf{Dec}(dk_0, c) = \mathsf{Dec}(dk_1, c)$. 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 ($m_0 \neq \perp \land m_1 \neq \perp$): We have two sub-cases:
   - $m_0 = m_1$: Let $m_0 = m_1 = (r, m) \in L_r \times L_m$. 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 $r \in L_r = T$, we have $r \neq 0 \in S/q$ (Lemma 6.1). In addition, we have $h_0 \equiv h_1 \pmod S/q$ 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. $m_0 \neq \perp \land m_1 = \perp$: In this case, we find a claw $((s_0, c), m_1)$ of $H_0$ and $H_1$. The probability that we find such a claw is negligible for any QPT adversary (Lemma 2.3).

3. $m_0 = \perp \land m_1 = \perp$: In this case, we find a claw $((m_0, s_1, c))$ of $H_0$ and $H_1$. The probability that we find such a 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 a collision is negligible for any QPT adversary (Lemma 2.2).

We conclude that the advantage of the adversary is negligible in any cases. \(\square\)

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 SPR-otCCA-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 $K = H(m)$ instead of $K = H(m, c)$. The former choice makes their simulation difficult. We avoid this technical barrier by using SPR-CCASecurity.

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.

References


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


