Anonymity of NIST PQC Round 3 KEMs*

Keita Xagawa

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

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

- NTRU is anonymous in the quantum random oracle model (QROM) if the underlying deterministic PKE is strongly disjoint-simulatable. NTRU is collision-free in the QROM. A hybrid PKE scheme constructed from NTRU as KEM and appropriate DEM is anonymous and robust. (Similar results for BIKE, FrodoKEM, HQC, NTRU LPrime, and SIKE hold except one of three parameter sets of HQC.)
- Classic McEliece is anonymous in the QROM if the underlying PKE is strongly disjoint-simulatable and a hybrid PKE scheme constructed from it as KEM and appropriate DEM is anonymous.
- Grubbs, Maram, and Paterson pointed out that Kyber and Saber have a gap in the current IND-CCA security proof in the QROM (EUROCRYPT 2022). We found that Streamlined NTRU Prime has another technical obstacle for the IND-CCA security proof in the QROM.

Those answer the open problem to investigate the anonymity and robustness of NIST PQC Round 3 KEMs posed by Grubbs, Maram, and Paterson (EUROCRYPT 2022).

We use strong disjoint-simulatability of the underlying PKE of KEM and strong pseudorandomness and smoothness/sparseness of KEM as the main tools, which will be of independent interest.

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

1 Introduction

Public-key encryption (PKE) allows us to send a message to a receiver confidentially if the receiver’s public key is available. However, a ciphertext of PKE may reveal the receiver’s public key, and the recipient of the ciphertext will be identified. This causes trouble in some applications, and researchers study the anonymity of PKE. Roughly speaking, PKE is said to be anonymous [BBDP01] if a ciphertext hides the receiver’s information. Anonymous primitive is often used in the context of privacy-enhancing technologies.

A ciphertext of anonymous PKE indicates (computationally) no information of a receiver. Thus, when a receiver receives a ciphertext, it should decrypt the ciphertext into a message and verify the message in order to check if the ciphertext is sent to the receiver or not. There may be a ciphertext from which two (or more) recipients can obtain messages in this situation, and this causes trouble in some applications, e.g., auction protocols [Sak00]. Intuitively speaking, PKE is said to be robust [ABN10] if only the intended receiver can obtain a meaningful message from a ciphertext.

Both anonymity and robustness are important and useful properties beyond the standard IND-CCA security. Anonymous PKE is an important building primitive for anonymous credential systems [CL01], auction protocols [Sak00], (weakly) anonymous authenticated key exchange [BCGPNP09, FSXY13, FSXY15, SSW20], and so on. Robust PKE has an application for searchable encryption [ABC+05] and auction [Sak00].

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

Grubbs, Maram, and Paterson [GMP21a] discussed anonymity and robustness of post-quantum KEM schemes and KEM/DEM framework in the quantum random oracle model (QROM). They also studied the anonymity and robustness of the hybrid PKE based on KEM with implicit rejection. On the variants of the Fujisaki-Okamoto (FO) transform [FO99, FO13], they showed that anonymity and collision-freeness of KEMs obtained by the FO transform with implicit rejection and its variant1, and they lead to anonymous, robust hybrid PKEs.

*

This article is based on an earlier article: Keita Xagawa: Anonymity of NIST PQC Round 3 KEMs, EUROCRYPT 2022, © IACR 2022

1 A variant of the FO transform with implicit rejection using ‘pre-key’ technique. They wrote “a variant of the FO⊥ transform” in their paper.
Table 1. Summary of anonymity and robustness of NIST PQC Round 3 KEM candidates (finalists and alternate candidates) and the hybrid PKEs using them. In the first row, IND = Indistinguishability, SPR = Strong Pseudorandomness, ANO = Anonymity, CF = Collision Freeness, and ROB = Robustness under chosen-ciphertext attacks in the QROM. Y = Yes, N = No, ? = Unknown. The underline implies our new findings.

<table>
<thead>
<tr>
<th>Name</th>
<th>IND</th>
<th>SPR</th>
<th>ANO</th>
<th>CF</th>
<th>ROB</th>
<th>ANO</th>
<th>ROB</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic McEliece [ABC^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Section K</td>
</tr>
<tr>
<td>NTRU [CDH^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section 5</td>
</tr>
<tr>
<td>BIKE [ABB^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section N</td>
</tr>
<tr>
<td>FrodoKEM [NAB^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section O</td>
</tr>
<tr>
<td>HQC-128/192 [AAB^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section P</td>
</tr>
<tr>
<td>HQC-256 [AAB^20]</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Section P</td>
</tr>
<tr>
<td>NTRU LPrime [BBC^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section R</td>
</tr>
<tr>
<td>SIKE [JAC^20]</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Section S</td>
</tr>
</tbody>
</table>

from appropriate assumptions. They also show anonymity and robustness of KEM obtained by a variant of the FO transform with explicit rejection and key-confirmation hash\(^2\) and show that it leads to anonymous, robust hybrid PKE from appropriate assumptions. They examined NIST PQC Standardization finalists (Classic McEliece [ABC^20], Kyber [SAB^20], NTRU [CDH^20], and Saber [DKR^20]). They showed the following results:

- **Classic McEliece**: They found that Classic McEliece is not collision-free. Since their anonymity proof in [GMP21a, Theorem 5] strongly depends on the collision-freeness of the underlying PKE, we cannot apply their anonymity proof to Classic McEliece. They also show that the hybrid PKE fails to achieve robustness since Classic McEliece is not collision-free.

- **Kyber**: They found that Kyber’s anonymity (and even IND-CCA security) has two technical obstacles (‘pre-key’ and ‘nested random oracles’) in the QROM.

- **NTRU**: NTRU’s anonymity has another technical obstacle: Their proof technique requires the computation of a key of KEM involving a message and a ciphertext, but, in NTRU, the computation of a key of NTRU involves only a message. The robustness of the hybrid PKE with NTRU is unclear.

- **Saber**: They insisted they show Saber’s anonymity and IND-CCA security and the robustness of the hybrid PKE with Saber in the QROM, because they considered that Saber employs the FO transform with ‘pre-key’. Unfortunately, Saber in [DKR^20] also uses both ‘pre-key’ and ‘nested random oracles’ as Kyber, and their proofs cannot be applied to Saber. See their slides [GMP21b]. (Fortunately, FrodoKEM can be shown anonymous and lead to anonymous, robust hybrid PKE, because FrodoKEM employs the FO transform with ‘pre-key’.)

Unfortunately, we do not know whether all four finalists are anonymous or not, although the much effort of Grubbs et al. and their clean and modular framework. Grubbs et al. left several open problems: One of them is the anonymity and robustness of NTRU; the other important one is the anonymity of Classic McEliece.

### 1.1 Our Contribution

We investigate anonymity and robustness of all NIST PQC Round 3 KEM candidates and obtain Table 1. This answers the open problems posed by Grubbs et al. In order to investigate anonymity, we first study strong pseudorandomness of PKE/KEM instead of studying anonymity directly. To show strong pseudorandomness of the hybrid PKE, we study strong pseudorandomness and introduce smoothness and sparseness of KEM. We then show such properties of KEM obtained by the variants of the FO transform if the underlying deterministic PKE is strongly disjoint-simulatable. We finally study the properties of NIST PQC Round 3 KEM candidates. See the details in the following.

**Anonymity through strong pseudorandomness, sparseness, and smoothness**: Our starting point is strong pseudorandomness instead of anonymity. We say PKE/KEM/DEM is strongly pseudorandom if its ciphertext is

\[^2\] They modify ‘key-confirmation hash’ to involve a ciphertext on input.
indistinguishable from a random string chosen by a simulator on input the security parameter.\(^3\) It is easy to show that strong pseudorandomness implies anonymity. Using this notion, we attempt to follow the IND-CCA security proof of the KEM/DEM framework [CS02], that is, we try to show that the hybrid PKE from strongly pseudorandom KEM/DEM is also strongly pseudorandom, which implies that the hybrid PKE is anonymous. If we directly try to prove the ANON-CCA security of the hybrid PKE, then we will need to simulate two decryption oracles as Grubbs et al. Considering pseudorandomness allows us to treat a single key and oracle and simplifies the security proof. Unfortunately, we face another obstacle in the security proof when considering pseudorandomness. To resolve the obstacle, we define *sparseness* of KEM with explicit rejection and *smoothness* of KEM with implicit rejection: We say KEM with explicit rejection is sparse if a ciphertext \(c\) chosen by a simulator is decapsulated into \(\perp\) with overwhelming probability. We say KEM with implicit rejection is smooth if, given a ciphertext \(c\) chosen by a simulator, any efficient adversary cannot distinguish a random key from a decapsulated key. This definition imitates the smoothness of the hash proof system [CS02]. Those notions help us to prove the pseudorandomness of the hybrid PKE.

*Pseudorandomness, smoothness, and collision-freeness of the FO variants:* In order to treat the case for Classic McEliece and NTRU, in which the underlying PKE is deterministic, we treat SXY [SXY18], variants of U [HHK17], and variants of Hu [JZM19]. Modifying the IND-CCA security proofs of them, we show that the obtained KEM is strongly pseudorandom and smooth if the underlying PKE is strongly disjoint-simulatable [SXY18]. We also show that the obtained KEM is collision-free if the underlying deterministic PKE is collision-free. We finally note that our reductions are tight as a bonus.

Grubbs et al. [GMP21a] discussed a barrier to show anonymity of NTRU (and Classic McEliece implicitly), which stems from the design choice \(K = H(\mu)\) instead of \(K = H(\mu, c)\). In addition, their proof technique requires the underlying PKE to be collision-free. Since the underlying PKE of Classic McEliece lacks collision freeness, they left the proof of anonymity of Classic McEliece as an open problem. Both barriers stem from the fact that we need to simulate two decapsulation oracles in the proof of ANON-CCA-security. We avoid those technical barriers by using a stronger notion, SPR-CCA-security; in the proof of SPR-CCA-security, we only need to simulate a single decapsulation oracle.

*Application to NIST PQC Round 3 KEM candidates:* Using the above techniques, we solve open problems posed by Grubbs et al. and extend the study of finalists and alternative candidates of NIST PQC Round 3 KEMs as depicted in Table 1.

We found the following properties (we omit the detail of the assumptions):
- Classic McEliece is anonymous and the hybrid PKE using it is anonymous, which is in the full version.
- NTRU is anonymous and collision-free. The hybrid PKE using it is anonymous and robust. See Section 5.
- Similar results for BIKE, HQC (HQC-128 and HQC-196)\(^4\), NTRU LPrime, and SIKE hold, which are in the full version.
- We found that Streamlined NTRU Prime has another technical obstacle for anonymity: the key and key-confirmation hash involves ‘pre-key’ problem.\(^5\) While this is not a big problem for the IND-CCA security in the ROM, we fail to show the IND-CCA security in the QROM. We will discuss it in detail in the full version.

**Remark 1.1.** Bernstein [Ber21] suggests to use quantum indifferentiability of the domain extension of quantum random oracles in [Zha19, Section 5]. While we did not check the detail, this quantum indifferentiability would solve the problems on ‘pre-key’ of Kyber, Saber, and Streamlined NTRU Prime.

**Open problems:** We leave showing anonymity and the IND-CCA security of Kyber, Saber, and Streamlined NTRU Prime in the QROM as an important open problem as Grubbs et al. posed.

**Organization:** Section 2 reviews the QROM, definitions of primitives, and the results of Grubbs et al. [GMP21a]. In addition, it also shows strong pseudorandomness implies anonymity. Section 3 studies the strong pseudorandomness of the KEM/DEM framework. Section 4 studies SXY’s security properties. Section 5 examines the anonymity and robustness of NTRU. Due to the space limit, we omit a lot of contents from the conference version.

---

\(^3\) If the simulator can depend on an encryption key, then we just say pseudorandom.

\(^4\) HQC-256 is not anonymous because the parity of the ciphertext leaks the parity of the encapsulation key. See the full version for the detail.

\(^5\) The key and key-confirmation value on a plaintext \(\mu\) and an encapsulation key \(ek\) is computed as \(K = H(k, c_0, c_1)\) and \(h = F(k, \text{Hash}(ek))\), where \(k = H_3(\mu)\) and \((c_0, c_1)\) is a main body of a ciphertext.
Appendix highlights: The appendices contain the properties of the variants of the FO transform, those for T in Section D, those for a variant of U in Section E, and those for variants of HU in Section F, Section G, Section H, Section I, and Section J. The appendices examine the other NIST PQC Round 3 KEM candidates, Classic McEliece in Section K, Kyber in Section L, Saber in Section M, BIKE in Section N, FrodoKEM in Section O, HQC in Section P, NTRU Prime (Streamlined NTRU Prime in Section Q and NTRU LPRime in Section R), and SIKE in Section S, as summarized in Table 1.

2 Preliminaries

Notations: A security parameter is denoted by \( \kappa \). We use the standard \( O \)-notations. DPT, PPT, and QPT stand for a 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 \( \negl \). 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 \( \inp \) is a string, then \( \text{"out} = A(\inp)\text{"} \) denotes the output of algorithm \( A \) when run on input \( \inp \). If \( A \) is deterministic, then \( \text{out} \) is a fixed value and we write \( \text{"out} = A(\inp)\text{"} \). We also use the notation \( \text{"out} = A(\inp; r)\text{"} \) to make the randomness \( r \) explicit.

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

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

**Lemma 2.1** (Generic distinguishing problem with bounded probabilities [HKSU20, Lemma 2.9], adapted). Let \( X \) be a finite set. Let \( F: X \to \{0,1\} \) be the following function: for each \( x \in X \), \( F(x) = 1 \) with probability \( \delta_x \leq \delta \) and \( F(x) = 0 \) else. Let \( Z: X \to \{0,1\} \) be the zero function, that is, \( Z(x) = 0 \) for all \( x \). If an unbounded-time quantum adversary \( \mathcal{A} \) makes a query to \( F \) or \( Z \) at most \( Q \) times, then we have
\[
\Pr[\exists b \leftarrow \mathcal{A}^F(\cdot) : b = 1] - \Pr[\exists b \leftarrow \mathcal{A}^Z(\cdot) : b = 1] \leq 8(Q+1)^2\delta.
\]
where all oracle accesses of \( \mathcal{A} \) can be quantum.

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

We review some useful lemmas for the properties of the quantum random oracle (QRO). The first one states that QRO is PRF. See [SXY18] and [JZC18] for the proof.

**Lemma 2.2** (QRO is PRF). Let \( \ell \) be a positive integer. Let \( X \) and \( Y \) be finite sets. Let \( H_{\text{prf}}: \{0,1\}^\ell \times X \to Y \) and \( H_y: X \to Y \) be two independent random oracles. If an unbounded-time quantum adversary \( \mathcal{A} \) makes a query to the random oracles at most \( Q \) times, then we have
\[
\Pr[x \leftarrow \mathcal{M}, b \leftarrow \mathcal{A}^{H_{\text{prf}}(\cdot,H_y(\cdot))}(\cdot) : b = 1] - \Pr[b \leftarrow \mathcal{A}^{H_{\text{prf}(\cdot,H_y(\cdot))}(\cdot)}(\cdot) : b = 1] \leq 2Q \cdot 2^{-\ell/2},
\]
where all oracle accesses of \( \mathcal{A} \) can be quantum.

The second one states that QRO is collision-resistant.

**Lemma 2.3** (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
\[
\Pr_{H,\mathcal{A}}[\exists (x,x') \leftarrow \mathcal{A}^{H(\cdot)}(\cdot) : x \neq x' \land H(x) = H(x') \leq C(Q+1)^3/|Y|,
\]
where all oracle accesses of \( \mathcal{A} \) can be quantum.

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

The third one states that two QROs are claw-free.

**Lemma 2.4** (QROs are claw-free). There is a universal constant \( C \) such that the following holds: Let \( X_0 \) and \( X_1 \) 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
\[
\Pr[(x_0,x_1) \leftarrow \mathcal{A}^{H_0(\cdot),H_1(\cdot)}(\cdot) : H_0(x_0) = H_1(x_1)] \leq C(Q_0 + Q_1 + 1)^3/|Y|,
\]
where all oracle accesses of \( \mathcal{A} \) can be quantum.
The following proof is due to Hosoyamada [Hos21]:

**Proof.** Let us reduce the problem to the collision-finding problem as follows: We assume that \(X_0\) and \(X_1\) are efficiently enumerable. Given \(H: [N_0 + N_1] \rightarrow Y\), we define \(H_0: X_0 \rightarrow Y\) and \(H_1: X_1 \rightarrow Y\) by \(H_0(x) = H(\text{index}_0(x))\) and \(H_1(x) = H(\text{index}_1(x) + N_0)\), where \(\text{index}_i: X_i \rightarrow [N_i]\) is an index function which returns the index of \(x\) in \(X_i\). \(H_0\) 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 \((\text{index}_0(x_0), \text{index}_1(x_1) + N_0)\) for \(H\) with \(Q_0 + Q_1\) queries. Using Lemma 2.3, we obtain the bound as we wanted. 

\[\square\]

### 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).

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

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

**Definition 2.2 (\(\delta\)-correctness [HHK17]).** Let \(\delta = \delta(\kappa)\). We say \(\text{PKE}\) is \(\delta\)-correct if

\[
\max_{\mu \in M} \Pr[c \leftarrow \text{Enc}(ek, \mu) : \text{Dec}(dk, c) \neq \mu] \leq \delta.
\]

In particular, we say that \(\text{PKE}\) is perfectly correct if \(\delta = 0\).

We also define a key pair’s accuracy.

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

\[
\Pr_{(ek, dk) \leftarrow \text{Gen}(1^\kappa)}[\text{Dec}(dk, c) = \mu] = 1.
\]

If a key pair is not accurate, then we call it inaccurate. We note that if \(\text{PKE}\) is deterministic and \(\delta\)-correct, then

\[
\Pr_{(ek, dk) \leftarrow \text{Gen}(1^\kappa)}[(ek, dk) \text{ is inaccurate}] \leq \delta.
\]

**Security notions:** We review one-wayness under chosen-plaintext attacks (OW-CPA), one-wayness under chosen-ciphertext attacks (OW-CCA), indistinguishability under chosen-plaintext attacks (IND-CPA), indistinguishability under chosen-ciphertext attacks (IND-CCA) [RS92, BDPR98]. We define pseudorandomness under chosen-ciphertext attacks (PR-CCA) and its strong version (SPR-CCA) with simulator \(S\) as a generalization of IND\(_S\)-CCA-security in [vH04, Hop05]. We also review anonymity (ANON-CCA) [BBDP01], collision-freeness (WCRF-CCA and SCFR-CCA) [Moh10], and robustness (WROB-CCA and SROB-CCA) [Moh10]. We additionally define extended collision-freeness (XCRF), in which any efficient adversary cannot find a colliding ciphertext even if the adversary is given two decryption keys.

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

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

\[
\text{Adv}_{\text{PKE},[\mathcal{S}],\mathcal{A}}^{\text{goal-atk}}(\kappa) := 2 \Pr[\text{Exp}_{\text{PKE},[\mathcal{S}],\mathcal{A}}^{\text{goal-atk}}(\kappa) = 1] - 1,
\]

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

For any \(\mathcal{A}\) and goal-atk \(\in \{\text{ow-cca, wcrf-cca, scfr-cca, xcrf, wrobg-cca, srobg-cca}\}\), we define its goal-atk advantage against \(\text{PKE}\) as follows:

\[
\text{Adv}_{\text{PKE},[\mathcal{D}_M],\mathcal{A}}^{\text{goal-atk}}(\kappa) := 2 \Pr[\text{Exp}_{\text{PKE},[\mathcal{D}_M],\mathcal{A}}^{\text{goal-atk}}(\kappa) = 1],
\]

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

For \(\text{GOAL-ATK} \in \{\text{IND-CCA, PR-CCA, ANON-CCA, OW-CCA, WCRF-CCA, SCFR-CCA, XCRF, WROB-CCA, SROB-CCA}\}\), we say that \(\text{PKE}\) is \(\text{GOAL-ATK}-\text{secure}\) if \(\text{Adv}_{\text{PKE},[\mathcal{D}_M],\mathcal{A}}^{\text{goal-atk}}(\kappa)\) is negligible for any QPT adversary \(\mathcal{A}\). We also say that \(\text{PKE}\) is \(\text{SPR-CCA}-\text{secure}\) if it is \(\text{PR-CCA}-\text{secure}\), and its simulator ignores \(ek\). We also say that \(\text{PKE}\) is GOAL-CPA-secure if it is GOAL-CCA-secure even without the decryption oracle.
Fig. 1. Games for PKE schemes
Disjoint simulatability: We review disjoint simulatability defined in [SXY18].

**Definition 2.5 (Disjoint simulatability [SXY18]).** Let \( \mathcal{D}_M \) 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 \( \mathcal{D}_M \)-disjoint-simulatable if there exists a PPT algorithm \( S \) that satisfies the following:

- (Statistical disjointness:
  \[
  \operatorname{Disj}_{\text{PKE},S}(\kappa) \coloneqq \max_{(ek,dk) \in \operatorname{Gen}(\kappa)} \Pr[ (\cdot,ek) \leftarrow S(1^\kappa, ek) : c \in \operatorname{Enc}(ek, M) ]
  \]

  is negligible.

- (Ciphertext-indistinguishability) For any QPT adversary \( \mathcal{A} \), its ds-ind advantage \( \operatorname{Adv}_{\text{PKE},\mathcal{D}_M,S,\mathcal{A}}^{\text{ds-ind}}(\kappa) \) is negligible: The advantage is defined as
  \[
  \operatorname{Adv}_{\text{PKE},\mathcal{D}_M,S,\mathcal{A}}^{\text{ds-ind}}(\kappa) \coloneqq \left| 2 \Pr[ \operatorname{Expt}_{\text{PKE},\mathcal{D}_M,S,\mathcal{A}}^{\text{ds-ind}}(\kappa) = 1 ] - 1 \right|,
  \]

  where \( \operatorname{Expt}_{\text{PKE},\mathcal{D}_M,S,\mathcal{A}}^{\text{ds-ind}}(\kappa) \) is an experiment described in Figure 1 and \( S \) is a PPT simulator.

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

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

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

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

### 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 (Gen, Enc, Dec):

- \( \operatorname{Gen}(1^\kappa) \rightarrow (ek,dk) \): 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.

- \( \operatorname{Enc}(ek) \rightarrow (c,K) \): an encryption algorithm that takes as input encapsulation key \( ek \) and outputs ciphertext \( c \in C \) and key \( K \in \mathcal{K} \).

- \( \operatorname{Dec}(dk,c) \rightarrow K \mid \bot \): 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 (\( \delta \)-correctness).** Let \( \delta = \delta(\kappa) \). We say that KEM = (Gen, Enc, Dec) is \( \delta \)-correct if

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

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

**Security notions:** We review indistinguishability under chosen plaintext attacks (IND-CPA) and indistinguishability under chosen-ciphertext attacks (IND-CCA) [RS92, BDPR98]. We define pseudorandomness under chosen-ciphertext attacks (PR-CCA) with simulator \( S \) as a generalization of IND\&\&-CCA-security in [vH04, Hop05] and its strong version (SPR-CCA). We also review anonymity (ANON-CCA), collision-freeness (WCFR-CCA and SCFR-CCA), and robustness (WROB-CCA and SROB-CCA) [GMP21a]. We also define smoothness under chosen-ciphertext attacks (denoted by SMT-CCA) by following smoothness of hash proof system [CS02]:

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

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

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

\[
\text{Adv}_{\text{KEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) := \mathbb{P}[\text{Exp}_{\text{KEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) = 1],
\]

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

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

We additionally define \( \epsilon \)-sparseness.

**Definition 2.10 \((\epsilon \text{-sparseness})\).** Let \( S \) be a simulator for the PR-CCA security. We say that KEM is \( \epsilon \)-sparse if

\[
\mathbb{P}[\langle ek, dk \rangle \leftarrow \text{Gen}(1^\kappa), c \leftarrow S(1^\kappa, ek) : \text{Dec}(dk, c) \neq \bot] \leq \epsilon.
\]

### 2.3 Data Encapsulation Mechanism (DEM)

The model for DEM schemes is summarized as follows:

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

- \(E(K, \mu) \rightarrow d\): an encapsulation algorithm that takes as input key \(K\) and data \(\mu\) and outputs ciphertext \(d\).
- \(D(K, d) \rightarrow m|\bot\): a decapsulation algorithm that takes as input key \(K\) and ciphertext \(d\) and outputs data \(\mu\) or a rejection symbol \(\bot \notin M\).

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

\[
\mathbb{P}[D(K, d) = \mu : d \leftarrow E(K, \mu)] = 1.
\]

**Security notions:** We review indistinguishability under chosen-ciphertext attacks (IND-CCA), pseudorandomness under one-time chosen-ciphertext attacks (PR-CCA) and pseudorandomness under one-time chosen-ciphertext attacks (PR-otCCA). We also review integrity of ciphertext (INT-CTXT). Robustness of DEM (FROB and XROB) are taken from Farshim, Orlandi, and Roși [FOR17].

**Definition 2.13 \((\text{Security notions for DEM})\).** Let DEM = \((E, D)\) be a DEM scheme whose key space is \(K\). For \(\mu \in M\), let \(C|\mu|\) be a ciphertext space defined by the length of message \(\mu\).

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

\[
\text{Adv}_{\text{DEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) := 2\mathbb{P}[\text{Exp}_{\text{DEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) = 1] - 1,
\]

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

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

\[
\text{Adv}_{\text{DEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) := \mathbb{P}[\text{Exp}_{\text{DEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa) = 1],
\]

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

For \(\text{GOAL-ATK} \in \{ \text{IND-CCA}, \text{PR-CCA}, \text{PR-otCCA}, \text{INT-CTXT}, \text{FROB}, \text{XROB} \}\), we say that DEM is \(\text{GOAL-ATK} \)-secure if \(\text{Adv}_{\text{DEM}, \mathcal{A}}^{\text{goal-atk}}(\kappa)\) is negligible for any QPT adversary \(\mathcal{A}\).
### Games for KEM schemes

#### Function Definitions:
- $\text{Expt}_{KEM, \mathcal{A}}^\text{blind-cca}(\kappa)$
  - $b \leftarrow \{0, 1\}$
  - $(ek, dk) \leftarrow \text{Gen}(1^\kappa)$
  - $(c^*, K_b^*) \leftarrow \text{Enc}(ek)$
  - $K_i^* \leftarrow \mathcal{K}$
  - $b' \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek, c^*, K_b^*)$
  - return boole($b = b'$)

#### Decryption Functions:
- $\text{Dec}_a(c)$
  - if $c = a$ then return ⊥
  - $K := \text{Dec}(dk, c)$
  - return $K$

- $\text{Dec}_b'(id, c)$
  - if $c = a$ then return ⊥
  - $K := \text{Dec}(dk_{id}, c)$
  - return $K$

#### Example Experiments:
- $\text{Expt}_{KEM,S, \mathcal{A}}^\text{ind-cca}(\kappa)$
  - $b \leftarrow \{0, 1\}$
  - $(ek, dk) \leftarrow \text{Gen}(1^\kappa)$
  - $(c^*, K_b^*) \leftarrow \text{Enc}(ek)$
  - $(c^*, K_b^*) \leftarrow S(1^\kappa, ek) \times \mathcal{K}$
  - $b' \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek, c^*, K_b^*)$
  - return boole($b = b'$)

- $\text{Expt}_{KEM,S, \mathcal{A}}^\text{anon-cca}(\kappa)$
  - $(ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa)$
  - $(ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa)$
  - $b \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek_0, ek_1)$
  - $(c, K_b) \leftarrow \text{Dec}(ek_b)$
  - $K_1 \leftarrow \text{Dec}(dk_{id}, c)$
  - return boole($K_0 = K_1 \neq ⊥$)

- $\text{Expt}_{KEM, \mathcal{A}}^\text{rob-cca}(\kappa)$
  - $(ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa)$
  - $(ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa)$
  - $b \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek_0, ek_1)$
  - $(c, K_b) \leftarrow \text{Dec}(ek_b)$
  - $K_{1-b} \leftarrow \text{Dec}(dk_{1-b}, c)$
  - return boole($K_{1-b} \neq ⊥$)

- $\text{Expt}_{KEM, \mathcal{A}}^\text{scfr-cca}(\kappa)$
  - $(ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa)$
  - $(ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa)$
  - $b \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek_0, ek_1)$
  - $(c, K_b) \leftarrow \text{Dec}(ek_b)$
  - $K_{1-b} \leftarrow \text{Dec}(dk_{1-b}, c)$
  - return boole($K_{1-b} \neq ⊥$)

- $\text{Expt}_{KEM, \mathcal{A}}^\text{smt-cca}(\kappa)$
  - $(ek_0, dk_0) \leftarrow \text{Gen}(1^\kappa)$
  - $(ek_1, dk_1) \leftarrow \text{Gen}(1^\kappa)$
  - $(c^*, K^*) \leftarrow \text{Enc}(ek_b)$
  - $b' \leftarrow \mathcal{A}^{\text{Dec}^{-1}(-)}(ek_0, ek_1, c^*, K^*)$
  - return boole($b = b'$)
\begin{itemize}
\item \textbf{Expt}_{\text{ind-cca}}^{\text{DEM}, \mathcal{A}}(\kappa) \\
\quad b \leftarrow \{0, 1\} \\
\quad K \leftarrow \mathcal{K} \\
\quad (\mu_0, \mu_1, \text{state}) \leftarrow \mathcal{A}_{\text{Enc}(.), \text{Dec}_{\cdot}(.)}(1^\kappa) \\
\quad d^* \leftarrow \mathcal{E}(K, \mu_b) \\
\quad b' \leftarrow \mathcal{A}_{\text{Enc}(.), \text{Dec}_{\cdot}(.)}(d^*, \text{state}) \\
\quad b_l \leftarrow \text{boole}(|\mu_0| = |\mu_1|) \\
\quad \text{return boole}(b = b' \land b_l)
\end{itemize}

\begin{itemize}
\item \textbf{Enc}(\mu) \\
\quad d \leftarrow \mathcal{E}(K, \mu) \\
\quad \text{return } d
\end{itemize}

\begin{itemize}
\item \textbf{Dec}_{\cdot}(d) \\
\quad \text{if } d = a \\
\quad \quad \text{then return } \bot \\
\quad \mu \leftarrow \mathcal{D}(K, d) \\
\quad \text{return } \mu
\end{itemize}

\begin{itemize}
\item \textbf{Expt}_{\text{DEM}, \mathcal{A}}^{\text{ctx}}(\kappa) \\
\quad K \leftarrow \mathcal{K} \\
\quad w \leftarrow \bot \\
\quad L \leftarrow \emptyset \\
\quad \mathcal{A}_{\text{Enc}2(.), \text{Dec}_{\cdot}2(.)}(1^\kappa) \\
\quad \text{return } w
\end{itemize}

\begin{itemize}
\item \textbf{Enc}_{\cdot}^{\text{ctx}}(\mathcal{A})(\mu) \\
\quad d \leftarrow \mathcal{E}(K, \mu) \\
\quad L \leftarrow L \cup \{d\} \\
\quad \text{return } d
\end{itemize}

\begin{itemize}
\item \textbf{Dec}_{\cdot}(d) \\
\quad \mu \leftarrow \mathcal{D}(K, d) \\
\quad \text{if } \mu \neq \bot \land d \notin L \text{ then set } w = \top \\
\quad \text{return } \mu
\end{itemize}

\begin{itemize}
\item \textbf{Expt}_{\text{pr-cca}}^{\text{DEM}, \mathcal{A}}(\kappa) \\
\quad b \leftarrow \{0, 1\} \\
\quad K \leftarrow \mathcal{K} \\
\quad (\mu_0, \mu_1, \text{state}) \leftarrow \mathcal{A}_{\text{Enc}(.), \text{Dec}_{\cdot}(.)}(1^\kappa) \\
\quad d^* \leftarrow \mathcal{E}(K, \mu_b) \\
\quad d^*_0 \leftarrow \mathcal{U}(\mathcal{C}_{\mu}) \\
\quad b' \leftarrow \mathcal{A}_{\text{Enc}(.), \text{Dec}_{\cdot}(.)}(d^*_0, \text{state}) \\
\quad \text{return boole}(b = b')
\end{itemize}

\begin{itemize}
\item \textbf{Enc}_{\cdot}^{\text{pr-otcca}}(\mathcal{A})(\mu) \\
\quad d \leftarrow \mathcal{E}(K, \mu) \\
\quad L \leftarrow L \cup \{d\} \\
\quad \text{return } d
\end{itemize}

\begin{itemize}
\item \textbf{Dec}_{\cdot}(d) \\
\quad \mu \leftarrow \mathcal{D}(K, d) \\
\quad \text{if } \mu \neq \bot \land d \notin L \text{ then set } w = \top \\
\quad \text{return } \mu
\end{itemize}

\begin{itemize}
\item \textbf{Expt}_{\text{frob}}^{\text{DEM}, \mathcal{A}}(\kappa) \\
\quad (d, k_0, k_1) \leftarrow \mathcal{A}(1^\kappa) \\
\quad \mu_0 \leftarrow \mathcal{D}(k_0, d) \\
\quad \mu_1 \leftarrow \mathcal{D}(k_1, d) \\
\quad b \leftarrow \text{boole}(\mu_0 \neq \bot \land \mu_1 \neq \bot) \\
\quad b_k \leftarrow \text{boole}(k_0 \neq k_1) \\
\quad \text{return boole}(b \land b_k)
\end{itemize}

\begin{itemize}
\item \textbf{Enc}_{\cdot}^{\text{frob}}(\mathcal{A})(\mu) \\
\quad (\mu_0, k_0, R_0, k_1, d_1) \leftarrow \mathcal{A}(1^\kappa) \\
\quad d_0 \leftarrow \mathcal{E}(k_0, \mu_0, R_0) \\
\quad \mu_1 \leftarrow \mathcal{D}(k_1, d_1) \\
\quad b \leftarrow \text{boole}(\mu_0 \neq \bot \land \mu_1 \neq \bot) \\
\quad b_k \leftarrow \text{boole}(k_0 \neq k_1) \\
\quad b_c \leftarrow \text{boole}(d_0 = d_1 \neq \bot) \\
\quad \text{return boole}(b \land b_k \land b_c)
\end{itemize}

Fig. 3. Games for DEM schemes
2.4 Review of Grubbs, Maram, and Paterson [GMP21a]

Grubbs et al. studied KEM’s anonymity and hybrid PKE’s anonymity and robustness by extending the results of Mohassel [Moh10]. We use KEM and KEM to indicate KEM with explicit rejection and implicit rejection, respectively. For KEM with explicit rejection, they showed the following theorem which generalizes Mohassel’s theorem [Moh10]:

**Theorem 2.1 ([GMP21a, Theorem 1]).** Let PKE by = Hyb[KEM, DEM], a hybrid PKE scheme obtained by composing KEM and DEM. (See Figure 4.)

1. If KEM is wANON-CPA-secure, IND-CCA-secure, WROB-CCA-secure, and δ-correct and DEM is INT-CTX-secure, then PKE by is ANON-CCA-secure.
2. If KEM is SROB-CCA-secure (and WROB-CCA-secure), then PKE by is SROB-CCA-secure (and WROB-CCA-secure), respectively.

Grubbs et al. [GMP21a] then treat KEM with implicit rejection, which is used in all NIST PQC Round 3 KEM candidates except HQC. Their results are related to the FO transform with implicit rejection, which is decomposed into two transforms, T and U. T transforms a probabilistic PKE scheme PKE into a deterministic PKE scheme PKE with a random oracle G; U transforms a deterministic PKE scheme PKE into a probabilistic KEM with a random oracle H. Roughly speaking, they showed the following two theorems on robustness and anonymity of hybrid PKE from KEM with explicit rejection:

**Theorem 2.2 (Robustness of PKE by [GMP21a, Theorem 2]).** Let PKE by = Hyb[KEM, DEM]. If KEM is SCFR-CCA-secure (and WCFR-CCA-secure) and DEM is SROB-secure (and XROB-secure), then PKE by is SROB-CCA-secure (and WROB-CCA-secure), respectively.

**Theorem 2.3 (Anonymity of PKE by using FO [GMP21a, Theorem 7]).** Let PKE by = Hyb[KEM, DEM]. If PKE is δ-correct, and γ-spread, PKE = T [PKE, G] is WCFR-CCA-secure, KEM is FO-secure [PKE, G, H] is ANON-CCA-secure and IND-CCA-secure, DEM is INT-CTX-secure, then PKE by is ANON-CCA-secure.

They also showed that the following theorem:

**Theorem 2.4 (Anonymity of KEM using FO [GMP21a, Theorem 5]).** If PKE is wANON-CPA-secure, OW-CPA-secure, and δ-correct, and PKE = T [PKE, G] is SCFR-CPA-secure, then a KEM scheme KEM = FO-secure [PKE, G, H] is ANON-CCA-secure.

Grubbs et al. reduced from the wANON-CPA-security of PKE to the ANON-CCA-security of KEM. We note that there are two decapsulation oracles in the security game of the ANON-CCA-security of KEM. Thus, they need to simulate both decapsulation oracles without secrets. Jiang et al. [JZC+18] used the simulation trick that replaces H(μ, c) with Hq(Enc(ek, μ)) if c = Enc(ek, μ) and Hq(μ, c) else, which helps the simulation of the decapsulation oracle without secrets in the QROM. Grubbs et al. extended this trick to simulate two decapsulation oracles by replacing H(μ, c) with Hq(Enc(ek, μ)) if c = Enc(ek, μ) and Hq(μ, c) else. Notice that this extended simulation heavily depends on the fact that H takes μ and c and the SCFR-CCA-security of PKE. If the random oracle takes μ only, their trick fails the simulation.

2.5 Strong Pseudorandomness Implies Anonymity

We observe that strong pseudorandomness of PKE/KEM immediately implies anonymity of PKE/KEM, which may be folklore. We give the proof for PKE for completeness.

**Theorem 2.5.** If PKE or KEM is SPR-CCA-secure, then it is ANON-CCA-secure.

**Proof.** Here we only consider the case for PKE, since the proof for the case for KEM is obtained by the similar way. Let us define four games Gamei,b for i, b ∈ {0, 1}:

- Game0,b for b ∈ {0, 1}: This is the original game Exp[^anonym-cca=](k) with b = 0 and 1.
- Game1,b for b ∈ {0, 1}: This is the same as Game0,b except that the target ciphertext is randomly taken from S(1^k) × U(CDEM, |μ|).

Let S0,b be the event that the adversary outputs 1 in Gamei,b. It is easy to see that there exist two adversaries A10 and A11 whose running times are the same as that of A satisfying

\[ |\Pr[S0,b] - \Pr[S1,b]| \leq \text{Adv}^{\text{spr-cca}}_{\text{PKE}, S, A10}(k). \]

In addition, we have

\[ \Pr[S1,0] = \Pr[S1,1] \]

since the distribution of the target ciphertext in both game is S(1^k) × U(CDEM, |μ|). Hence, we have

\[ \text{Adv}^{\text{anonym-cca}}_{\text{PKE}, A}(k) = |\Pr[S0,0] - \Pr[S0,1]| \leq |\Pr[S0,0] - \Pr[S1,0]| + |\Pr[S1,0] - \Pr[S1,1]| + |\Pr[S1,1] - \Pr[S0,1]| \leq \text{Adv}^{\text{spr-cca}}_{\text{PKE}, S, A10}(k) + \text{Adv}^{\text{spr-cca}}_{\text{PKE}, S, A11}(k). \]

This completes the proof. □
3 Strong Pseudorandomness of Hybrid PKE

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

<table>
<thead>
<tr>
<th>( \text{Gen}_{\text{hy}}(1^k) )</th>
<th>( \text{Enc}_{\text{hy}}(ek, \mu) )</th>
<th>( \text{Dec}_{\text{hy}}(dk, ct = (c, d)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ek, dk) ( \leftarrow \text{Gen}(1^k) )</td>
<td>(c, K) ( \leftarrow \text{Enc}(ek) )</td>
<td>( K' \leftarrow \text{Dec}(dk, c) )</td>
</tr>
<tr>
<td>return (ek, dk)</td>
<td>( d \leftarrow E(K, \mu) )</td>
<td>if ( K' = \perp ) then return ( \perp )</td>
</tr>
<tr>
<td></td>
<td>return ( ct := (c, d) )</td>
<td>( \mu' \leftarrow D(K', d) )</td>
</tr>
</tbody>
</table>

Fig. 4. \( \text{PKE}_{\text{hy}} = \text{Hyb}[\text{KEM}, \text{DEM}] \)

We show the following two theorems on strong pseudorandomness and anonymity of a hybrid PKE:

**Theorem 3.1 (Case for KEM with explicit rejection).** Let \( \text{PKE}_{\text{hy}} = (\text{Gen}_{\text{hy}}, \text{Enc}_{\text{hy}}, \text{Dec}_{\text{hy}}) \) be a hybrid encryption scheme obtained by composing a KEM scheme \( \text{KEM}^\perp = (\text{Gen}, \text{Enc}, \text{Dec}) \) and a DEM scheme \( \text{DEM} = (E, D) \) that share key space \( \mathcal{K} \). If \( \text{KEM}^\perp \) is SPR-CCA-secure, \( \delta \)-correct with negligible \( \delta \), and \( \epsilon \)-sparse and \( \text{DEM} \) is PR-otCCA-secure and INT-CTXT-secure, then \( \text{PKE}_{\text{hy}} \) is SPR-CCA-secure (and ANON-CCA-secure).

Formally speaking, for any \( \mathcal{A} \) against the SPR-CCA security of \( \text{PKE}_{\text{hy}} \), there exist \( \mathcal{A}_{23} \) against the SPR-CCA security of \( \text{KEM}^\perp \), \( \mathcal{A}_{34} \) against the SPR-CCA security of \( \text{DEM} \), and \( \mathcal{A}_{45} \) against the INT-CTXT security of \( \text{DEM} \) such that

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

**Theorem 3.2 (Case for KEM with implicit rejection).** Let \( \text{PKE}_{\text{hy}} = (\text{Gen}_{\text{hy}}, \text{Enc}_{\text{hy}}, \text{Dec}_{\text{hy}}) \) be a hybrid encryption scheme obtained by composing a KEM scheme \( \text{KEM}^\perp = (\text{Gen}, \text{Enc}, \text{Dec}) \) and a DEM scheme \( \text{DEM} = (E, D) \) that share key space \( \mathcal{K} \). If \( \text{KEM}^\perp \) is SPR-CCA-secure, SSMT-CCA-secure, and \( \delta \)-correct with negligible \( \delta \) and \( \text{DEM} \) is PR-otCCA-secure, then \( \text{PKE}_{\text{hy}} \) is SPR-CCA-secure (and ANON-CCA-secure).

Formally speaking, for any \( \mathcal{A} \) against the SPR-CCA security of \( \text{PKE}_{\text{hy}} \), there exist \( \mathcal{A}_{23} \) against the SPR-CCA security of \( \text{KEM}^\perp \), \( \mathcal{A}_{34} \) against the SPR-CCA security of \( \text{DEM} \), and \( \mathcal{A}_{45} \) against the SSMT-CCA security of \( \text{KEM}^\perp \) such that

\[
\text{Adv}^{\text{spr-cca}}_{\text{PKE}_{\text{hy}}, \mathcal{S}, \mathcal{A}}(k) \leq \text{Adv}^{\text{spr-cca}}_{\text{KEM}^\perp, \mathcal{S}, \mathcal{A}_{23}}(k) + \text{Adv}^{\text{spr-otcca}}_{\text{DEM}, \mathcal{A}_{34}}(k) + \text{Adv}^{\text{ssmt-cca}}_{\text{KEM}^\perp, \mathcal{S}, \mathcal{A}_{45}}(k) + \delta.
\]

We here prove Theorem 3.2 and give the proof of Theorem 3.1 in subsection B.1.

**Proof of Theorem 3.2** Let us consider Game\(_i\) for \( i = 0, \ldots, 6 \). We summarize the games in Table 2. Let \( S_1 \) denote the event that the adversary outputs \( b' = 1 \) in Game\(_1\).

Let \( \mathcal{S} \) be the simulator for the SPR-CCA security of \( \text{KEM}^\perp \). We define \( S_{\text{hy}}(1^k, |\mu'|) := \mathcal{S}(1^k) \times U(G_{|\mu'|}) \) as the simulator for the SPR-CCA security of \( \text{PKE}_{\text{hy}} \).

The security proof is similar to the security proof of the IND-CCA security of KEM/DEM \([\text{CS03}]\) for Game\(_0\), \ldots, Game\(_4\).

We need to take care of pseudorandom ciphertexts when moving from Game\(_4\) to Game\(_5\) and require the SSMT-CCA security of \( \text{KEM}^\perp \).

**Game\(_5\):** This is the original game \( \text{Exp}_{\text{PKE}_{\text{hy}}, \mathcal{S}, \mathcal{A}}^{\text{spr-cca}}(k) \) with \( b = 0 \). Given \( \mu' \), the challenge ciphertext is computed as follows:

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

We have

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

**Game\(_1\):** In this game, \( c^* \) and \( K^* \) are generated before invoking \( \mathcal{A} \) with \( ek \). This change is just conceptual, and we have

\[
\Pr[S_0] = \Pr[S_1].
\]
Table 2. Summary of Games for the Proof of Theorem 3.2

<table>
<thead>
<tr>
<th>Game</th>
<th>$c^<em>$ and $K^</em>$</th>
<th>$d^*$</th>
<th>Decryption</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game1</td>
<td>Enc$(ek)$</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game2</td>
<td>Enc$(ek)$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game3</td>
<td>Enc$(ek)$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game4</td>
<td>$S(1^k) \times U(K)$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game5</td>
<td>$S(1^k) \times U(K)$ at first</td>
<td>$U(C_{\mu^<em>}^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game6</td>
<td>$S(1^k) \times U(K)$ at first</td>
<td>$U(C_{\mu^<em>}^</em>)$</td>
<td>$d^*$</td>
<td>conceptual change</td>
</tr>
</tbody>
</table>

**Game1:** In this game, the decryption oracle uses $K^*$ if $c = c^*$ instead of $K = \overline{\text{Dec}}(dk, c^*)$. Game1 and Game2 differ if correctly generated ciphertext $c^*$ with $K^*$ is decapsulated into different $K \neq K^*$ or ⊥, which violates the correctness and occurs with probability at most $\delta$. Hence, the difference of Game1 and Game2 is bounded by $\delta$, and we have

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

We note that this corresponds to the event BadKeyPair in [CS03].

**Game3:** In this game, the challenger uses random $(c^*, K^*)$ and uses $K^*$ in DEM. The challenge ciphertext is generated as follows:

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

The difference is bounded by the SPR-CCA security of KEM$^\perp$.

**Game4:** In this game, the challenger uses random $d^*$. The challenge ciphertext is generated as follows:

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

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

$$|\Pr[S_1] - \Pr[S_3]| \leq \text{Adv}_{\text{DEM, } A_{34}}^{\text{spr-otcca}}(\kappa).$$

We omit the detail of $A_{34}$ since it is straightforward.

**Game5:** In this game, we replace the decryption oracle defined as follows: If given $ct = (c^*, d)$, the decryption oracle uses $K = \overline{\text{Dec}}(dk, c^*)$ instead of $K^*$.

The difference is bounded by the SSMT-CCA security of KEM$^\perp$.

We omit the detail of $A_{45}$ since it is straightforward.

**Game6:** We finally change the timing of the generation of $(c^*, K^*)$. This change is just conceptual, and we have

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

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

$$\Pr[S_6] = \Pr[\text{Exp}_{\text{PKE}_{\mu^*}, S, A}^{\text{spr-cca}}(\kappa) = 1 | b = 1].$$

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

$$\text{Adv}_{\text{PKE}_{\mu^*}, S, A}^{\text{spr-cca}}(\kappa) = |\Pr[S_6] - \Pr[S_3]| \leq \sum_{i=0}^5 |\Pr[S_i] - \Pr[S_{i+1}]|$$

$$\leq \text{Adv}_{\text{KEM}^\perp, S, A_{34}}^{\text{spr-otcca}}(\kappa) + \text{Adv}_{\text{DEM, } A_{34}}^{\text{spr-otcca}}(\kappa) + \text{Adv}_{\text{KEM}^\perp, S, A_{45}}^{\text{ssmt-cca}}(\kappa) + \delta.$$  

$\square$
4 Properties of SXY

Let us review SXY [SXY18] as known as $U_m^\ell$ with explicit re-encryption check [HHHK17]. Let $PKE = (Gen, Enc, Dec)$ be a deterministic PKE scheme. Let $M, C,$ and $K$ be a plaintext, ciphertext, and key space of $PKE$, respectively. Let $H: M \rightarrow K$ and $H_{prf}: \{0, 1\}^\ell \times C \rightarrow K$ be hash functions modeled by random oracles. $KEM = (Gen, Enc, Dec) = SXY[PKE, H, H_{prf}]$ is defined as in Figure 5.

<table>
<thead>
<tr>
<th>Gen($1^k$)</th>
<th>Enc($ek$)</th>
<th>Dec$(dk, c)$, where $dk = (dk, ek, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: $(ek, dk) \leftarrow Gen(1^k)$</td>
<td>1: $\mu \leftarrow D_M$</td>
<td>1: $\mu' \leftarrow Dec(dk, c)$</td>
</tr>
<tr>
<td>2: $s \leftarrow {0, 1}^\ell$</td>
<td>2: $c \leftarrow Enc(ek, \mu)$</td>
<td>2: if $\mu' = \perp$ or $c \neq Enc(ek, \mu')$</td>
</tr>
<tr>
<td>3: $dk \leftarrow (dk, ek, s)$</td>
<td>3: $K \leftarrow H(\mu)$</td>
<td>3: then return $K \leftarrow H_{prf}(s, c)$</td>
</tr>
<tr>
<td>4: return $(ek, dk)$</td>
<td>4: return $(c, K)$</td>
<td>4: else return $K \leftarrow H(\mu')$</td>
</tr>
</tbody>
</table>

Fig. 5. KEM = SXY[PKE, H, H_{prf}]

4.1 SPR-CCA Security

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

**Theorem 4.1 (Case of derandomized PKE).** Let $PKE_0$ be a probabilistic PKE. Let us consider a derandomized PKE $PKE = T[PKE_0, G]$. Suppose that a ciphertext space $C$ of $PKE$ depends on the public parameter only. If PKE is strongly disjoint-simulatable and $\delta$-correct with negligible $\delta$, then KEM = SXY[PKE, H, H_{prf}] is SPR-CCA-secure.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{Dec}$ queries to the decryption oracle and $q_{G}, q_{H}$, and $q_{H_{prf}}$ queries to G, H, and $H_{prf}$, respectively, there exists $\mathcal{A}_{S_X}$ against ciphertext-indistinguishability of PKE such that

$$
\text{Adv}_{KEM, S_X, \mathcal{A}}^{\text{SPR-CCA}}(\kappa) \leq \text{Adv}_{PKE, D_M, S_X, \mathcal{A}_{S_X}}^{\text{IND-CCA}}(\kappa) + \text{Disj}_{PKE, S_X}(\kappa) + 4\delta + 4(q_{H_{prf}} + q_{Dec}) \cdot 2^{-k/2} + 16(q_{G} + q_{Dec} + 1)^2 \delta + 16(q_{G} + q_{H} + 1)^2 \delta.
$$

?? ?? +1 to +2 ?? ??

**Theorem 4.2 (Case for non-derandomized PKE).** Suppose that a ciphertext space $C$ of $PKE$ depends on the public parameter only. If PKE is strongly disjoint-simulatable and $\delta$-correct with negligible $\delta$, then KEM = SXY[PKE, H, H_{prf}] is SPR-CCA-secure.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{Dec}$ queries to the decryption oracle and $q_{G}, q_{H}$, and $q_{H_{prf}}$ queries to G, H, and $H_{prf}$, respectively, there exists $\mathcal{A}_{S_X}$ against ciphertext-indistinguishability of PKE such that

$$
\text{Adv}_{KEM, S_X, \mathcal{A}}^{\text{SPR-CCA}}(\kappa) \leq \text{Adv}_{PKE, D_M, S_X, \mathcal{A}_{S_X}}^{\text{IND-CCA}}(\kappa) + \text{Disj}_{PKE, S_X}(\kappa) + 4\delta + 4(q_{H_{prf}} + q_{Dec}) \cdot 2^{-k/2}.
$$

For simplicity, we here prove Theorem 4.2 because it is simple and suffices for the NTRU case. We give the security proof of Theorem 4.1 in subsection B.2.

**Proof of Theorem 4.2:** We use the game-hopping proof. We consider Game; for $i = 0, \ldots, 8$. We summarize the games in Table 3. Let $S_i$ denote the event that the adversary outputs $b' = 1$ in game Game; Let Acc be an event that a key pair $(ek, dk)$ is accurate. Let $\neg$Acc denote the event that a key pair $(ek, dk)$ is inaccurate. We note that we have $\Pr[\neg\text{Acc}] \leq \delta$ since PKE is deterministic. We extend the security proof for IND-CCA security of SXY in [SXY18, XY19, LW21].

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

$$
\Pr[S_0] = 1 - \Pr[\text{Exp}_{KEM, \mathcal{A}}^{\text{SPR-CCA}}(\kappa) = 1 | b = 0].
$$
Table 3. Summary of games for the proof of Theorem 4.2

<table>
<thead>
<tr>
<th>Game</th>
<th>H</th>
<th>c*</th>
<th>K*</th>
<th>Decapsulation</th>
<th>valid c</th>
<th>invalid c</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game0</td>
<td>H(·)</td>
<td>Enc(ek, μ*)</td>
<td>H(μ*)</td>
<td>H(μ)</td>
<td>H_{prf}(s, c)</td>
<td></td>
<td>Lemma 2.2</td>
</tr>
<tr>
<td>Game1</td>
<td>H(·)</td>
<td>Enc(ek, μ*)</td>
<td>H(μ')</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game1,5</td>
<td>H_{q'}(Enc(ek, ·))</td>
<td>Enc(ek, μ*)</td>
<td>H(μ')</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game2</td>
<td>H_{q}(Enc(ek, ·))</td>
<td>Enc(ek, μ*)</td>
<td>H(μ')</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game3</td>
<td>H_{q}(Enc(ek, ·))</td>
<td>Enc(ek, μ*)</td>
<td>H(μ')</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game4</td>
<td>H(·)</td>
<td>S(1*)</td>
<td>U(K)</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game5</td>
<td>H(·)</td>
<td>S(1*)</td>
<td>U(K)</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game6</td>
<td>H(·)</td>
<td>S(1*)</td>
<td>U(K)</td>
<td>H(μ)</td>
<td>H_{q}(c)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Game1:** This game is the same as Game0 except that $H_{prf}(s, c)$ in the decapsulation oracle is replaced with $H_{q}(c)$ where $H_{q} : C \rightarrow \mathcal{K}$ is another random oracle. We remark that $\mathcal{A}$ cannot access $H_{q}$ directly. As in [XY19, Lemma 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_0] - \Pr[S_1]| \leq 2(q_{H_{prf}} + q_{Dec}) \cdot 2^{-\ell/2},$$

where $q_{H_{prf}}$ and $q_{Dec}$ denote the number of queries to $H_{prf}$ and $Dec$ the adversary makes, respectively. In addition, according to Lemma A.1, for any $p \geq 0$, we have

$$|\Pr[S_1] - p| \leq |\Pr[S_1 \land Acc] - p| + \delta.$$

**Game1,5:** This game is the same as Game1 except that the random oracle $H(·)$ is simulated by $H_{q'}(Enc(ek, ·))$ 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 $H_{q'}(Enc(ek, ·))$ as $H(·)$. If the key pair $(ek, dk)$ is accurate, then $g(μ) := Enc(ek, μ)$ is injective. Thus, if the key pair is accurate, then $H_{q'} \circ g : M \rightarrow \mathcal{K}$ is a random function and the two games Game1 and Game1,5 are equal to each other. Thus, we have

$$Pr[S_1 \land Acc] = Pr[S_{1,5} \land Acc].$$

**Game2:** This game is the same as Game1,5 except that the random oracle $H$ is simulated by $H_{q} \circ g$ instead of $H_{q'} \circ g$. A ciphertext $c$ is said to be valid if we have $Enc(ek, Dec(dk, c)) = c$ and invalid otherwise. Notice that, in Game1,5, $H_{q}$ is used for invalid ciphertext, and an adversary cannot access a value of $H_{q}$ for a valid ciphertext. In addition, in Game1,5, an adversary can access a value of $H_{q}$ on input a valid ciphertext and cannot access a value of $H_{q'}$ on input an invalid ciphertext if the key pair is accurate. Thus, there is no difference between Game1,5 and Game2 if the key pair is accurate and we have

$$Pr[S_{1,5} \land Acc] = Pr[S_2 \land Acc].$$

**Game3:** This game is the same as Game2 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 is denoted by $Dec'$. If the key pair is accurate, for a valid ciphertext $c$ and its decrypted result $μ$, we have $H(μ) = H_{q}(Enc(ek, μ)) = H_{q}(c)$. Thus, the two games Game2 and Game3 are equal to each other and we have

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

According to Lemma A.1, for any $p \geq 0$, we have

$$|Pr[S_3 \land Acc] - p| \leq |Pr[S_3] - p| + \delta.$$

**Game4:** This game is the same as Game3 except that $c^*$ is generated by $S(1^*)$. The difference between two games Game3 and Game4 is bounded by the advantage of ciphertext indistinguishability in disjoint simulatability as in [XY19, Lemma 4.7]. The reduction algorithm is obtained straightforwardly, and we omit it. We have

$$|Pr[S_3] - Pr[S_4]| \leq Adv_{\text{ds-ind}}^{PKE, D_M, S, A_3}(κ).$$
Games: This game is the same as Game except that $K^* \leftarrow \mathcal{K}$ instead of $K^* \leftarrow H_q(c^*)$.

In Game, if $c^* \leftarrow S(1^\lambda)$ 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 and Game is bounded by the statistical disjointness in disjoint simulatability as in [XY19, Lemma 4.8]. We have

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

According to Lemma A.1, for any $p \geq 0$, we have

$$|\Pr[S_5] - p| \leq |\Pr[S_5 \land \text{Acc}] - p| + \delta.$$

Game: This game is the same as Game except that the decapsulation oracle is reset as Dec. Similar to the case for Game and Game, if a key pair is accurate, the two games Game5 and Game6 are equal to each other as in the proof of [XY19, Lemma 4.5]. We have

$$\Pr[S_6 \land \text{Acc}] = \Pr[S_6 \land \text{Acc}].$$

Games: This game is the same as Game6 except that the random oracle $H$ is simulated by $H'_q \circ g$ where $H'_q : \{0,1\}^\lambda \rightarrow \mathcal{K}$ is yet another random oracle as in Game1 rather than $H_q \circ g$. If a key pair is accurate, then two games Game6 and Game6 are equal to each other as the two games Game1 and Game2 are equal to each other. We have

$$\Pr[S_6 \land \text{Acc}] = \Pr[S_6 \land \text{Acc}].$$

Game: This game is the same as Game6 except that the random oracle $H(\cdot)$ is set as the original. If a key pair is accurate, then the two games Game6 and Game are equal to each other as the two games Game1 and Game are equal to each other. We have

$$\Pr[S_6 \land \text{Acc}] = \Pr[S_7 \land \text{Acc}].$$

According to Lemma A.1, for any $p \geq 0$, we have

$$|\Pr[S_7 \land \text{Acc}] - p| \leq |\Pr[S_7] - p| + \delta.$$

Game: This game is the same as Game except that $H_q(c)$ in the decapsulation oracle is replaced by $H_{prf}(s, c)$. As we discussed the difference between the two games Game0 and Game1, from Lemma 2.2 we have the bound

$$|\Pr[S_7] - \Pr[S_8]| \leq 2(q_{\text{prf}} + q_{\text{Dec}}) \cdot 2^{-\ell/2}.$$}

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

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

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

$$\text{Adv}_{\text{KEM}, \mathcal{A}}^{\text{spr-cca}}(\kappa) = |\Pr[S_0] - \Pr[S_8]| \leq \sum_{i=0}^{2} |\Pr[S_i] - \Pr[S_{i+1}]|$$

$$\leq \text{Adv}_{\text{PKE}, \mathcal{A} \cdot \mathcal{M}}^{\text{dis-ind}}(\kappa) + \text{Disj}_{\text{PKE}, S}(\kappa) + 4(q_{\text{prf}} + q_{\text{Dec}}) \cdot 2^{-\ell/2} + 4\delta.$$

### 4.2 SSMT-CCA Security

We next show that KEM is strongly smooth if the underlying PKE is strongly disjoint-simulatable.

**Theorem 4.3.** Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then $\text{KEM} = \text{SKY}[\text{PKE}, H, H_{prf}]$ is SSMT-CCA-secure.

Formally speaking, for any adversary $\mathcal{A}$ against SSMT-CCA security of KEM issuing at most $q_{\text{prf}}$ and $q_{\text{Dec}}$ queries to $H_{prf}$ and Dec, we have

$$\text{Adv}_{\text{KEM}, \mathcal{A}, \mathcal{S}}^{\text{ssmt-cca}}(\kappa) \leq 2\text{Disj}_{\text{PKE}, S}(\kappa) + 4(q_{\text{prf}} + q_{\text{Dec}}) \cdot 2^{-\ell/2}.$$
Table 4. Summary of games for the proof of Theorem 4.3: \( S(1^k) \setminus \text{Enc}(ek, M) \) implies that the challenger generates \( c^* \leftarrow S(1^k) \) and returns \( \bot \) if \( c^* \in \text{Enc}(ek, M) \).

<table>
<thead>
<tr>
<th>Game</th>
<th>( (c^* \mid k^*) )</th>
<th>Decapsulation</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game0</td>
<td>( S(1^k) )</td>
<td>valid ( c ) ( \quad ) invalid ( c )</td>
<td>Lemma 2.2</td>
</tr>
<tr>
<td>Game1</td>
<td>( S(1^k) \setminus \text{Enc}(ek, M) )</td>
<td>random</td>
<td>( H(\mu) ) ( H_{prf}(s, c) ) statistical disjointness</td>
</tr>
<tr>
<td>Game2</td>
<td>( S(1^k) \setminus \text{Enc}(ek, M) )</td>
<td>random</td>
<td>( H(\mu) ) ( H_{prf}(s, c) ) statistical disjointness</td>
</tr>
<tr>
<td>Game3</td>
<td>( S(1^k) \setminus \text{Enc}(ek, M) )</td>
<td>( H_q(c^*) )</td>
<td>( H(\mu) ) ( H_q(c) ) ( H_q(c^*) ) is hidden</td>
</tr>
<tr>
<td>Game4</td>
<td>( S(1^k) \setminus \text{Enc}(ek, M) )</td>
<td>( \text{Dec}(dk, c^*) )</td>
<td>( H(\mu) ) ( H_{prf}(s, c) ) re-encryption check</td>
</tr>
<tr>
<td>Game5</td>
<td>( S(1^k) )</td>
<td>( \text{Dec}(dk, c^*) )</td>
<td>( H(\mu) ) ( H_{prf}(s, c) ) statistical disjointness</td>
</tr>
</tbody>
</table>

**Proof:** We use the game-hopping proof. We consider Game\(_i\) for \( i = 0, \ldots, 6 \). We summarize those games in Table 4. Let \( S_i \) denote the event that the adversary outputs \( b' = 1 \) in game Game\(_i\).

**Game0:** This game is the original game \( \text{Exp}^{\text{ssmt-cca}}_{\text{KEM}, S, \mathcal{A}}(k) \) with \( b = 0 \). The challenge is generated as \( c^* \leftarrow S(1^k) \) and \( K_0^* \leftarrow \mathcal{K} \). We have

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

**Game1:** In this game, the challenge ciphertext is set as \( \bot \) if \( c^* \) is in \( \text{Enc}(ek, M) \). Since the difference between two games Game0 and Game1 is bounded by statistical disjointness, we have

\[
|\Pr[S_0] - \Pr[S_1]| \leq \text{Disj}_{\text{PKE}, S}(k).
\]

**Game2:** This game is the same as Game1 except that \( H_{prf}(s, c) \) in the decapsulation oracle is replace with \( H_q(c) \) where \( H_q \) is another random oracle. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

\[
|\Pr[S_1] - \Pr[S_2]| \leq 2(q_{H_{prf}} + q_{\text{Dec}}) \cdot 2^{-f/2}.
\]

**Game3:** This game is the same as Game2 except that \( k^* \) is set as \( H_q(c^*) \) instead of chosen randomly. Since \( c^* \) is always outside of \( \text{Enc}(ek, M) \), \( \mathcal{A} \) cannot obtain any information about \( H_q(c^*) \). Hence, the two games Game2 and Game3 are equal to each other and we have

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

**Game4:** This game is the same as Game3 except that \( H_q(\cdot) \) is replaced by \( H_{prf}(s, \cdot) \). As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

\[
|\Pr[S_3] - \Pr[S_4]| \leq 2(q_{H_{prf}} + q_{\text{Dec}}) \cdot 2^{-f/2}.
\]

**Game5:** This game is the same as Game4 except that \( k^* \) is set as \( \text{Dec}(dk, c^*) \) instead of \( H_{prf}(s, c^*) \). Recall that \( c^* \) is always in outside of \( \text{Enc}(ek, M) \). Thus, we always have \( \text{Dec}(c^*) = \bot \) or \( \text{Enc}(ek, \text{Dec}(c^*)) \neq c^* \) and, thus, \( k^* = H_{prf}(s, c^*) \) in Game5. Hence, the two games are equal to each other and we have

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

**Game6:** We finally replace the way to compute \( c^* \): In this game, the ciphertext is chosen by \( S(1^k) \) as in Game0. Again, since the difference between two games Game5 and Game6 is bounded by statistical disjointness, we have

\[
|\Pr[S_5] - \Pr[S_6]| \leq \text{Disj}_{\text{PKE}, S}(k).
\]

Moreover, this game Game6 is the original game \( \text{Exp}^{\text{ssmt-cca}}_{\text{KEM}, S, \mathcal{A}}(k) \) with \( b = 1 \) and we have

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

Summing those inequalities, we obtain Theorem 4.3:

\[
\text{Adv}^{\text{ssmt-cca}}_{\text{KEM}, S, \mathcal{A}}(k) = |\Pr[S_0] - \Pr[S_6]| \leq 2\text{Disj}_{\text{PKE}, S}(k) + 4(q_{H_{prf}} + q_{\text{Dec}}) \cdot 2^{-f/2}.
\]
4.3 SCFR-CCA Security

Finally, we show that KEM is strongly collision-free if the underlying PKE is strongly collision-free or extended collision-free.

**Theorem 4.4.** If PKE is SCFR-CCA-secure (or XCFR-secure), then KEM = SXY[PKE, H, H_{prf}] is SCFR-CCA-secure in the QROM.

**Proof.** Suppose that an adversary against KEM’s SCFR-CCA security outputs a ciphertext c which is decapsulated into K ≠ ⊥ by both Sk_0 and Sk_1, that is, K = Dec(Sk_0, c) = Dec(Sk_1, c) ≠ ⊥. For i ∈ {0, 1}, we define μ'_i as an internal decryption result under Sk_i, that is, μ'_i = Dec(Sk_i, c). For i ∈ {0, 1}, we also define μ_i = μ'_i if c = Enc(e_k, μ_i) and μ_i ⊨ ⊥ otherwise.

We have five cases classified as follows:

- Case 1 (μ_0 = μ_1 ≠ ⊥): The condition that μ_0 = μ_1 ≠ ⊥ violates the SCFR-CCA security (or the XCFR security) of the underlying PKE and it is easy to make a reduction.

- Case 2 (μ_0 = ⊥ ≠ μ_1 ≠ ⊥): In this case, the decapsulation algorithm outputs K = H(μ_0) = H(μ_1) and we find a collision for H. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

- Case 3 (μ_0 = ⊥ ≠ μ_1 ≠ ⊥): In this case, the decapsulation algorithm outputs K = H_{prf}(μ_0, μ_1) and we find a claw ((s_0, c), μ_1) of H_{prf} and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).

- Case 4 (μ_0 ≠ ⊥ and μ_1 = ⊥): In this case, the decapsulation algorithm outputs K = H(μ_0) = H_{prf}(s_1, c) and we find a claw ((μ_0, s_1), c) of H and H_{prf}. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).

- Case 5 (The other cases): In this case, we find a collision ((s_0, c), (s_1, c)) of H_{prf}, which is indeed collision if s_0 ≠ s_1 which occurs with probability at least 1 - 1/2^ℓ. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

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

5 NTRU

We briefly review NTRU [CDH*20] in subsection 5.1, discuss the security properties of the underlying PKE, NTRU-DPKE, in subsection 5.2, and discuss the security properties of NTRU in subsection 5.3. We want to show that, under appropriate assumptions, NTRU is ANON-CCA-secure in the QROM, and NTRU leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM. In order to do so, we show that the underlying NTRU-DPKE of NTRU is strongly disjoint-simulatable under the modified DSPR and PLWE assumptions and XCFR-secure in subsection 5.2. Since NTRU is obtained by applying SXY to NTRU-DPKE, the former implies that NTRU is SPR-CCA-secure and SSMT-CCA-secure in the QROM under those assumptions and the latter implies that NTRU is SCFR-CCA-secure in the QROM. Those three properties lead to the anonymity of NTRU and hybrid PKE in the QROM as we wanted.

5.1 Review of NTRU

**Preliminaries:** Φ_n denotes the polynomial x - 1 and Φ_n denotes (x^n - 1)/(x - 1) = x^{n-1} + x^{n-2} + ⋯ + 1. We have x^n - 1 = Φ_n Φ_n R, R/3, and R/q denotes Z[x]/(Φ_n Φ_n), Z[x]/(3, Φ_n), and Z[x]/(q, Φ_n), respectively. S, S/3, and S/q denotes Z[x]/(Φ_n), Z[x]/(3, Φ_n), and Z[x]/(q, Φ_n), respectively.

We say a polynomial ternary if its coefficients are in {−1, 0, +1}. S3(a) returns a canonical S/3-representative of z ∈ Z[x], that is, b ∈ Z[x] of degree at most n - 2 with ternary coefficients in {−1, 0, +1} such that a ≡ b (mod (3, Φ_n)). Let T be a set of non-zero ternary polynomials of degree at most n - 2, that is, T = {a = Σ_{i=1}^{n-2} a_i x^i : a ≠ 0 ∧ a_i ∈ {−1, 0, +1}}. We say a ternary polynomial v = Σ_i v_i x^i has the non-negative correlation property if Σ_i v_i x^i  0. T_F 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 ∈ T : |{a_i : a_i = 1}| = |{a_i : a_i = -1}| = d/2}. The following lemma is due to Schank [Sch21]. (See, e.g., [CDH*20] for this design choice.)

**Lemma 5.1.** Suppose that (n, q) = (509, 2048), (677, 2048), (821, 4096), or (701, 8192), which are the parameter sets in NTRU. If r ∈ T, then r has an inverse in S/q.

**Proof.** Φ_n is irreducible over P_3 if and only if n is prime and 2 is primitive element in P_3^n (See e.g., Cohen et al. [CFA05]). The conditions are satisfied for all n = 509, 677, 701, and 821. Hence, Z[x]/(2, Φ_n) is a finite field and every polynomial r in T has an inverse in Z[x]/(2, Φ_n). Such r is also invertible in S/q = Z[x]/(q, Φ_n) with q = 2^k for some k and, indeed, one can find it using the Newton method or the Hensel lifting. □
NTRU: NTRU involves four subsets $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m$ of $R$. It uses Lift$(m): \mathcal{L}_m \rightarrow R$. NTRU has two types of parameter sets, NTRU-HPS and NTRU-HRSS, specified as later.

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

It uses \text{Samplefg()} to sample $f$ and $g$ from $\mathcal{L}_f$ and $\mathcal{L}_g$. NTRU also uses \text{Sample_rm()} to sample $r$ and $m$ from $\mathcal{L}_r$ and $\mathcal{L}_m$.

The underlying DPKE of NTRU, which we call NTRU-DPKE, is defined as Figure 6. We note that, for an encryption key $h$, we have $h \equiv 0 \pmod{(q, \Phi_1)}$, $h$ is invertible in $S/q$, and $hr + m \equiv 0 \pmod{(q, \Phi_1)}$. (See [CDH*20, Section 2.3].)

NTRU then applies SX to NTRU-DPKE in order to obtain IND-CCA-secure KEM as in Figure 7, where $H = \text{SHA3-256}$ and $H_{prf} = \text{SHA3-256}$. Since the lengths of their input spaces differ, we can treat them as different random oracles.

Rigidity: NTRU uses SXY, while its KEM version (Figure 7) seems to lack the re-encryption check. We note that NTRU implicitly checks $hr + \text{Lift}(m) = c$ by checking if $(r, m) \in \mathcal{L}_r \times \mathcal{L}_m$ in NTRU-DPKE (Figure 6). See [CDH*20] for the details.

5.2 Properties of NTRU-DPKE

We show that NTRU-DPKE is strongly disjoint-simulatable and XCFR-secure.

We have known that the generalized NTRU PKE is pseudorandom [SS10] and disjoint-simulatable [SXY18] if the decisional small polynomial ratio (DSPR) assumption [LTV12] and the polynomial learning with errors (PLWE) assumption [SSTX09, LPR10] hold. See [SXY18, Section 3.3 of the ePrint version].

Let us adapt their arguments to NTRU-DPKE. We modify the DSPR and the PLWE assumptions as follows:

**Definition 5.1.** Fix the parameter set. Define $R' \equiv \{c \in R/q : c \equiv 0 \pmod{(q, \Phi_1)}\}$, which is efficiently sampleable.
Moreover, because of the check in the decryption, we have

\[ \text{SSMT-CCA} \]

Due to Theorem 5.1, under the modified DSPR and PLWE assumptions, NTRU is secure in the QROM, combined with ANON-CCA-secure SCFR, which implies

\[ \text{ANON-CCA} \]

we obtain the following theorems.

Theorem 5.3. NTRU-DPKE is strongly disjoint-simulatable under those two assumptions:

**Lemma 5.2.** Suppose that the modified DSPR and PLWE assumptions hold. Then, NTRU-DPKE is strongly disjoint-simulatable with a simulator \( S \) that outputs a random polynomial chosen from \( R' \).

**Proof.** The proof of ciphertext-indistinguishability is obtained by modifying the proof in [SXY18]. We want to show that \((h, c') = hr + \text{Lift}(m) \mod (q, \Phi_1 \Phi_n)\) if \( h \equiv 3g \cdot f q \mod (q, \Phi_1 \Phi_n) \) and \( f = (1/f) \mod (q, \Phi_n) \).

- We first replace \( h \) with \( h' \), which is justified by the modified DSPR assumption.
- We next replace \( c = h'r + \text{Lift}(m) \mod (q, \Phi_1 \Phi_n) \) with \( c' \), which is justified by the modified PLWE assumption.
- We then go backward by replacing random \( h' \) with \( h \), which is justified by the modified DSPR assumption again.

Statistical disjointness follows from the fact that \(|R'| = q^{n-1} \gg 2^{2n} = |T \times T| \geq |L_m \times L_r| \geq |\text{Enc}(h, L_m \times L_r)| \)

Since \( R' \) is independent of an encryption key \( k \), NTRU-DPKE is strong disjoint-simulatability.

We next show the XCFR security of NTRU-DPKE.

**Lemma 5.3.** NTRU-DPKE is XCFR-secure.

**Proof.** Suppose that the adversary wins with its output \( e k_i, d k_0, d k_1 \), where \( e k_i = h_i \) for \( i \in \{0, 1\} \). Let us define \( \mu_0 = \text{Dec}(d k_0, c) \) and \( \mu_1 = \text{Dec}(d k_1, c) \).

If the adversary wins, we can assume \( \mu_0 = \mu_1 = (r, m, 0) \in L_r \times L_m \times \{0, 1\} \). Otherwise, that is, if \( \mu_0 = \mu_1 = (0, 0, 1) \), then the output is treated as \( \perp \) and the adversary loses.

Moreover, because of the check in the decryption, we have \( c \equiv b_0 \cdot r + \text{Lift}(m) \equiv h_1 \cdot r + \text{Lift}(m) \mod (q, \Phi_1 \Phi_n) \), which implies \( r (h_0 - h_1) \equiv 0 \mod (q, \Phi_n) \). On the other hand, according to Lemma 5.1, for any \( r \in L_r = T \), we have \( r \cdot 0 \equiv 0 \in S/q \) with negligible probability. Thus, all but negligible choices of \( h_0 \) and \( h_1 \), any \( r \in L_r = T \) results in \( r (h_0 - h_1) \equiv 0 \mod (q, \Phi_n) \) and \( h_0 \cdot r + \text{Lift}(m) \equiv h_1 \cdot r + \text{Lift}(m) \mod (q, \Phi_1 \Phi_n) \). Hence, the probability that the adversary wins is negligible, concluding the proof.

5.3 Properties of NTRU

Combining NTRU-DPKE’s strong disjoint-simulatability and XCFR security with previous theorems on XSY, we obtain the following theorems.

**Theorem 5.1.** Suppose that the modified DSPR and PLWE assumptions hold. Then, NTRU is SPR-CCA-secure and SSMT-CCA-secure in the QROM.

**Proof.** Under the modified DSPR and PLWE assumptions, NTRU-DPKE is strongly disjoint-simulatable [Lemma 5.2]. In addition, NTRU-DPKE is perfectly correct. Applying **Theorem 4.2** and **Theorem 4.3**, we obtain the theorem.

**Theorem 5.2.** NTRU is SCFR-CCA-secure in the QROM.

**Proof.** NTRU-DPKE is XCFR-secure [Lemma 5.3]. Applying **Theorem 4.4**, we have that NTRU is SCFR-CCA-secure in the QROM.

**Theorem 5.3.** Under the modified DSPR and PLWE assumptions, NTRU is ANON-CCA-secure in the QROM.

**Proof.** Due to **Theorem 5.1**, under the modified DSPR and PLWE assumptions, NTRU is SPR-CCA-secure in the QROM. Thus, applying **Theorem 2.5**, we have that, under those assumptions, NTRU is ANON-CCA-secure in the QROM.

**Theorem 5.4.** Under the modified DSPR and PLWE assumptions, NTRU leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM, combined with SPR-ortCCA-secure and FROB-secure DEM.

**Proof.** Due to **Theorem 5.1**, under the modified DSPR and PLWE assumptions, NTRU is SPR-CCA-secure and SSMT-CCA-secure in the QROM. Moreover, NTRU is perfectly correct. Thus, combining NTRU with SPR-ortCCA-secure DEM, we obtain a SPR-CCA-secure hybrid PKE in the QROM [**Theorem 3.2**]. Moreover, NTRU is SCFR-CCA-secure in the QROM [**Theorem 5.2**]. Thus, if DEM is FROB-secure, then the hybrid PKE is SROB-CCA-secure [**Theorem 2.2**].
Acknowledgement

The author is grateful to John Schanck for insightful comments and suggestions on NTRU, Akinori Hosoyamada and Takashi Yamakawa for insightful comments and discussion on quantum random oracles, and Kohei Nakagawa for discussion on the collision problem in SIKE. The author would like to thank Daniel J. Bernstein for insightful comments and discussion on the indistinguishability of the quantum random oracles. The author would like to thank anonymous reviewers for their valuable comments and suggestions on this paper.

References


Hos21. Akinori Hosoyamada. personal communication, June 2021. 5


A Missing Lemma

Lemma A.1. Let A and B denote events. Suppose that we have \( \Pr[A] \leq \delta \). For any \( p \geq 0 \), we have
\[
|\Pr[B] - p| \leq |\Pr[B \land \neg A] - p| + \delta \quad \text{and} \quad |\Pr[B \land \neg A] - p| \leq |\Pr[B] - p| + \delta.
\]

Proof. Those bounds are obtained by using the triangle inequality. We have
\[
|\Pr[B] - p| = |\Pr[B \land A] + \Pr[B \land \neg A] - p| \leq \Pr[B \land A] + |\Pr[B \land \neg A] - p|
\]
and
\[
|\Pr[B \land \neg A] - p| = |\Pr[B \land \neg A] + \Pr[B \land A] - \Pr[B \land A] - p|
\]
\[
= |\Pr[B] - p - \Pr[B \land A]| \leq |\Pr[B] - p| + |\Pr[B \land A]|
\]
\[
\leq |\Pr[B] - p| + |\Pr[A]| \leq |\Pr[B] - p| + \delta
\]
as we wanted. \( \Box \)

The following lemma is called the oneway-to-hiding (O2H) lemma, which is proven by Unruh [Unr15, Lemma 6.2]. Roughly speaking, the lemma states that if any quantum adversary issuing at most \( q \) queries to a quantum random oracle \( H \) can distinguish \((x, H(x))\) from \((x, y)\), where \( y \) is chosen uniformly at random, then we can find \( x \) by measuring one of the adversary’s queries. The following lemma is a generalized version of the O2H lemma taken from [AHU19].

Lemma A.2 (Oneway to Hiding [AHU19, Theorem 3]). Let \( S \subseteq X \) be random. Let \( G, H : X \to Y \) be random functions satisfying \( G(x) \approx H(x) \) for every \( x \notin S \). Let \( z \) be a random bit string. \((S, G, H, z)\) may have arbitrary joint distribution.

Let \( A \) be a quantum oracle algorithm with query depth \( d \) (not necessarily unitary).

Let \( B^H \) be an oracle algorithm that on input \( z \) does the following: pick \( i \leftarrow \{1, \ldots, d\} \), run \( A^i(z) \) until (just before) the \( i \)-th query, measure all query input registers in the computational basis, output the set \( T = \{t_1, \ldots, t_{|T|}\} \) of measurement outcomes.

Let
\[
P_{\text{left}} := \Pr_{H, z}[b \leftarrow A^i(z) : b = 1],
\]
\[
P_{\text{right}} := \Pr_{G, z}[b \leftarrow A^G(z) : b = 1],
\]
\[
P_{\text{guess}} := \Pr_{H, G, S, z}[T \leftarrow B^H(z) : S \cap T \neq \emptyset].
\]

Then,
\[
|P_{\text{left}} - P_{\text{right}}| \leq 2d \sqrt{P_{\text{guess}}} \quad \text{and} \quad \sqrt{P_{\text{left}} - P_{\text{right}}} \leq 2d \sqrt{P_{\text{guess}}}.
\]

The same result holds with \( B^G \) instead of \( B^H \) in the definition of \( P_{\text{guess}} \).

In this paper, we use lemma in [Unr15, HHK17] stated as follows:
Table 5. Summary of Games for the Proof of Theorem 3.1

<table>
<thead>
<tr>
<th>Game</th>
<th>$c^<em>$ and $K^</em>$</th>
<th>$d^*$</th>
<th>Derpory oracle</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game_0</td>
<td>$Enc(ek)$</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Game_1</td>
<td>$Enc(ek)$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>Use $K^<em>$ if $c = c^</em>$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game_2</td>
<td>$Enc(ek)$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>Use $K^<em>$ if $c = c^</em>$</td>
<td>SPR-CCA security of KEM^4</td>
</tr>
<tr>
<td>Game_3</td>
<td>$S(1^* \times U(\mathcal{K}))$ at first</td>
<td>$E(K^<em>, \mu^</em>)$</td>
<td>Use $\perp$ if $c = c^*$</td>
<td>SPR-orCCA security of DEM</td>
</tr>
<tr>
<td>Game_4</td>
<td>$S(1^* \times U(\mathcal{K}))$ at first</td>
<td>$U(C_{\mu^*})$</td>
<td>$\perp^*$-sparserness of KEM^4</td>
<td></td>
</tr>
</tbody>
</table>

Corollary A.1 (Algorithmic One-way-to-Hiding lemma). Let $H: \mathcal{X} \to \mathcal{Y}$ be a quantum random oracle, and let $\mathcal{A}$ be an adversary issuing at most $q$ queries to $H$ that on input $(x, y) \in \mathcal{X} \times \mathcal{Y}$ outputs either 0/1. Let $\mathcal{D}_X$ be a some distribution over $\mathcal{X}$.

For all (probabilistic) algorithms $\mathcal{F}$ whose input space is $\mathcal{X} \times \mathcal{Y}$ and which do not make any hash queries to $H$, we have

$$\Pr[x \leftarrow \mathcal{D}_X; y \leftarrow H(x); \text{inp} \leftarrow \mathcal{F}(x, y); b \leftarrow \mathcal{A}^H(\text{inp}: b = 1)] - \Pr[x \leftarrow \mathcal{D}_X; y \leftarrow \mathcal{Y}; \text{inp} \leftarrow \mathcal{F}(x, y); b \leftarrow \mathcal{A}^H(\text{inp}: b = 1)] \leq 2q \cdot \sqrt{\Pr[x \leftarrow \mathcal{D}_X; y \leftarrow \mathcal{Y}; \text{inp} \leftarrow \mathcal{F}(x, y); x' \leftarrow \text{Ext}^H(\text{inp}: x' = x)}$$

where $\text{Ext}$ picks $i \leftarrow \{1, \ldots, q\}$, runs $H^i(\text{inp})$ until $i$-th query $i$ to $H$, and returns $x' := \text{Meas}(i)$ (when $\mathcal{A}$ makes fewer than $i$ queries, $\text{Ext}$ outputs $\perp \notin \mathcal{X}$).

We can obtain the corollary by picking $H$ uniformly at random, pick $x \leftarrow \mathcal{D}_X$, pick $y$ uniformly at random, set $\mathcal{S} := \{1\}$, $G(x) := y$, and $\zeta := \text{inp} \leftarrow H(x, H(x))$. We then have

$$P_{\text{right}} = \Pr[b \leftarrow \mathcal{A}^C(z): b = 1] = \Pr[x \leftarrow \mathcal{D}_X, y \leftarrow \mathcal{Y}, \text{inp} \leftarrow H(x, H(x)), b \leftarrow \mathcal{A}^C(\text{inp}: b = 1] = \Pr[x \leftarrow \mathcal{D}_X, y \leftarrow \mathcal{Y}, \text{inp} \leftarrow H(x, y), b \leftarrow \mathcal{A}^C(\text{inp}: b = 1].$$

The last equality follows from the fact that the distribution of $H(x)$ and $y$ are equivalent and we can switch them.

**B Missing Proofs**

B.1 Proof of Theorem 3.1

We consider $Game_i$, for $i = 0, \ldots, 7$ defined later. We summarize the games in Table 5. Let $S_i$ denote the event that the adversary outputs $b' = 1$ in $Game_i$.

Let $\mathcal{S}$ be the simulator for the SPR-CCA security of KEM$^4$. We define $S_{spr}^\mathcal{K}(1^*, \mu^*) := \mathcal{S}(1^*) \times U(C_{\mu^*})$ as the simulator for the SPR-CCA security of PKEm$^\mathcal{K}$.

The security proof is similar to the security proof of the IND-CCA security of KEM/DEM [CS03] for $Game_0, \ldots, Game_4$. We need to take care of pseudorandom ciphertexts when moving from $Game_4$ to $Game_7$ and require the INT-CTXT security of DEM and the $\perp$-sparserness of KEM$^4$.

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

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

We have

$$\Pr[S_0] = \Pr[\text{Expt}_{\text{PKEm}^\mathcal{K}, \mathcal{S}_{\text{spr}}, \mathcal{A}}^{\text{spr-cca}}(k) = 1 | b = 0].$$

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

$$\Pr[S_0] = \Pr[S_1].$$
Game₂: In this game, the decryption oracle uses $K^*$ if $c = c^*$ instead of $K = \text{Dec}(dk, c^*)$. Game₁ and Game₂ differ if correctly generated ciphertext $c^*$ with $K^*$ is decapsulated into different $K \neq K^*$ or $\bot$, which violates the correctness and occurs with probability at most $\delta$. Hence, the difference of Game₁ and Game₂ is bounded by $\delta$, and we have
\[ \Pr[S_2] - \Pr[S_1] \leq \delta. \]
This bound is corresponding to the event BadKeyPair in [CS03].

Game₃: In this game, the challenger uses random $(c^*, K^*)$ generated by the simulator and uses $K^*$ in DEM. The challenge ciphertext is generated as follows:
\[ (c^*, K^*) \leftarrow S(1^k) \times U(\mathcal{K}); d^* \leftarrow E(K^*, \mu^*); \text{return } ct^* = (c^*, d^*). \]
The difference between Game₂ and Game₃ is bounded by the SPR-CCA security of KEM²: There is an adversary $\mathcal{A}_{23}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying
\[ \Pr[S_2] - \Pr[S_1] \leq \text{Adv}^{\text{spr-cca}}_{\text{KEM}^2, S, \mathcal{A}_{23}}(\kappa). \]
We omit the detail of $\mathcal{A}_{23}$ since it is straightforward.

Game₄: In this game, the challenger uses random $d^*$. The challenge ciphertext is generated as follows:
\[ (c^*, K^*) \leftarrow S(1^k) \times U(\mathcal{K}); d^* \leftarrow U(\mathcal{G}_{\mu^*}); \text{return } ct^* = (c^*, d^*). \]
The difference between Game₃ and Game₄ is bounded by the SPR-CCA security of DEM: There is an adversary $\mathcal{A}_{34}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying
\[ \Pr[S_2] - \Pr[S_1] \leq \text{Adv}^{\text{spr-cca}}_{\text{DEM}, \mathcal{A}_{34}}(\kappa). \]
We omit the detail of $\mathcal{A}_{34}$ since it is straightforward.

Game₅: We replace the decryption oracle. If given $ct = (c^*, d)$, the decryption oracle always return $\bot$. Let Forge be an event that the adversary queries $d \neq d^*$ decrypted into some $\mu \neq \mu^*$ by using $K^*$. Game₄ and Game₅ are equal to each other until the event Forge occurs in Game₄. Hence, the difference between Game₄ and Game₅ is bounded by the INT-CTXT security of DEM: There is an adversary $\mathcal{A}_{45}$ whose running time is approximately the same as that of $\mathcal{A}$ satisfying
\[ \Pr[S_2] - \Pr[S_1] \leq \Pr[\text{Forge}] \leq \text{Adv}^{\text{int-ctxt}}_{\text{DEM}, \mathcal{A}_{45}}(\kappa). \]
We omit the detail of $\mathcal{A}_{45}$ since it is straightforward. (We note that $\mathcal{A}_{45}$ makes no queries to Enc₂.)

Game₆: We replace the decryption oracle in Game₅ with the original one. Let Bad be the event that a randomly chosen $c^* \leftarrow S(1^k)$ is decapsulated into a key $K \neq \bot$. Game₅ and Game₆ are equivalent unless the event Bad occurs. Since KEM² is $\epsilon$-sparse, we have
\[ \Pr[S_2] - \Pr[S_1] \leq \Pr[\text{Bad}] \leq \epsilon. \]

Game₇: We change the timing of the generation of $(c^*, K^*)$ as the original. This change is just conceptual, and we have
\[ \Pr[S_0] = \Pr[S_7]. \]
Notice that this is the original game $\text{Expt}^{\text{spr-cca}}_{\text{PKE}_{\mu^*}, S_{\mu^*}, \mathcal{A}}(\kappa)$ with $b = 1$, thus, we have
\[ \Pr[S_7] = \text{Pr}[\text{Expt}^{\text{spr-cca}}_{\text{PKE}_{\mu^*}, S_{\mu^*}, \mathcal{A}}(\kappa) = 1 | b = 1]. \]
Summing the (in)equalities, we obtain the bound in the statement as follows:
\[ \text{Adv}^{\text{spr-cca}}_{\text{PKE}_{\mu^*}, S_{\mu^*}, \mathcal{A}}(\kappa) = \Pr[S_0] - \Pr[S_1] \leq \sum_{i=0}^{6} \Pr[S_i] - \Pr[S_{i+1}] \leq \text{Adv}^{\text{spr-cca}}_{\text{KEM}^2, S, \mathcal{A}_{23}}(\kappa) + \text{Adv}^{\text{spr-cca}}_{\text{DEM}, \mathcal{A}_{34}}(\kappa) + \text{Adv}^{\text{int-ctxt}}_{\text{DEM}, \mathcal{A}_{45}}(\kappa) + \delta + \epsilon. \]
Table 6. Summary of Games for the Proof of Theorem 4.1. We define $g(μ) := Enc(ek, μ) = Enc_0(ek, μ; G(μ))$.

<table>
<thead>
<tr>
<th>Game</th>
<th>$H$</th>
<th>$G$</th>
<th>$c^*$</th>
<th>$K^*$</th>
<th>Decapsulation valid $c$ invalid $c$</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game$_0$</td>
<td>$H$</td>
<td>$F(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{prf}(s, c)$</td>
<td>Lemma 2.2</td>
</tr>
<tr>
<td>Game$_1$</td>
<td>$H$</td>
<td>$F(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{q}(c)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game$_1,1$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{q}(c)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game$_1,2$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{q}(c)$ if a key pair is good</td>
<td></td>
</tr>
<tr>
<td>Game$_2$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{q}(c)$ if a key pair is good</td>
<td></td>
</tr>
<tr>
<td>Game$_3$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$Enc(ek, μ^*)$</td>
<td>$H(μ^*)$</td>
<td>$H(μ)$ $H_{q}(c)$ conceptual</td>
<td></td>
</tr>
<tr>
<td>Game$_4$</td>
<td>$H$</td>
<td>$F(M, R)$</td>
<td>$S(1^k)$</td>
<td>$U(K)$</td>
<td>$H(μ)$ $H_{q}(c)$ statistical disjointness</td>
<td></td>
</tr>
<tr>
<td>Game$_5$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$S(1^k)$</td>
<td>$U(K)$</td>
<td>$H(μ)$ $H_{q}(c)$ If a key pair is good</td>
<td></td>
</tr>
<tr>
<td>Game$_6$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$S(1^k)$</td>
<td>$U(K)$</td>
<td>$H(μ)$ $H_{q}(c)$ conceptual</td>
<td></td>
</tr>
<tr>
<td>Game$_7$</td>
<td>$H$</td>
<td>$F_{ek,dk}(M, R)$</td>
<td>$S(1^k)$</td>
<td>$U(K)$</td>
<td>$H(μ)$ $H_{q}(c)$ If a key pair is good</td>
<td></td>
</tr>
<tr>
<td>Game$_8$</td>
<td>$H$</td>
<td>$F(M, R)$</td>
<td>$S(1^k)$</td>
<td>$U(K)$</td>
<td>$H(μ)$ $H_{prf}(s, c)$</td>
<td>Lemma 2.2</td>
</tr>
</tbody>
</table>

B.2 Proof of Theorem 4.1:

We use the game-hopping proof. We consider Game$_i$ for $i = 0, \ldots, 8$. We summarize the games in Table 6. Let $S_i$ denote the event that the adversary outputs $b_1 = 1$ in game Game$_i$. We extend the security proof for SXYS in [LW21], which extends the security proof for SXYS [SXY18, XY19] to the case that the underlying PKE is randomized by $K \circ T$.

Game$_0$: This game is the original game $Exp_{KEM,A}^{scca}(s)$ with $b = 0$. Thus, we have

$Pr[S_0] = 1 - Pr[Exp_{KEM,A}^{scca}(s) = 1 | b = 0]$.

Game$_1$: This game is the same as Game$_0$ except that $H_{prf}(s, c)$ in the decapsulation oracle is replaced by $H_{q}(c)$ where $H_q : C \rightarrow K$ is another random oracle. We remark that $A$ cannot access $H_q$ directly.

As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$|Pr[S_0] - Pr[S_1]| \leq 2(q_{H_{prf}} + q_{Dec}) \cdot 2^{-ℓ/2}$,

where $q_{H_{prf}}$ and $q_{Dec}$ denote the number of queries to $H_{prf}$ and $Dec$ the adversary makes, respectively.

Definition of $F_{ek,dk}(M, R)$: We consider a set of good random oracles $G, F_{ek,dk}(M, R)$. The following definition is taken from [HHK17, JZC18, HKSU20, LW21]: For $(ek, dk) \in Gen_0()$ and $μ ∈ M$, we define the set of good randomness as $R_{ek,dk,μ} := \{r ∈ R : Dec_0(dk, Enc_0(ek, μ; r)) = μ\}$, which could be empty. Let $F_{ek,dk}(M, R)$ be a set of functions $G : M \rightarrow R$ satisfying $G(μ) ∈ F_{ek,dk,μ}$ for all $μ ∈ M$. Define $δ_{ek,dk,μ} := |R \setminus G_{ek,dk,μ}| / |R|$, which is the fraction of the bad randomness for $μ$. Define $δ_{ek,dk} := \max_{μ ∈ M} δ_{ek,dk,μ}$. We note that $δ = Exp_{ek,dk} \rightarrow Gen_0(1^k) [δ_{ek,dk}]$.

Game$_1,1$: This game is the same as Game$_1$ except that the random oracle $G$ is chosen from $F_{ek,dk}(M, R)$ instead of $F(M, R)$.

If we fix $(ek, dk)$, then we have $|Pr[S_1 | (ek, dk)] - Pr[S_{11} | (ek, dk)]| \leq 8(q_G + q_{Dec} + 1)^2 \cdot δ_{ek,dk}$. See [HKSU20, Theorem 3.2] and [LW21, Claim 1] for the analysis using Lemma 2.1. Taking the average over $(ek, dk) \leftarrow Gen_0(1^k)$, we obtain

$|Pr[S_1] - Pr[S_{11}]| \leq 8(q_G + q_{Dec} + 1)^2 \cdot Exp_{ek,dk} \rightarrow Gen_0(1^k) [δ_{ek,dk}] = 8(q_G + q_{Dec} + 1)^2 δ$.

where $q_G$ denotes the number of queries to $G$ the adversary makes.
Definition of Bad and Good: We next define a bad event for key pairs. This definition is taken from [LW21]. Let us define an event Bad that there exists $\mu \in M$ such that any $r \in R$ is bad randomness, that is,
\[
\text{Bad} \triangleq \text{boole } \left( \exists \mu \in M : \mathcal{R}_{\text{ek},dk,\mu}^{\text{good}} = 0 \right),
\]
where randomness is taken over $(ek,dk) \leftarrow \text{Gen}_0(1^k)$. We define $\text{Good} = \neg \text{Bad}$. We have $\Pr[\neg \text{Good}] = \Pr[\text{Bad}] \leq \delta$ ([LW21, Claim 3]).

According to Lemma A.1, for any $p \geq 0$, we also have
\[
|\Pr[S_{1,1}] - p| \leq |\Pr[S_{1,1} \land \text{Good}] - p| + \delta.
\]

Game$_1$: This game is the same as Game$_{1,1}$ except that the random oracle $H(\cdot)$ is simulated by $H'_{\mathcal{R}}(\text{Enc}(ek,\cdot))$ where $H'_{\mathcal{R}} : C \rightarrow \mathcal{K}$ is yet another random oracle. We remark that the decapsulation oracle and the generation of $K^*$ also use $H'_{\mathcal{R}}(\text{Enc}(ek,\cdot))$ as $H(\cdot)$.

If Good occurs, then $\text{PKE} = \mathcal{T}[\text{PKE},G]$ is perfectly correct from the definition of $G$ and $g(\mu) := \text{Enc}(ek,\mu;G(\mu))$ is injective. Thus, if Good occurs, then $H'_{\mathcal{R}} \circ g : M \rightarrow \mathcal{K}$ is a random function and the two games Game$_{1,1}$ and Game$_1$ are equivalent. We have
\[
\Pr[S_{1,1} \land \text{Good}] = \Pr[S_{1,2} \land \text{Good}].
\]

Game$_2$: This game is the same as Game$_{1,2}$ except that the random oracle $H$ is simulated by $H_g \circ g$ instead of $H'_{\mathcal{R}} \circ g$.

As in the discussion of Theorem 4.2 on the difference between Game$_{1,5}$ and Game$_2$, using the fact that, if Good occurs, $\text{PKE} = \mathcal{T}[\text{PKE},G]$ is perfectly correct, we can show that the two games Game$_{1,2}$ and Game$_2$ are equivalent and we have
\[
\Pr[S_{1,2} \land \text{Good}] = \Pr[S_2 \land \text{Good}].
\]

Game$_3$: This game is the same as Game$_2$ except that $K^*$ is set as $H'_{\mathcal{R}}(c^*)$ and the decapsulation oracle always returns $H'_{\mathcal{R}}(c)$ as long as $c \neq c^*$. This modified decapsulation oracle is denoted by $\text{Dec}'$.

If Good occurs, then $\text{PKE} = \mathcal{T}[\text{PKE},G]$ is perfectly correct from the definition of $G$. Thus, the two games Game$_2$ and Game$_3$ are equivalent and we have
\[
\Pr[S_2 \land \text{Good}] = \Pr[S_3 \land \text{Good}].
\]

In addition, according to Lemma A.1, for any $p \geq 0$, we have
\[
|\Pr[S_3 \land \text{Good}] - p| \leq |\Pr[S_3] - p| + \delta.
\]

Game$_{1,1}$: This game is the same as Game$_3$ except that $G$ is chosen from $\mathcal{F}(M,R)$ instead of $\mathcal{F}_{\text{ek},dk}(M,R)$.

As the difference between Game$_1$ and Game$_{1,1}$, we have
\[
|\Pr[S_3] - \Pr[S_{1,1}]| \leq 8(q_G + q_H + 1)^2 \cdot \text{exp}_{(ek,dk)\rightarrow \text{Gen}_0(1^*)}[\delta_{ek,dk}] = 8(q_G + q_H + 1)^2 \delta,
\]
where $q_H$ is the number of queries to $H$ the adversary makes. We note that $H$ queries to $G$ internally.

Game$_4$: This game is the same as Game$_3$ except that $c^*$ is generated by $S(1^k)$.

The difference between two games Game$_3$ and Game$_4$ is bounded by the advantage of ciphertext indistinguishability in disjoint simulatability as in [XY19, Lemma 4.7]. We have
\[
|\Pr[S_3] - \Pr[S_4]| \leq \text{Adv}^{\text{dis-ind}}_{\text{PKE},\mathcal{D}_M,S,\mathcal{A}_M}(\kappa).
\]

The reduction algorithm is obtained straightforwardly.

Game$_5$: This game is the same as Game$_4$ except that $K^* \leftarrow \mathcal{K}$ instead of $K^* \leftarrow H_g(c^*)$.

In Game$_4$, if $c^* \leftarrow S(1^k)$ is not in $\text{Enc}(ek,M)$, then the adversary has no information about $K^* = H_g(c^*)$ and thus, $K^*$ looks uniformly at random. Hence, the difference between two games Game$_4$ and Game$_5$ is bounded by the statistical disjointness in disjoint simulatability as in [XY19, Lemma 4.8]. We have
\[
|\Pr[S_4] - \Pr[S_5]| \leq \text{Disj}^{\text{PKE,S}}_{\text{S}}(\kappa).
\]
Game $G_{1,1}$: This game is the same as $\text{Game}_5$ except that $G$ is chosen from $\mathcal{F}^\text{good}_{ek,dk}(M, \mathcal{R})$ instead of $\mathcal{F}(M, \mathcal{R})$. As the difference between $\text{Game}_3$ and $\text{Game}_{1,1}$, we have

$$|\Pr[S_5] - \Pr[S_{1,1}]| \leq 8(q_G + q_H + 1)^2 \cdot \text{Exp}(ek,dk) \cdot \Gamma_{\text{good}}(1^\kappa) [\delta, dl] \leq 8(q_G + q_H + 1)^2 \delta.$$

We note that $H$ queries to $G$ internally. In addition, according to Lemma A.1, for any $p \geq 0$, we have

$$|\Pr[S_{1,1}] - p| \leq |\Pr[S_{1,1} \wedge \text{Good}] - p| + \delta.$$

**Game $G_{1,2}$**: This game is the same as $\text{Game}_1$ except that the decapsulation oracle is reset as $\text{Dec}$. Similar to the case for $\text{Game}_2$ and $\text{Game}_3$, if $G$ occurs, then the two games $\text{Game}_6$ and $\text{Game}_6$ are equivalent. We have

$$\Pr[S_{6,1} \wedge \text{Good}] = \Pr[S_{6} \wedge \text{Good}].$$

**Game $G_{6,1}$**: This game is the same as $\text{Game}_6$ except that the random oracle $H$ is simulated by $H'_q \circ g$ where $H'_q : C \rightarrow K$ is yet another random oracle as in $\text{Game}_{1,2}$. Similar to the case for $\text{Game}_{1,2}$ and $\text{Game}_2$, if $G$ occurs, then the two games $\text{Game}_{6,1}$ and $\text{Game}_{6,2}$ are equivalent. We have

$$\Pr[S_{6,1} \wedge \text{Good}] = \Pr[S_{6,2} \wedge \text{Good}].$$

In addition, according to Lemma A.1, for any $p \geq 0$, we have

$$|\Pr[S_{6,2} \wedge \text{Good}] - p| \leq |\Pr[S_{6,2}] - p| + \delta.$$

**Game $G_{6,2}$**: This game is the same as $\text{Game}_{6,1}$ except that the random oracle $H(\cdot)$ is set as the original. Similar to the case for $\text{Game}_{1,1}$ and $\text{Game}_{1,2}$, if $G$ occurs, then the two games $\text{Game}_{6,1}$ and $\text{Game}_{6,2}$ are equivalent. We have

$$\Pr[S_{6,1} \wedge \text{Good}] = \Pr[S_{6,2} \wedge \text{Good}].$$

As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_7] - \Pr[S_8]| \leq 2(q_{H_{\text{prf}}} + q_{\text{Dec}}) \cdot 2^{-\ell/2}.$$

We note that this game is the original game $\text{Exp}_{\text{KEM}, A}(\kappa)$ with $b = 1$. Thus, we have

$$\Pr[S_8] = \Pr[\text{Exp}_{\text{KEM}, A}(\kappa) = 1 | b = 1].$$

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

$$\text{Adv}_{\text{KEM}, A}(\kappa) = |\Pr[S_6] - \Pr[S_8]| \leq \text{Adv}_{\text{KEM}, A}(\kappa) + \text{Adv}_{\text{KEM}, A}(\kappa) + 16(q_G + q_{\text{Dec}} + 1)^2 \delta + 16(q_G + q_{H} + 1)^2 \delta + 4(q_{H_{\text{prf}}} + q_{\text{Dec}}) \cdot 2^{-\ell/2}.$$

**C Variants of the Fujisaki-Okamoto Transform**

In this section we review the variants of the FO transforms. The Fujisaki-Okamoto (FO) transform FO converts weakly-secure probabilistic PKE scheme $\text{PKE}_0$ into IND-CCA-secure KEM scheme. Hotheinze et al. [HHK17] decomposed the FO transform FO into two transforms $T$ and $U$. In this section we review the variants of the FO transforms, we define variants of $U$ and then define the variants of FO by combining with $T$.

**C.1 Transform $T$**

In the original $T$ in [HHK17, Section 3.1], the decryption algorithm checks the validity of $c$ by re-encryption check. We omit this re-encryption check.

Let $\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ be a probabilistic PKE scheme, whose ciphertext space is $C_{\text{PKE}}$, message space is $M$, and randomness space is $R_{\text{Enc}}$. Let $G : M \rightarrow R_{\text{Enc}}$ be a hash function modeled by the random oracle.

$\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec}) = T[\text{PKE}_0, G]$ is defined as in Figure 8.
whose ciphertext space is $C$. Let us review the definitions of the variants: Let $HU$'s variants, $U^L$, $U^L_\perp$, $U^L_m$, and $U^L_\perp_m$ [HHK17], where the superscript "$L$" and "$\perp$" implies \textit{implicit rejection} and \textit{explicit rejection}, respectively, and the subscript "$m$" implies the computation of key $K$ involves a plaintext $\mu$ only, while if there is no subscript, then it involves $\mu$ and ciphertext $c$.

Saito et al. defined $SXY$, which is essentially the same as $U^L_\perp$ [SXY18]. Bindel et al. discussed the relations of the IND-CCA security of KEM schemes obtained by those transforms via indifferentiable reductions [BHH’19]. In their discussion, they modify $U^L$, which we write $U^{L,pf}$ here. In their $U^L$, they use $K := H_{pf}(s, c)$ for invalid ciphertext $c$ instead of $K := H(s, c)$ as in [HHK17].

Let us review the definitions of the transforms. Let $PKE = (Gen, Enc, Dec)$ be a deterministic PKE scheme, whose ciphertext space is $C$ and message space is $M$. Let $H: M \times C \rightarrow \mathcal{K}$ be a hash function modeled by the random oracle. Let $H_{pf}: M \times C \rightarrow \mathcal{K}$ be another hash function modeled by the random oracle.

\begin{itemize}
  \item $U^L[PKE, H]: KEM = (Gen, Enc, Dec) = U^L[PKE, H]$ is defined as in Figure 9.
  \item $U^{L,pf}[PKE, H, H_{pf}]:$ This transform is the same as $U^L$ except that line 3 of $Dec$ is replaced by "then return $K := H_{pf}(s, c)$".
  \item $U^L_\perp[PKE, H]:$ This transform is the same as $U^L$ except that line 3 of $Dec$ is replaced by "then return $K := H(s, c)$." This variant does not require $s$ in $\overline{\mathcal{K}}$.
  \item $U^{L_\perp}[PKE, H, H_{pf}]:$ Let $H: M \rightarrow \mathcal{K}$ be a hash function modeled by the random oracle. This transform is the same as $U^{L_\perp}$ except that line 3 of $Enc$ is replaced by "$K := H(\mu)" and line 4 of $Dec$ is replaced by "else return $K := H(\mu)$." 
  \item $U^{L_\perp}_m[PKE, H, H_{pf}]:$ Let $H: M \rightarrow \mathcal{K}$ be a hash function modeled by the random oracle. This transform is the same as $U^{L_\perp}$ except that line 3 of $Enc$ is replaced by "$K := H(\mu)" and line 4 of $Dec$ is replaced by "$\perp\mathrm{return} K := H(\mu)$." This variant does not require $s$ in $\overline{\mathcal{K}}$.
\end{itemize}

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

C.3 Variants of $HU$

Targh and Unruh [TU16] introduced a variant of $FO$ transform for PKE, whose ciphertext has an additional hash value of a random message $\mu$. Hofheinz et al. called this variant $QFO$ and they decomposed it into $T$ and $QU$. [HHK17]. Hofheinz et al. defined $QU$’s variants, $QU^L_m$ and $QU^L_\perp_m$. In those variants a ciphertext includes an additional hash $L := F(\mu)$, where $F: M \rightarrow \mathcal{M}$. (They require $\mathcal{M}$ to be a subset of a finite field.)

Jiang et al. [JZM19] defined $HU^L_m$ as a variant of $QU^L_m$, where $F: M \rightarrow \mathcal{H}$ with arbitrary $\mathcal{M}$ and $\mathcal{H}$. This allows us to make a ciphertext shorter. We define its variants $HU^L_\perp_m$, $HU^L_m$, $HU^L_\perp$, $HU^{L,pf}_m$, and $HU^{L,pf}_\perp$ as the variants of $U$. In the definition, we allow $t_L$ optional.

Let us review the definitions of the variants: Let $PKE = (Gen, Enc, Dec)$ be a deterministic PKE scheme, whose ciphertext space is $C$ and message space is $M$. Let $H: M \times C \rightarrow \mathcal{K}$ be a hash function modeled by the random oracle, and $H_{pf}: M \times C \rightarrow \mathcal{K}$ be another hash function modeled by the random oracle.

\begin{itemize}
  \item $U^L[PKE, H]: KEM = (Gen, Enc, Dec) = U^L[PKE, H]$ is defined as in Figure 9.
  \item $U^{L,pf}[PKE, H, H_{pf}]:$ This transform is the same as $U^L$ except that line 3 of $Dec$ is replaced by "then return $K := H_{pf}(s, c)$".
  \item $U^L_\perp[PKE, H]:$ This transform is the same as $U^L$ except that line 3 of $Dec$ is replaced by "then return $K := H(s, c)$." This variant does not require $s$ in $\overline{\mathcal{K}}$.
  \item $U^{L_\perp}[PKE, H, H_{pf}]:$ Let $H: M \rightarrow \mathcal{K}$ be a hash function modeled by the random oracle. This transform is the same as $U^{L_\perp}$ except that line 3 of $Enc$ is replaced by "$K := H(\mu)" and line 4 of $Dec$ is replaced by "else return $K := H(\mu)$." This variant does not require $s$ in $\overline{\mathcal{K}}$.
\end{itemize}

We adapt the discussions of Bindel et al. to SPR-CCA-security of KEM schemes obtained by the variants of $U$. See the left hand side of Figure 10.
the random oracle. Let \( H_{prf} \colon \mathcal{M} \times \mathcal{C} \rightarrow \mathcal{K} \) be another hash function modeled by the random oracle. Let \( F \colon \mathcal{M} \rightarrow \mathcal{H} \) be yet another hash function modeled by the random oracle.

- \( \text{HU}^L \circ \text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{HU}^L \circ \text{PKE}, \text{H}, \text{F} \) is defined as in Figure 11.

- \( \text{HU}^L \circ \text{prf} \circ \text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{HU}^L \circ \text{PKE}, \text{H}, \text{F}, \text{prf} \): This transform is the same as \( \text{U}^L \circ \text{KEM} \) except that line 3 of \( \text{Dec} \) is replaced by "then return \( K := H_{prf}(s, c_0, c_1) \)." This variants does not require \( s \) in \( \overline{dK} \).

- \( \text{HU}^L \circ \text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{HU}^L \circ \text{PKE}, \text{H}, \text{F} \): Let \( H \colon \mathcal{M} \rightarrow \mathcal{K} \) be a hash function modeled by the random oracle. This transform is the same as \( \text{U}^L \circ \text{KEM} \) except that line 4 of \( \text{Enc} \) is replaced by "then return \( K := H(\mu) \)," line 3 of \( \text{Dec} \) is replaced by "then return \( K := H(\mu) \)," and line 4 of \( \text{Dec} \) is replaced by "else return \( K := H(\mu) \)." This variants does not require \( s \) in \( \overline{dK} \).

---

**Figure 10.** The relation between IND-CCA and SPR-CCA security of KEMs using the variants of \( \text{U} \) and \( \text{HU} \). Dashed arrow implies the implications in [BHH+19].

---

**Figure 11.** KEM = (Gen, Enc, Dec) = HU^L [PKE, H, F]

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

### C.4 Variants of FO

Combining \( T \) and the variants of \( \text{U} \) or \( \text{HU} \), we obtain several variants of FO as follows: Let \( \text{PKE}_0 = (\text{Gen}_0, \text{Enc}_0, \text{Dec}_0) \) be a probabilistic PKE scheme: If we combine \( T \) and \( U_{prf}^L \), then we obtain \( \text{FO}_{prf}^L \). If we combine \( T \) and \( HU_{prf}^L \), then we obtain \( \text{HFO}_{prf}^L \).

### D Property of T

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

**Theorem D.1.** Suppose that a probabilistic PKE scheme \( \text{PKE}_0 \) is ciphertext indistinguishable and OW-CPA-secure. Then, \( \text{PKE} := T[\text{PKE}_0, G] \) is also ciphertext indistinguishable in the QROM.
Combining the above two lemmas, we obtain the wanted result. 

\[ \square \]

Applying the O2H lemma (Corollary A.1) with restated as Proof (Proof of Lemma D.1).

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

Lemma D.2. There exists an adversary \( A_{01} \) against OW-CPA security of PKE\(_0\) and \( A_{12} \) against ciphertext indistinguishability of PKE\(_0\) such that

\[
\text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}(\mathcal{G}) \leq 2q_G \sqrt{\text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G}) + \text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G})}. 
\]

Proof: Let us consider the following three games, Games0, Game1, and Game2. Let \( S_i \) denote the event that the adversary outputs \( b' = 1 \) in \( Game_i \).

Game0: This game is defined as follows:

\[
(ek, dk) \leftarrow \text{Gen}_0(1^k); \mu^* \in D_M; r^* \leftarrow G(\mu^*); c^* \leftarrow \text{Enc}_0(ek, \mu^*; r^*) \in SG(\mu^*); b' \leftarrow A_G(\mu^*; c^*); \text{return } b'.
\]

Game1: This game is the same as Game0 except that a randomness to generate a challenge ciphertext is freshly generated:

\[
(ek, dk) \leftarrow \text{Gen}_0(1^k); \mu^* \in D_M; r^* \leftarrow \mathcal{R}; c^* \leftarrow \text{Enc}_0(ek, \mu^*; r^*) \in SG(\mu^*); b' \leftarrow A_G(\mu^*; c^*); \text{return } b'.
\]

Game2: This game is the same as Game1 except that a challenge ciphertext is generated by the simulator \( S(1^k, \mathcal{G}) \):

\[
(ek, dk) \leftarrow \text{Gen}_0(1^k); c^* \leftarrow S(1^k, \mathcal{G}); b' \leftarrow A_G(\mu^*; c^*); \text{return } b'.
\]

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

\[
\text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}(\mathcal{G}) = |\Pr[S_0] - \Pr[S_1]|.
\]

We give an upperbound for this advantage by the following lemmas.

Lemma D.1. There exists a quantum adversary \( A_{01} \) such that

\[
|\Pr[S_0] - \Pr[S_1]| \leq 2q_G \sqrt{\text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G}) + \text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G})}.
\]

Proof (Proof of Lemma D.1). Let \( F \) be an algorithm described in Figure 12. It is easy to see that Game0 can be restated as

\[
\mu^* \leftarrow D_M; r^* \leftarrow G(\mu^*); \text{inp} \leftarrow F(\mu^*, r^*); b' \leftarrow A_G(\mu^*; \text{inp}); \text{return } b'.
\]

and Game1 can be restated as

\[
\mu^* \leftarrow D_M; r^* \leftarrow \mathcal{R}; \text{inp} \leftarrow F(\mu^*, r^*); b' \leftarrow A_G(\mu^*; \text{inp}); \text{return } b'.
\]

Applying the O2H lemma (Corollary A.1) with \( X = M, Y = \mathcal{R}, D_X = D_M, x = \mu^*, y = r^* \), and algorithms \( \mathcal{A} \) and \( F \), we have

\[
|\Pr[S_0] - \Pr[S_1]| \leq 2q_G \sqrt{\text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G}) + \text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G})},
\]

where \( \mathcal{G}_i \) is the algorithm described in Figure 12. \( (ek, dk) \leftarrow \text{Gen}_0(1^k), \mu^* \in D_M; r^* \leftarrow \mathcal{R}, \text{and } c^* \leftarrow \text{Enc}_0(ek, \mu^*, r^*). \)

We have \( \Pr[\mu^* \leftarrow \mathcal{G}_i(ek, c^*)] \leq \text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G}) \). By combining these inequalities, the lemma is proven.

Lemma D.2. There exists an adversary \( A_{12} \) such that

\[
|\Pr[S_1] - \Pr[S_2]| \leq \text{Adv}_{\text{PKE},D_M,S,\mathcal{A}}^{\text{OW-CPA}}(\mathcal{G}).
\]

Since the proof is obtained straightforwardly, we omit it.

Combining the above two lemmas, we obtain the wanted result.
Open problem: One might wonder whether we could make the above lemma tighter by using the semi-classical O2H lemma [AHU19], the double-sided O2H lemma [BHHT^19], or the MRM O2H lemma [KSS^20]. Essentially speaking, in some game transition, we need to replace $c^* = \text{Enc}(ek, \mu_i; G(\rho_i))$ with $c^* = \text{Enc}(ek, \mu_i^r, r^*)$ with fresh randomness $r^* \leftarrow \mathcal{R}$. This change is an obstacle for tight security.

The existing tight security proof for transform in [BHHT^19] strongly depends on the fact that the goal is onewayness and the adversary finally outputs $T$. The condition allows us to use $\sqrt{P_{\text{left}} - \sqrt{P_{\text{right}}}} \leq 2d\sqrt{P_{\text{guess}}}$ with $P_{\text{guess}} = 0$ in Lemma A.2, which yields $P_{\text{left}} \leq 4d^2 P_{\text{guess}}$.

If we invoke the MRM O2H lemma [KSS^20], we will consider an algorithm $\text{Ext}$ such that

$$\Pr[\text{Ext}(G, c^*) \neq \text{Ext}(\overline{G}, \overline{c^*})] = \delta.$$ 

Notice that, on input $\mu_i^r$, $G'$ is overwritten by $r^*$. We want to connect this probability with the advantage of the OW-CPA/IND-CPA/DS security of PKE\textsubscript{0}, but this seems impossible. The reduction algorithm on input $ek$ and $c^* = \text{Enc}(ek, \mu_i^r, r^*)$ is given an access to a random oracle $G$. In order to implement $G'$ on input $\mu_i^r$, it should know $\mu_i^r$ and $r^*$, which is already the solution of the challenge ciphertext of the security game in the OW-CPA/IND-CPA/DS security. Thus, we cannot use them in the context of $T$ unfortunately.

E Properties of U\textsuperscript{L}

As we see in Figure 10, $U^L$ and $\text{SKY} = U_{m}^L$ are not connected. Indeed, we face a subtle problem to apply the indifferentiable reduction in Bindel et al. [BHHT^19]: Suppose that we have $\mathcal{A}$ against the SPR-CCA security of KEM obtained by $U^L$. In their indifferentiable reduction, they construct $\mathcal{A}_m$ against the SPR-CCA security of KEM obtained by $U_{m}^L$. $\mathcal{A}_m$ given $H_m: M \rightarrow K$ simulates $H: M \times C \rightarrow K$ by

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

Unfortunately, this simulation makes $H(s, c)$ different from $H_{\text{pref}}(s, c)$ at the point $(s, c)$ with $c = \text{Enc}(ek, s)$. We here directly prove the security properties of $U^L$. We give proof sketches, because the proofs are very similar to those of $\text{SKY}$ in Section 4.

E.1 SPR-CCA Security

We can use the proof of the SPR-CCA security of $\text{SKY} = U_{m}^L$ (subsection B.2) with slight modifications. Roughly speaking, we replace $H(s, c)$ with $H_p(c)$ and, then, apply the above indifferentiable reduction. Doing so, we can find the situation is essentially equivalent to Game\textsubscript{1} (or Game\textsubscript{2}) of Table 6.

Theorem E.1 (Case for derandomized PKE). Let PKE\textsubscript{0} be a probabilistic PKE scheme. Let us consider a derandomized PKE scheme PKE = $\mathcal{T}[\text{PKE}, G]$. Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable and $\delta$-correct with negligible $\delta$, then KEM = $U^L[\text{PKE}, H]$ is SPR-CCA-secure.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{\text{Dec}}$ queries to the decapsulation oracle and $q_G$ and $q_{H_1}$ queries to $G$ and $H$ respectively, there exist $\mathcal{A}_3$ against ciphertext-indistinguishability of PKE such that

$$\text{Adv}_{\text{KEM}, S, \mathcal{A}}(x) \leq \text{Adv}_{\text{PKE}, D_M, S, \mathcal{A}_3}(x) + \text{Disj}_{\text{PKE}, S}(x) + 4\delta + 16(q_G + q_{\text{Dec}} + 1)^2\delta + 16(q_G + q_H + 1)^2\delta + 4(q_H + q_{\text{Dec}}) / \sqrt{|M|}.$$ 

Theorem E.2 (Case for non-derandomized PKE). Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable and $\delta$-correct with negligible $\delta$, then KEM = $U^L[\text{PKE}, H]$ is SPR-CCA-secure.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{\text{Dec}}$ queries to the decapsulation oracle and $q_G$ and $q_{H_1}$ queries to $G$ and $H$, respectively, there exist $\mathcal{A}_3$ against ciphertext-indistinguishability of PKE such that

$$\text{Adv}_{\text{KEM}, \mathcal{A}, \mathcal{A}_3}(x) \leq \text{Adv}_{\text{PKE}, D_M, S, \mathcal{A}_3}(x) + \text{Disj}_{\text{PKE}, S}(x) + 4(q_{H_1} + q_{\text{Dec}}) / \sqrt{|M|} + 4\delta.$$ 

Proof of Theorem E.1: We use the game-hopping proof. We consider Game\textsubscript{i} for $i = 0, \ldots, 8$. We summarize the games in Table 7. Let $S_i$ denote the event that the adversary outputs $b' = 1$ in game Game\textsubscript{i}. Let Acc and $\overline{\text{Acc}}$ denote the event that the key pair $(ek, dk)$ is accurate and inaccurate, respectively.
Table 7. Summary of Games for the Proof of Theorem E.1. We define $g(\mu) = \text{Enc}(ek, \mu) = \text{Enc}_0(ek, \mu; G(\mu))$.

<table>
<thead>
<tr>
<th>Game</th>
<th>H</th>
<th>G</th>
<th>$c^*$</th>
<th>$K^*$</th>
<th>Decryption valid $c$</th>
<th>invalid $c$</th>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game0</td>
<td>H</td>
<td>$\mathcal{F}(M, R)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$(s, c)$</td>
<td>Lemma 2.2</td>
</tr>
<tr>
<td>Game1</td>
<td>H</td>
<td>$\mathcal{F}(M, R)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$(s, c)$</td>
<td>Lemma 2.2 + correctness</td>
</tr>
<tr>
<td>Game1.1</td>
<td>H</td>
<td>$\mathcal{F}_e(ek, dk)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game1.2</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}_e(ek, dk)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>if a key pair is good</td>
</tr>
<tr>
<td>Game2</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}_e(ek, dk)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>if a key pair is good</td>
</tr>
<tr>
<td>Game3</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}_e(ek, dk)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$_g(c)$</td>
<td>H$_g(c)$</td>
<td>conceptual</td>
</tr>
<tr>
<td>Game3.1</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}(M, R)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$_g(c^*)$</td>
<td>H$_g(c)$</td>
<td>H$_g(c)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game4</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}(M, R)$</td>
<td>$S(1^k)$</td>
<td>H$_g(c^*)$</td>
<td>H$_g(c)$</td>
<td>H$_g(c)$</td>
<td>ciphertext indistinguishability</td>
</tr>
<tr>
<td>Game5</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}(M, R)$</td>
<td>$S(1^k)$</td>
<td>U$(\mathcal{K})$</td>
<td>H$_g(c)$</td>
<td>H$_g(c)$</td>
<td>statistical disjointness</td>
</tr>
<tr>
<td>Game6</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}_e(ek, dk)$</td>
<td>Enc$(ek, \mu')$</td>
<td>H$(\mu', c^*)$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game7</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}(M, R)$</td>
<td>$S(1^k)$</td>
<td>U$(\mathcal{K})$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>if a key pair is good</td>
</tr>
<tr>
<td>Game8</td>
<td>H$_g$ o g / H'</td>
<td>$\mathcal{F}(M, R)$</td>
<td>$S(1^k)$</td>
<td>U$(\mathcal{K})$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>if a key pair is good</td>
</tr>
<tr>
<td>Game9</td>
<td>H</td>
<td>$\mathcal{F}(M, R)$</td>
<td>$S(1^k)$</td>
<td>U$(\mathcal{K})$</td>
<td>H$(\mu, c)$</td>
<td>H$_g(c)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
</tbody>
</table>

Game5: This game is the original game $\text{Exp}^{\text{spr-cca}}_{\text{KEM}, \mathcal{A}}(x)$ with $b = 0$. Thus, we have

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

Game1: This game is the same as Game0 except that $H(s, c)$ in the decapsulation oracle is replace with $H_g(c)$ where $H_g : \mathcal{C} \rightarrow \mathcal{K}$ is another random oracle. We remark that $\mathcal{A}$ is not given direct access to $H_g$.

As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound

$$|\Pr[S_0] - \Pr[S_1]| \leq 2(q_H + q_{\text{Dec}})/\sqrt{|M|},$$

where $q_H$ and $q_{\text{Dec}}$ denote the number of queries to $H$ and $\text{Dec}$ the adversary makes, respectively.

Game1.1: This game is the same as Game1 except that the random oracle $G(\cdot)$ is chosen from $\mathcal{F}_e(ek, dk)$ instead of $\mathcal{F}(M, R)$. See subsection B.2 for the definitions of $\mathcal{F}_e(ek, dk)$, Bad, and Good. As in the argument in subsection B.2, we obtain

$$|\Pr[S_1] - \Pr[S_{1,1}]| \leq 8(q_G + q_{\text{Dec}} + 1)^2 \delta.$$

In addition, We have $\Pr[\text{Bad}] \leq \delta$ ([LW21, Claim 3]). According to Lemma A.1, for any $p \geq 0$, we also have

$$|\Pr[S_{1,1}] - p| \leq |\Pr[S_{1,1} \wedge \text{Good}] - p| + \delta.$$

Game1.2: This game is the same as Game1.1 except that the random oracle $H(\cdot, \cdot)$ is simulated as follows: Let $H'_g : \mathcal{C} \rightarrow \mathcal{K}$ and $H' : M \times \mathcal{C} \rightarrow \mathcal{K}$ be random oracles. Define

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

We remark that the decapsulation oracle and the generation of $K^*$ also use this simulation.

If Good occurs, then $\text{PKE} = \{\text{PK}_{\text{K0}}, G\}$ is perfectly correct from the definition of $G$ and $g(\mu) = \text{Enc}(ek, \mu; G(\mu))$ is injective. Thus, $H'_g o g : M \rightarrow \mathcal{K}$ is a random function and the two games Game1.1 and Game1.2 are equivalent if Bad does not occur. We have

$$\Pr[S_{1,1} \wedge \text{Good}] = \Pr[S_{1,2} \wedge \text{Good}].$$
Game$_2$: This game is the same as Game$_{1,2}$ except that the random oracle $H$ is simulated by $H_q = g$ and $H'$ instead of $H'_q = g$ and $H'$. If Good occurs, then PKE = $T[$PKE, $G]$ is perfectly correct from the definition of $G$. Hence, the two games Game$_{1,2}$ and Game$_2$ are equivalent, because a value of $H'_q(c)$ for an invalid $c$ is not used in Game$_{1,2}$. We have
\[ \Pr[S_{1,2} \wedge \text{Good}] = \Pr[S_2 \wedge \text{Good}] \]

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 be denoted by Dec$^*$. If Good occurs, then PKE = $T[$PKE, $G]$ is perfectly correct from the definition of $G$. If so, the two games Game$_2$ and Game$_3$ are equivalent, and we have
\[ \Pr[S_2 \wedge \text{Good}] = \Pr[S_3 \wedge \text{Good}] \]

According to Lemma A.1, for any $p \geq 0$, we have
\[ |\Pr[S_3 \wedge \text{Good}] - p| \leq |\Pr[S_3] - p| + \delta. \]

Game$_{1,1}$: This game is the same as Game$_3$ except that $G$ is chosen from $\mathcal{F}(M, \mathcal{R})$ instead of $\mathcal{F}_{\text{good}}^{ek, dk}(M, \mathcal{R})$. As in the argument in subsection B.2, we obtain
\[ |\Pr[S_3] - \Pr[S_{1,1}]| \leq 8(q_G + q_H + 1)^2 \delta. \]
(We note that $H$ and the challenge ciphertext also query to $G$ internally.)

Game$_4$: This game is the same as Game$_3$ except that $c^*$ is generated by $S(1^*)$. The difference between two games Game$_3$ and Game$_4$ is bounded by the advantage of ciphertext indistinguishability. We have
\[ |\Pr[S_3] - \Pr[S_4]| \leq \text{Adv}_{\text{PKE, D}_{\mathcal{M}, \mathcal{S}}, \mathcal{R}_3}^{\text{ds-ind}}(\kappa). \]

Game$_5$: This game is the same as Game$_4$ except that $K^* \leftarrow K$ instead of $K^* \leftarrow H_q(c^*)$. In Game$_4$, if $c^* \leftarrow S(1^*)$ is not in 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, Lemma 4.8]. We have
\[ |\Pr[S_4] - \Pr[S_5]| \leq \text{Disj}_{\text{PKE, S}}^{\text{ind}}(\kappa). \]

Game$_{5,1}$: This game is the same as Game$_5$ except that $G$ is chosen from $\mathcal{F}_{\text{good}}^{ek, dk}(M, \mathcal{R})$ instead of $\mathcal{F}(M, \mathcal{R})$. As in the argument in subsection B.2, we obtain
\[ |\Pr[S_5] - \Pr[S_{5,1}]| \leq 8(q_G + q_H + 1)^2 \delta. \]
(We note that $H$ and the challenge ciphertext also query to $G$ internally.) According to Lemma A.1, for any $p \geq 0$, we have
\[ |\Pr[S_{5,1} \wedge \text{Good}] - p| \leq |\Pr[S_{5,1}] - p| + \delta. \]

Game$_6$: This game is the same as Game$_5$ except that the decapsulation oracle is reset as Dec$^*$. Similar to the case for Game$_2$ and Game$_3$, if a key pair is good, the two games Game$_5$ and Game$_6$ are equivalent as in the proof of [XY19, Lemma 4.5]. We have
\[ \Pr[S_{5,1} \wedge \text{Good}] = \Pr[S_6 \wedge \text{Good}] \]

Game$_{6,1}$: This game is the same as Game$_6$ except that the random oracle $H$ is simulated by $H'_q = g$ and $H'$ as in Game$_{1,2}$. If Good occurs, the two games Game$_6$ and Game$_{6,1}$ are equivalent. We have
\[ \Pr[S_6 \wedge \text{Good}] = \Pr[S_{6,1} \wedge \text{Good}] \]

Game$_{6,2}$: This game is the same as Game$_{6,1}$ except that the random oracle $H(\cdot)$ is set as the original. If Good occurs, the two games Game$_{6,1}$ and Game$_{6,2}$ are equivalent. We have
\[ \Pr[S_{6,1} \wedge \text{Good}] = \Pr[S_{6,2} \wedge \text{Good}] \]

We also have, for any $p \geq 0$,
\[ |\Pr[S_{6,2} \wedge \text{Good}] - p| \leq |\Pr[S_{6,2}] - p| + \delta \]
from Lemma A.1.
Table 8. Summary of Games for the Proof of Theorem E.3: \( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) \) implies that the challenger generates \( c^* \leftarrow \mathcal{S}(1^*) \) and returns \( \bot \) if \( c^* \in \text{Enc}(ek, M) \).

<table>
<thead>
<tr>
<th>Game</th>
<th>( H )</th>
<th>( c^* )</th>
<th>( K^* )</th>
<th>Decryption justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{Game}_0 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) )</td>
<td>random</td>
<td>( H(\mu, c) ) ( H(s, c) )</td>
</tr>
<tr>
<td>( \text{Game}_1 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>random</td>
<td>( H(\mu, c) ) ( H(s, c) )</td>
</tr>
<tr>
<td>( \text{Game}_2 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>random</td>
<td>( H(\mu, c) ) ( H(q) )</td>
</tr>
<tr>
<td>( \text{Game}_3 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>( H(q)(c^*) )</td>
<td>( H(\mu, c) ) ( H(q) )</td>
</tr>
<tr>
<td>( \text{Game}_4 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>( H(s, c^*) )</td>
<td>( H(\mu, c) ) ( H(s, c) )</td>
</tr>
<tr>
<td>( \text{Game}_5 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>( \text{Dec}(dk, c^*) )</td>
<td>( H(\mu, c) ) ( H(s, c^*) )</td>
</tr>
<tr>
<td>( \text{Game}_6 )</td>
<td>( H )</td>
<td>( \mathcal{S}(1^*) \setminus \text{Enc}(ek, M) )</td>
<td>( \text{Dec}(dk, c^*) )</td>
<td>( H(\mu, c) ) ( H(s, c) )</td>
</tr>
</tbody>
</table>

**Game**\( _0 \): This game is the same as Game\( _0\), except that the random oracle \( G \) is chosen from \( \mathcal{T}(M, \mathcal{R}) \) instead of \( \mathcal{T}_{ek,dk}(M, \mathcal{R}) \). As in the argument in subsection B.2, we have,

\[
| \Pr[S_0] - \Pr[S_2] | \leq 8(q_{G} + q_{\text{Dec}})^2. \\
\]

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

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

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

\[
\text{Adv}^{\text{spr-cca}}_{\text{KEM}, \mathcal{A}}(k) = | \Pr[S_0] - \Pr[S_2] | \\
\leq \text{Adv}^{\text{ind}}_{\mathcal{K}, \mathcal{D}, \mathcal{A}}(k) + \text{Disj}_{\text{KEM}, \mathcal{S}}(k) + 4\delta \\\n+ 16(q_{G} + q_{\text{Dec}} + 1)^2. \\
\]

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

\[
\text{Adv}^{\text{spr-cca}}_{\text{KEM}, \mathcal{S}, \mathcal{A}}(k) = | \Pr[S_0] - \Pr[S_2] | \\
\leq 4(q_{H} + q_{\text{Dec}})/\sqrt{|M|} + 4\delta + \text{Adv}^{\text{ind}}_{\mathcal{K}, \mathcal{D}, \mathcal{A}}(k) + \text{Disj}_{\text{KEM}, \mathcal{S}}(k).
\]

**E.2 SSMT-CCA Security**

We can show the SSMT-CCA security of \( U_{K} \) by using the essentially same proof of that for SYX.

**Theorem E.3.** Suppose that a ciphertext space \( C \) of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then KEM = \( U_{K} \) [PKE, \( H \)] is SSMT-CCA-secure.

Formally speaking, for any adversary \( \mathcal{A} \) against SSMT-CCA security of KEM, we have

\[
\text{Adv}^{\text{ssmt-cca}}_{\text{KEM}, \mathcal{S}, \mathcal{A}}(k) \leq 2\text{Disj}_{\text{KEM}, \mathcal{S}}(k) + 4(q_{H} + q_{\text{Dec}})/\sqrt{|M|}.
\]

Note that this security proof is independent of that PKE is deterministic PKE or one derandomized by \( T \).

**Proof Sketch:** We use the game-hopping proof. We consider Game\( _i \) for \( i = 0, \ldots, 6 \). We summarize the games in Table 8. Let \( S_i \) denote the event that the adversary outputs \( b' = 1 \) in game Game\( _i \). Let 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 $\text{Exp}^\text{ssmt-cca}_{\text{KEM}, S, \mathcal{A}}(\kappa)$ with $b = 0$. The challenge is generated as 

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

We have 

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

Game$_1$: In this game, the ciphertext is set as the QROM. The difference between two games Game$_0$ and Game$_1$ is bounded by statistical disjointness.

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

Game$_2$: This game is the same as Game$_1$ except that $H(s, c)$ in the decapsulation oracle is replace with $H_p(c)$ where $H_p: C \rightarrow K$ is another random oracle.

As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound 

$$|\Pr[S_1] - \Pr[S_2]| \leq 2(q_{\text{H}} + q_{\text{Dec}})/\sqrt{|M|},$$

where $q_{\text{H}}$ denote the number of queries to $H_{\text{prf}}$ the adversary makes.

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

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

Game$_4$: This game is the same as Game$_3$ except that $H_p(\cdot)$ is replaced by $H(s, \cdot)$. As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound 

$$|\Pr[S_3] - \Pr[S_4]| \leq 2(q_{\text{H}} + q_{\text{Dec}})/\sqrt{|M|},$$

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

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

Game$_6$: We finally replace how to compute $c^*$. In this game, the ciphertext is chosen by $S(1^k)$ as in Game$_0$. The difference between two games Game$_5$ and Game$_6$ is bounded by statistical disjointness.

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

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

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

Summing the (in)equalities, we obtain Theorem E.3:

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

E.3 SCFR-CCA Security

**Theorem E.4.** If PKE is SCFR-CCA-secure (or XCFR-secure), then KEM = $U^\perp [\text{PKE}, H]$ is SCFR-CCA-secure in the QROM.

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

**Proof.** Suppose that an adversary outputs a ciphertext $c$ which is decapsulated into $K \neq \bot$ by both $\overline{\mathcal{D}_{\text{K}}}$ and $\overline{\mathcal{D}_{t}}$, that is, $\text{Dec}(\overline{\mathcal{D}_{\text{K}}}, c) = \text{Dec}(\overline{\mathcal{D}_{t}}, c)$. Let us define $\mu_i' = \text{Dec}(d_{k_i}, c)$ for $i \in \{0, 1\}$. We also define $\mu_i := \mu_i'$ if $c = \text{Enc}(e_k, \mu_i')$ and $\bot$ otherwise.

We have five cases defined as follows:

1. Case 1 $(\mu_0 = \mu_1 \neq \bot)$: This violates the SCFR-CCA security (or the XCFR security) of the underlying PKE and it is easy to make a reduction.
2. Case 2 ($\perp \neq \mu_0 \neq \mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = H(\mu_0, c) = H(\mu_1, c)$. Thus, we succeed to find a collision for $H$, which is negligible for any QPT adversary (Lemma 2.3).

3. Case 3 ($\mu_0 = \perp$ and $\mu_1 \neq \perp$): In this case, the decapsulation algorithm outputs $K = H(s_0, c) = H(\mu_1, c)$. Notice that we can replace $H(s_0, \cdot)$ with $H_q(\cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $(c, (\mu_1, c))$ between $H_q$ and $H$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).

4. Case 4 ($\mu_0 \neq \perp$ and $\mu_1 = \perp$): In this case, the decapsulation algorithm outputs $K = H(\mu_0, c) = H(s_1, c)$. Again, we can replace $H(\mu_0, \cdot)$ with $H_q(\cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((\mu_0, c), c)$ between $H$ and $H_q$. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).

5. Case 5 (The other cases): In this case, we find a collision $((s_0, c), (s_1, c))$ of $H$, which is indeed collision if $s_0 \neq s_1$ which occurs with probability at least $1 - 1/2^t$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

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

F Properties of $\text{HU}^\perp_m$

In this section, we review $\text{HU}^\perp_m$ [JZM19], which allows explicit rejection by using the additional 'key-confirmation' hash. Since $\text{HU}^\perp_m$ is KEM with explicit rejection, we only consider the SPR-CCA security and smoothness.

Let $\text{PKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let $C$ and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $H: \mathcal{M} \rightarrow \mathcal{K}$ and $F: \mathcal{M} \rightarrow \mathcal{H}$ be hash functions modeled by random oracles. $\text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{HU}^\perp_m[\text{PKE}, H, F]$ obtained by using $\text{HU}^\perp_m$ is defined as in Figure 13.

![Figure 13. KEM = HU^\perp_m[PKE, H, F]](image)

F.1 SPR-CCA Security

Theorem F.1 (Case of derandomized PKE). Let $\text{PKE} = \mathcal{T}[\text{PKE}_0, G]$. Suppose that a ciphertext space $C$ of PKE depends on the public parameter only if PKE is strongly disjoint-simulatable with simulator $S$ and $\delta$-correct with negligible $\delta$, then $\text{KEM} = \text{HU}^\perp_m[\text{PKE}, H, F]$ is SPR-CCA-secure, where we use a new simulator $S’ = S \times U(\mathcal{H})$.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{\text{Dec}}$ queries to the decapsulation oracle and $q_f, q_G$ and $q_{\text{H1}}$ queries to $F$ and $H$, respectively, there exists $\mathcal{A}_{\text{S4}}$ against ciphertext-indistinguishability of PKE such that

$$
\text{Adv}_{\text{KEM}, \mathcal{S}, \mathcal{A}}^{\text{spr-CCA}}(k) \leq \text{Adv}_{\text{PKE}, 2\mathcal{M}, S, \mathcal{A}_{\text{S4}}}(k) + 16(q_G + q_{\text{Dec}} + 1)^2 \delta + 4\delta + 8(q_G + q_{\text{H1}} + q_f + 1)^2 \delta + 8(q_G + q_{\text{H1}} + q_f + q_{\text{Dec}} + 1)^2 \delta + (4q_{\text{Dec}} + 1)/|\mathcal{H}|.
$$

Theorem F.2 (Case of non-derandomized PKE). Suppose that a ciphertext space $C$ of PKE depends on the public parameter only if PKE is strongly disjoint-simulatable with simulator $S$ and $\delta$-correct with negligible $\delta$, then $\text{KEM} = \text{HU}^\perp_m[\text{PKE}, H, F]$ is SPR-CCA-secure, where we use a new simulator $S’ = S \times U(\mathcal{H})$.

Formally speaking, for any $\mathcal{A}$ against the SPR-CCA security of KEM issuing at most $q_{\text{Dec}}$ queries to the decapsulation oracle and $q_f$ and $q_{\text{H1}}$ queries to $F$ and $H$, respectively, there exists $\mathcal{A}_{\text{S4}}$ against ciphertext-indistinguishability of PKE such that

$$
\text{Adv}_{\text{KEM}, \mathcal{S}, \mathcal{A}}^{\text{spr-CCA}}(k) \leq \text{Adv}_{\text{PKE}, 2\mathcal{M}, S, \mathcal{A}_{\text{S4}}}(k) + 2\text{Disj}_{\text{PKE}, \mathcal{S}}(k) + 4\delta + (4q_{\text{Dec}} + 1)/|\mathcal{H}|.
$$

38
Table 9. Summary of Games for the Proof of Theorem F.1. We define $g(\mu) = \text{Enc}(ek, \mu) = \text{Enc}_0(ek, \mu; G(\mu))$.

<table>
<thead>
<tr>
<th>Game</th>
<th>$H$</th>
<th>$F$</th>
<th>$G$</th>
<th>$c_0^2$</th>
<th>$c_1^2$</th>
<th>$K^*$</th>
<th>Decapsulation $\mathcal{K}$ condition</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game0</td>
<td>$H$</td>
<td>$F$</td>
<td>$\mathcal{T}(M, \mathcal{R})$</td>
<td>$\text{Enc}(ek, \mu')$</td>
<td>$\text{F}(\mu')$</td>
<td>$H(\mu')$</td>
<td>$H(\mu)$ if $c_0 = \text{Enc}(ek, \mu)$ and $c_1 = F(\mu)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game1</td>
<td>$H$</td>
<td>$F$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$\text{Enc}(ek, \mu')$</td>
<td>$\text{F}(\mu')$</td>
<td>$H(\mu')$</td>
<td>$H(\mu)$ if $c_0 = \text{Enc}(ek, \mu)$ and $c_1 = F(\mu)$ if key is not bad</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game2</td>
<td>$H$</td>
<td>$\text{H}_0 \circ g \circ F_q \circ g$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$\text{Enc}(ek, \mu')$</td>
<td>$F_q(c_0^2)$</td>
<td>$H_q(c_0^2)$</td>
<td>$H_q(c_0)$ if $c_0 = \text{Enc}(ek, \mu)$ and $c_1 = F_q(c_0)$</td>
<td>statistical</td>
</tr>
<tr>
<td>Game3</td>
<td>$H$</td>
<td>$\text{H}_0 \circ g \circ F_q \circ g$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$\text{Enc}(ek, \mu')$</td>
<td>$F_q(c_0^2)$</td>
<td>$H_q(c_0^2)$</td>
<td>$H_q(c_0)$ if $c_1 = F_q(c_0)$</td>
<td>Lemma 2.1 + correctness</td>
</tr>
<tr>
<td>Game4</td>
<td>$H$</td>
<td>$\text{H}_0 \circ g \circ F_q \circ g$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$S(\tau^*)$</td>
<td>$U(\mathcal{H})$</td>
<td>$U(\mathcal{K})$</td>
<td>$H_q(c_0)$ if $c_1 = F_q(c_0)$ if $c_1 = F_q(c_0)$</td>
<td>statistical</td>
</tr>
<tr>
<td>Game5</td>
<td>$H$</td>
<td>$\text{H}_0 \circ g \circ F_q \circ g$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$S(\tau^*)$</td>
<td>$U(\mathcal{H})$</td>
<td>$U(\mathcal{K})$</td>
<td>$H(\mu)$ if $c_0 = \text{Enc}(ek, \mu)$ and $c_1 = F(\mu)$</td>
<td>conceptual change</td>
</tr>
<tr>
<td>Game6</td>
<td>$H$</td>
<td>$\text{H}_0 \circ g \circ F_q \circ g$</td>
<td>$\mathcal{T}_{\text{good}}^{\text{dec}}(M, \mathcal{R})$</td>
<td>$S(\tau^*)$</td>
<td>$U(\mathcal{H})$</td>
<td>$U(\mathcal{K})$</td>
<td>$H(\mu)$ if $c_0 = \text{Enc}(ek, \mu)$ and $c_1 = F(\mu)$ if key is not bad</td>
<td>Lemma 2.1 + correctness</td>
</tr>
</tbody>
</table>

Proof of Theorem F.1: We use the game-hopping proof. We consider Game$_i$ for $i = 0, \ldots, 8$. We summarize the games in Table 9. Let $\delta_i$ denote the event that the adversary outputs $b' = 1$ in game Game$_i$.

We mainly follow the security proof in [JZM19, XY19, LW21], while we use a new simulator $S' = S \times U(\mathcal{H})$ instead of $S' = \text{Enc}(ek, M) \times U(\mathcal{H})$.

Game$_0$: This game is the original game $\text{Exp}^{\text{spr-cca}}_{\text{KEM, } \mathcal{A}}(x)$ with $b = 0$. By the definition, we have

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

Game$_1$: This game is the same as Game$_0$ except that the random oracle $G$ is chosen from $\mathcal{G}_{\text{good}, \text{ek}, \text{dk}}^{\text{dec}}(M, \mathcal{R})$ instead of $\mathcal{G}(M, \mathcal{R})$, where $\mathcal{G}_{\text{good}, \text{ek}, \text{dk}}^{\text{dec}}(M, \mathcal{R})$ is a set of functions $G : M \rightarrow \mathcal{R}$ satisfying $G(\mu) \in \mathcal{R}_{\text{good}, \text{ek}, \text{dk}, \mu}$ for all $\mu \in M$ with $\mathcal{R}_{\text{good}, \text{ek}, \text{dk}, \mu} := \{r \in \mathcal{R} : \text{Dec}(dk, \text{Enc}(ek, \mu, r)) = \mu\}$. We define $\text{Bad} := \text{boole}(\exists \mu \in M : \mathcal{R}_{\text{good}, \text{ek}, \text{dk}, \mu} = \emptyset)$ and $\text{Good} := \neg \text{Bad}$.

As in the proof of Theorem 4.1 in subsection B.2, we have

$$\left| \Pr[S_0] - \Pr[S_{0,1}] \right| \leq 8(q_G + q_{\text{Dec}} + 1)^2 \delta.$$

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

Game$_2$: This game is the same as Game$_0$ except that the random oracles $H$ and $F$ are simulated by $H_q \circ g$ and $F_q \circ g$, respectively, where $H_q : C \rightarrow \mathcal{K}$ and $F_q : C \rightarrow \mathcal{H}$ are random oracles and $g(\mu) := \text{Enc}(ek, \mu)$. If $G$ occurs, then $H_q \circ g$ and $F_q \circ g$ are random functions and those two games are equal to each other. We have

$$\Pr[S_{0,1} \wedge \text{Good}] = \Pr[S_1 \wedge \text{Good}].$$

Game$_3$: This game is the same as Game$_1$ except that the decapsulation oracle internally computes $c_1$ as $F_q(c_0)$ instead of $F(\mu')$ and $K = H_q(c_0)$ instead of $H(\mu')$, where $\mu' = \text{Dec}(dk, c_0)$, that is, we rewrite the line 2 of Dec with "if $\mu' = \bot$ or $c_0 \neq \text{Enc}(ek, \mu')$ or $c_1 \neq F(\mu')"$ and the line 4 of Dec with "else return $K := H_q(c_0)".

If the two conditions $\mu' \neq \bot$ and $c_0 = \text{Enc}(ek, \mu')$ are satisfied, then the former change is just conceptual since we set $F = F_q \circ g$ in the previous game and we have $F_q(c_0) = F_q(\text{Enc}(ek, \mu')) = (F_q \circ g)(\mu')$. The latter change is also conceptual since we set $H = H_q \circ g$ in the previous game and we have $H_q(c_0) = H_q(\text{Enc}(ek, \mu')) = (H_q \circ g)(\mu')$. Thus, we have

$$\Pr[S_1 \wedge \text{Good}] = \Pr[S_2 \wedge \text{Good}].$$

Game$_4$: In this game the decapsulation oracle ignores the condition "$\mu' = \bot$ or $c_0 \neq \text{Enc}(ek, \mu')"$ that is, we rewrite the line 2 of Dec with "if $c_1 \neq F(\mu')."$ By this modification, when $(c_0, c_1) \neq (c_0^*, c_1^*)$, the oracle returns $K = H_q(c_0) |_{c_1} = F_q(c_0)$. Let us consider the following three cases:
As in the proof of Theorem 4.1 in subsection B.2, we have

Assuming that

Dec

Thus, the difference occurs when c₀ is outside of Enc(ek, M) and c₁ = F_q(c₀). Notice that the adversary cannot access such hash values F_q(C \ \{ek, M\}) directly, since it is given F instead of F_q. Therefore, any c₁ hits the value F_q(c₀) with probability at most 1/|H| and we obtain the bound q_{Dec}/|H|. (If a decapsulation query is quantum, we will get another bound 2q_{Dec}/\sqrt{|H|}. ) We have

\[|Pr[S_2 \land Good] − Pr[S_3 \land Good]| ≤ q_{Dec}/|H|.\]

We also have for any p ≥ 0,

\[|Pr[S_3 \land Good] − p| ≤ |Pr[S_3] − p| + δ.\]

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

\[|Pr[S_1] − Pr[S_{3.1}]| ≤ 8(q_G + q_H + q_F + q_{Dec} + 1)^2δ\]

as in the proof of Theorem 4.1 in subsection B.2. (We note that H, F, Dec, and the challenge ciphertext also query to G internally.)

**Game_{1}**: We replace \( c'_0 := Enc(ek, \mu^*; G(\mu^*)) \) with \( c'_0 \leftarrow S(1^κ) \). The difference is bounded by the advantage of ciphertext indistinguishability and we have a quantum adversary \( \mathcal{A}_{34} \) satisfying

\[|Pr[S_{3.1}] − Pr[S_4]| ≤ Adv_{\text{PKE}}^{\text{ds-ind}}(\mathcal{A}_{34}, S_4, \mathcal{A}_{34})(κ).\]

We omit the detail of the reduction algorithm since it is easy to construct.

**Game_{3}**: This game is the same as Game_4 except that \( K^* \leftarrow \mathcal{K} \) instead of \( K^* \leftarrow \mathcal{H}_q(c'_0) \).

We note that the adversary cannot access to \( K^* = \mathcal{H}_q(c'_0) \) via \( H \) if \( c'_0 \) is outside of Enc(ek, M) in both games. Let \((c_0, c_1)\) be a query to Dec the adversary makes. If \( c_0 = c'_0 \) and \( c_1 = c'_1 \), then the adversary receives \( \perp \) in both games. If \( c_0 = c'_0 \) and \( c_1 \neq c'_1 \), then \( c_1 = F_q(c'_0) = c'_1 \) holds and the adversary receives \( \perp \) in both games. Thus, if \( c'_0 \) is outside of Enc(ek, M), the two games are equal to each other. Hence, the difference is bounded by the statistical disjointness in disjoint simulatability. We have

\[|Pr[S_4] − Pr[S_3]| ≤ \text{Disj}_{\text{PKE}, S}(κ).\]

**Game_{5.1}**: This game is the same as Game_5.1 except that \( c'_1 \leftarrow U(H) \) instead of \( c'_1 := \mathcal{H}_q(c'_0) \), in which our proof is different from that of Jiang et al. [JZM19].

When the adversary queries \((c_0, c_1)\) for \( c_0 \neq c'_0 \), there is no leak on \( \mathcal{H}_q(c'_0) \). In addition, when \( c'_0 \) is the outside of Enc(ek, M), the adversary cannot obtain the real hash value \( c'_1 = F_q(c'_0) \) directly.

Suppose that \( c'_0 \) is the outside of Enc(ek, M). We consider the case that the adversary queries \((c'_0, c_1)\) for Dec.

- In Game_5, we have \( c'_1 = F_q(c'_0) \). If \( c_1 = c'_1 \), then the adversary receives \( \perp \); otherwise, that is, if \( c_1 \neq c'_1 \), it also receives \( \perp \).
- In Game_{5.1}, we have \( c'_1 \leftarrow U(H) \).
  - If \( c'_1 = F_q(c'_0) \), then this game is the same as Game_5.
  - Suppose that \( c'_1 \neq F_q(c'_0) \). If \( c_1 = c'_1 \), then the adversary receives \( \perp \); otherwise, it receives \( \perp \) if and only if \( c_1 \neq F_q(c'_0) \); it receives \( K = \mathcal{H}_q(c'_0) \) if \( c_1 = F_q(c'_0) \).

Assuming that \( c'_0 \) is the outside of Enc(ek, M) and \( c'_1 \neq F_q(c'_0) \), a value \( c_1 \) hits \( F_q(c'_0) \) with probability at most 1/(|H| − 1). Thus, we have

\[|Pr[S_5] − Pr[S_{5.1}]| ≤ \text{Disj}_{\text{PKE}, S}(κ) + 1/|H| + q_{Dec}/(|H| − 1).\]

**Game_{5.2}**: This game is the same as Game_{5.1} except that G is chosen from \( \mathcal{P}_{\text{good}}^{\text{dec}}(M, R) \).

As in the proof of Theorem 4.1 in subsection B.2, we have

\[|Pr[S_5] − Pr[S_{5.2}]| ≤ 8(q_G + q_H + q_F + 1)^2δ.\]

We also have, for any \( p \geq 0 \),

\[|Pr[S_{5.2}] − p| ≤ |Pr[S_{5.2} \land Good] − p| + δ.\]
Game\textsubscript{5} \textbf{:} This game is the same as Game\textsubscript{5,2} except that the decapsulation algorithm checks if \( c_0 = \text{Enc}(ek, \mu) \) and \( c_1 = \text{Enc}(c_0) \) or not.

As in the argument for the difference between Game\textsubscript{2} and Game\textsubscript{3}, we consider the following three cases for a decapsulation query \((c_0, c_1)\):

- **Case 1:** \((c_0 = \text{Enc}(ek, \mu))\) for some \( \mu \): In this case, the answers of the decapsulation oracles in both games are equal to each other.

- **Case 2:** \((c_0 \not= \text{Enc}(ek, M))\) and \( c_1 \not= F_q(c_0)\): In this case, the answers of the decapsulation oracles in both games are \(\perp\).

- **Case 3:** \((c_0 \not= \text{Enc}(ek, M))\) and \( c_1 = F_q(c_0)\): In this case, the answer in Game\textsubscript{5,2} is \( K = H_q(c) \), but the answer in Game\textsubscript{6} is \(\perp\). This is outside of Enc(ek, M) and we get another bound \(2q_{\text{dec}}(|H|)^{-1/2}\).

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

We have
\[
|\Pr[S_5,2 \land \text{Good}] - \Pr[S_6 \land \text{Good}]| \leq q_{\text{dec}}/|H|.
\]

Game\textsubscript{6} \textbf{:} This game is the same as Game\textsubscript{6} except that the decapsulation oracle use \( H \) and \( F \) instead of \( H_q \) and \( F_q \), respectively. As in the argument for Game\textsubscript{1} and Game\textsubscript{2}, if the key pair is good, then this is the conceptual change and we have
\[
\Pr[S_6 \land \text{Good}] = \Pr[S_7 \land \text{Good}].
\]

We also have, for any \( p \geq 0 \),
\[
|\Pr[S_{7,1} \land \text{Good}] - p| \leq |\Pr[S_{7,1}] - p| + \delta.
\]

Game\textsubscript{7} \textbf{:} This game is the same as Game\textsubscript{7,1} except that the random oracle \( G \) is chosen from \( \mathcal{F}(M, R) \). As in the argument for Game\textsubscript{6} and Game\textsubscript{0,1}, we have
\[
|\Pr[S_{7,1}] - \Pr[S_8]| \leq 8(q_G + q_{\text{dec}} + 1)^2 \delta.
\]

We note that this game is the original game \( \text{Exp}_{\text{KEM}, A}^{\text{spr-cca}}(\kappa) \) with \( b = 1 \). We have
\[
\Pr[S_8] = \Pr[\text{Exp}_{\text{KEM}, A}^{\text{spr-cca}}(\kappa) = 1 | b = 1].
\]

**Summary:** Summing those (in)equalities, we obtain the following bound:
\[
\text{Adv}_{\text{KEM}, A}^{\text{spr-cca}}(\kappa) = |\Pr[S_8] - \Pr[S_8]|
\leq \text{Adv}_{\text{KEM}, A}^{\text{spr-cca}}(\kappa) + 2\text{Disj}_{\text{KEM}, A}(\kappa) + 16(q_G + q_{\text{dec}} + 1)^2 \delta + 4\delta
+ 8(q_G + q_H + q_f)^2 \delta + 8(q_G + q_H + q_f + q_{\text{dec}} + 1)^2 \delta
+ (2q_{\text{dec}} + 1)/|H| + q_{\text{dec}}/(|H| - 1)
\]

and we replace \((2q_{\text{dec}} + 1)/|H| + q_{\text{dec}}/(|H| - 1))\) with \((4q_{\text{dec}} + 1)/|H|\).

**F.2 Sparseness**

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

**Proof.** Let us consider \((c_0, c_1) \leftarrow \mathcal{S}(1^k) \times U(H)\). If \( c_0 \) is decrypted into \( \mu' \not= \perp \), then \( c_1 = F(\mu') \) with probability at most \(1/|H|\). Thus, \( \text{KEM} \) is \(1/|H|\)-sparse. \(\square\)
G Properties of HU\(^\perp\)

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

G.1 SPR-CCA security:

In order to show the SPR-CCA security of HU\(^\perp\), we consider the following theorem on indifferentiable reduction, which is obtained by mimicking that for \(U_m^x \leftrightarrow U^x\) in [BHHT19, Theorem 5].

**Theorem G.1** (HU\(_{m}^x \leftrightarrow HU^\perp\)). Let PKE be a deterministic PKE. Let KEM\(_{m} = HU_{m}^\perp\)[PKE, H\(_m\), F\(_m\)] and KEM = HU\(^\perp\)[PKE, H, F].

1. If KEM\(_{m}\) is SPR-CCA-secure, then KEM is SPR-CCA-secure also.
2. If KEM is SPR-CCA-secure, then KEM\(_{m}\) is SPR-CCA-secure also.

**Proof (The first part).** Suppose that we have an adversary \(\mathcal{A}\) against the SPR-CCA security of KEM. We construct an adversary \(\mathcal{A}_{m}\) against the SPR-CCA security of KEM\(_{m}\) with random oracle \(H_{m}: \mathcal{M} \rightarrow K\) as follows: \(\mathcal{A}_{m}\) samples a fresh random oracle \(H' \leftarrow \text{Func}(\mathcal{M} \times C \times H, K)\) and set

\[
H(\mu, c_0, c_1) = \begin{cases} 
H_{m}(\mu) & \text{if } c_0 = \text{Enc}(ek, \mu) \text{ and } c_1 = F(\mu) \\
H'(\mu, c_0, c_1) & \text{otherwise.}
\end{cases}
\]

This simulation is perfect and we conclude the proof. \(\Box\)

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

\[
H_{m}(\mu) = H(\mu, \text{Enc}(ek, \mu), F(\mu)).
\]

This simulation is perfect and we conclude the proof. \(\Box\)

We obtain the following theorems by combining the above theorem with Theorem F.1 and Theorem F.2:

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

**Theorem G.3** (Case of non-derandomized PKE). Suppose that a ciphertext space \(C\) of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then KEM = HU\(^\perp\)[PKE, H, F] is SPR-CCA-secure.

G.2 Sparseness

KEM = HU\(^\perp\)[PKE, H, F] is 1/|\(H|\)-sparse as HU\(_{m}^\perp\).

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

**Proof.** Let us consider \((c_0, c_1) \leftarrow S(1^{|x} \times U(H)). If c_0 is decrypted into \(\mu' \neq \bot\), then \(c_1 = F(\mu')\) with probability at most 1/|\(H|.|. Thus, KEM is 1/|\(H|\)-sparse. \(\Box\)
H Properties of $\text{HU}_m^\mathcal{L}$

Let us review $\text{HU}_m^\mathcal{L}$. Let PKE = (Gen, Enc, Dec) be a deterministic PKE scheme whose plaintext space is $\mathcal{M}$. Let $\mathcal{C}$ and $\mathcal{K}$ be a ciphertext and key space. Let $\mathcal{H}$ be a some finite space. Let $H: \mathcal{M} \rightarrow \mathcal{K}$, $\text{H}_{\text{prf}}: \{0,1\}^\ell \times \mathcal{C} \times \mathcal{H} \rightarrow \mathcal{K}$, and $F: \mathcal{M} \rightarrow \mathcal{H}$ be hash functions modeled by random oracles. KEM = (Gen, Enc, Dec) = $\text{HU}_m^\mathcal{L}$[PKE, H, F, $\text{H}_{\text{prf}}$] is defined as in Figure 15.

### H.1 SPR-CCA Security

Bindel et al. showed that if $\text{KEM}^\mathcal{L} = \text{U}_{\mathcal{M}}^\mathcal{L}$[PKE, H] is IND-CCA-secure then $\text{KEM}^\mathcal{L} = \text{U}_{\mathcal{M}}^\mathcal{L}$[PKE, H, $\text{H}_{\text{prf}}$] is also IND-CCA-secure [BHIT' 19; Theorem 3] by overwriting $\perp$ from the decapsulation query $c$ with the PRF value $\text{H}_{\text{prf}}(s, c)$. The same indifferentiable reduction can be applied to the SPR-CCA security of $\text{HU}_m^\mathcal{L}$ and $\text{HU}_m^\mathcal{L}$, and we obtain the following theorem.

**Theorem H.1** ($\text{HU}_m^\mathcal{L} \rightarrow \text{HU}_m^\mathcal{L}$). Let PKE be a deterministic PKE. Let $\text{KEM}^\mathcal{L} = \text{U}_{\mathcal{M}}^\mathcal{L}$[PKE, H, F] and $\text{KEM}^\mathcal{L} = \text{U}_{\mathcal{M}}^\mathcal{L}$[PKE, H, F, $\text{H}_{\text{prf}}$]. If $\text{KEM}^\mathcal{L}$ is SPR-CCA-secure, then $\text{KEM}^\mathcal{L}$ is also SPR-CCA-secure.

**Proof.** Suppose that we have an adversary $\mathcal{A}$ against the SPR-CCA security of $\text{KEM}^\mathcal{L}$. We construct an adversary $\mathcal{A}'$ against the SPR-CCA security of $\text{KEM}^\mathcal{L}$ as follows: Given an encapsulation key $ek$, a target ciphertext $(c_0, c_1)$, and a key $K'_m$, $\mathcal{A}'$ samples a fresh seed $s \leftarrow \mathcal{M}$. It runs $\mathcal{A}$ on input $ek$, $(c_0', c_1')$, and $K'_m$. If $\mathcal{A}$ queries a ciphertext $(c_0, c_1)$ to the decapsulation oracle, then $\mathcal{A}'$ queries the ciphertext $(c_0, c_1)$ and receives $K$. If $K \neq \perp$, then it returns $K$ to $\mathcal{A}$. Otherwise, it queries $(s, (c_0, c_1))$ to the random oracle $\text{H}_{\text{prf}}$ receives $K$, and returns $K$ to $\mathcal{A}$. If $\mathcal{A}$ outputs $b'$ and halts, then $\mathcal{A}'$ also outputs $b'$ and halts. This simulation is clearly perfect and the theorem follows. \hfill $\Box$

Apply the above indifferentiable reduction with **Theorem F.1** and **Theorem F.2**, we obtain the following theorems:

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

**Theorem H.3** (Case of non-derandomized PKE). Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If $\text{PKE}$ is strongly disjoint-simulatable, then KEM = $\text{HU}_m^\mathcal{L}$[PKE, H, F, $\text{H}_{\text{prf}}$] is SPR-CCA-secure, where we use the new simulator $\mathcal{S}' = \mathcal{S} \times \mathcal{U}(\mathcal{H})$.

### H.2 SSMT-CCA Security

**Theorem H.4**. Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If $\text{PKE}$ is strongly disjoint-simulatable, then KEM = $\text{HU}_m^\mathcal{L}$[PKE, H, F, $\text{H}_{\text{prf}}$] is SSMT-CCA-secure.

Formally speaking, for any $\mathcal{A}$, we have

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

The security proof is essentially same as that for SY (**Theorem 4.3**). Note that this security proof is irrelevant to PKE is deterministic PKE or one derandomized by $\text{T}$. 

Fig. 15. KEM = $\text{HU}_m^\mathcal{L}$[PKE, H, F]
Table 10. Summary of Games for the Proof of Theorem H.4: Enc’(ek, M) = \{(c_0, c_1) = (Enc(ek, m), F(\mu)) \mid m \in M\}.

\[ S(1^k) \times U(H) \setminus \text{Enc’(ek, M)} \] implies that the challenger generates \( c'_0 \leftarrow S(1^k), c'_1 \leftarrow H \) and returns \( \perp \) if \( (c'_0, c'_1) \in \text{Enc’(ek, M)}. \)

<table>
<thead>
<tr>
<th>Game</th>
<th>H F</th>
<th>( c'_0 )</th>
<th>( c'_1 )</th>
<th>( K^* )</th>
<th>Decryption</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game_0</td>
<td>H F</td>
<td>( S(1^k) )</td>
<td>U(H)</td>
<td>U(K)</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
<tr>
<td>Game_1</td>
<td>H F</td>
<td>( S(1^k) \setminus \text{Enc(ek, M)} )</td>
<td>U(H)</td>
<td>U(K)</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
<tr>
<td>Game_2</td>
<td>H F</td>
<td>( S(1^k) \setminus \text{Enc(ek, M)} )</td>
<td>U(H)</td>
<td>U(K)</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
<tr>
<td>Game_3</td>
<td>H F</td>
<td>( S(1^k) \setminus \text{Enc(ek, M)} )</td>
<td>U(H)</td>
<td>U(T)</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
<tr>
<td>Game_4</td>
<td>H F</td>
<td>( S(1^k) \setminus \text{Enc(ek, M)} )</td>
<td>U(H)</td>
<td>U(T)</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
<tr>
<td>Game_5</td>
<td>H F</td>
<td>( S(1^k) )</td>
<td>U(H)</td>
<td>Dec(\overline{\mu}, (c'_0, c'_1))</td>
<td>H(\mu)</td>
<td>H_{\text{prf}}(s, c_0, c_1)</td>
</tr>
</tbody>
</table>

Game_0: This game is the original game \( \text{Exp}^\text{smt-cca}_{\text{KEM},A}(\kappa) \) with \( b = 0 \). The challenge is generated as
\[
(c'_0, c'_1, K'_0) \leftarrow S(1^k) \times U(H) \times \mathcal{K}.
\]
We have
\[
\Pr[S_0] = 1 - \Pr[\text{Exp}^\text{smt-cca}_{\text{KEM},A}(\kappa) = 1 \mid b = 0].
\]

Game_1: In this game, the ciphertext is set as \( \perp \) if \( c'_0 \) is in \( \text{Enc(ek, M)} \).
The difference between two games Game_0 and Game_1 is bounded by statistical disjointness.

\[
[\Pr[S_0] - \Pr[S_1]] \leq \text{Disj}_{\text{PKE},S}(\kappa).
\]

Game_2: This game is the same as Game_1 except that \( H_{\text{prf}}(s, c, d) \) in the decapsulation oracle is replaced with \( H_q(c_0, c_1) \) where \( H_q : C \times \mathcal{K} \rightarrow \mathcal{K} \) is another random oracle.
As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound
\[
[\Pr[S_1] - \Pr[S_2]] \leq 2q_{H_{\text{prf}}} + q_{\text{Dec}} \cdot 2^{-\ell/2},
\]
where \( q_{H_{\text{prf}}} \) denote the number of queries to \( H_{\text{prf}} \) the adversary makes.

Game_3: This game is the same as Game_2 except that \( K^* := H_q(c'_0, c'_1) \) instead of chosen random. Since \( c'_0 \) is always outside of \( \text{Enc(ek, M)} \), \( \mathcal{A} \) cannot obtain any information about \( H_q(c'_0, c'_1) \) via the decapsulation oracle. Hence, the two games Game_2 and Games are equivalent and we have
\[
\Pr[S_2] = \Pr[S_3].
\]

Game_4: This game is the same as Game_3 except that \( H_q(\cdot, \cdot) \) is replaced by \( H_{\text{prf}}(s, \cdot, \cdot) \). As in [XY19, Lemmas 4.1], from Lemma 2.2 we have the bound
\[
[\Pr[S_3] - \Pr[S_4]] \leq 2q_{H_{\text{prf}}} + q_{\text{Dec}} \cdot 2^{-\ell/2}.
\]

Game_5: This game is the same as Game_4 except that \( K^* := \text{Dec}(\overline{\mu}, (c'_0, c'_1)) \) instead of \( H_{\text{prf}}(s, c'_0, c'_1) \). Recall that \( c'_0 \) is always in outside of \( \text{Enc(ek, M)} \). Thus, we always have \( \text{Dec}(c'_0) = \perp \) or \( \text{Enc(ek, Dec(c'_0))} \neq c'_0 \) and, thus, \( K^* = H_{\text{prf}}(s, c'_0, c'_1) \). Hence, the two games are equivalent. We have
\[
\Pr[S_4] = \Pr[S_5].
\]

Game_6: We finally how to compute \( (c'_0, c'_1) \). In this game, the ciphertext is chosen by \( S(1^k) \times U(H) \) as in Game_0.
The difference between two games Game_5 and Game_6 is bounded by statistical disjointness.
\[
[\Pr[S_5] - \Pr[S_6]] \leq \text{Disj}_{\text{PKE},S}(\kappa).
\]
Moreover, this game Game_6 is the original game \( \text{Exp}^\text{smt-cca}_{\text{KEM},A}(\kappa) \) with \( b = 1 \).
\[
\Pr[S_6] = \Pr[\text{Exp}^\text{smt-cca}_{\text{KEM},A}(\kappa) = 1 \mid b = 1].
\]

Summarizing the (in)equalities, we obtain
\[
\text{Adv}^\text{smt-cca}_{\text{KEM},A}(\kappa) = \Pr[S_6] - \Pr[S_5] \\
\leq 2\text{Disj}_{\text{PKE},S}(\kappa) + 4(q_{H_{\text{prf}}} + q_{\text{Dec}}) \cdot 2^{-\ell/2}.
\]
H.3 SCFR-CCA Security

**Theorem H.5.** If PKE is SCFR-CCA-secure (or XCFR-secure), then \( \text{KEM} = \text{HU}_m^H \) [PKE, H, F, H_{prf}] is SCFR-CCA-secure in the quantum random oracle model.

Note that this security proof is independent of that PKE is deterministic PKE or one derandomized by T.

**Proof.** Suppose that an adversary outputs a ciphertext \( c = (c_0, c_1) \) which is decapsulated into \( K \not\perp \) by \( dk_0 \) and \( dk_1 \), that is, \( \text{Dec}(dk_0, c) = \text{Dec}(dk_1, c) \). Let us define \( \mu_i = \text{Dec}(dk_i, c) \) for \( i \in \{0, 1\} \). We also define \( \mu_i = \mu_i' \) if \( c_0 = \text{Enc}(ek_i, \mu_i) \) and \( c_1 = F(\mu_i') \), and \( \perp \) otherwise.

We have five cases defined as follows:

1. Case 1 (\( \mu_0 = \mu_1 \not\perp \)): This violates the SCFR-CCA security (or the XCFR security) of the underlying PKE.
2. Case 2 (\( \perp \neq \mu_0 \neq \mu_1 \neq \perp \)): In this case, the decapsulation algorithm outputs \( K = H(\mu_0) = H(\mu_1) \) and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 3 (\( \mu_0 \neq \perp \) and \( \mu_1 \neq \perp \)): In this case, the decapsulation algorithms output \( K = H_{prf}(s_0, c_0, c_1) \) and \( H(\mu_1) \) and we find a claw \( ((s_0, c_0, c_1), \mu_1) \) of \( H_{prf} \) and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
4. Case 4 (\( \mu_0 \neq \perp \) and \( \mu_1 = \perp \)): In this case, the decapsulation algorithms output \( K = H(\mu_0) = H_{prf}(s_1, c_0, c_1) \) and we find a claw \( (\mu_0, (s_1, c_0, c_1)) \) of \( H_{prf} \) if \( s_0 \neq s_1 \). The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).
5. Case 5 (The other cases): In this case, the decapsulation algorithms output \( K = H_{prf}(s_0, c_0, c_1) = H_{prf}(s_1, c_0, c_1) \) and we find a collision \( ((s_0, c_0, c_1), (s_1, c_0, c_1)) \) of \( H_{prf} \) if \( s_0 \neq s_1 \). The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

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

If we add \( ek \) to F’s input, we can reduce the assumption on PKE.

**Theorem H.6.** Let \( \text{ColGen} \) be the event that when generating two keys \( (ek_i, dk_i) \leftarrow \text{Gen}(\star) \) for \( i \in \{0, 1\} \), they collide, that is, \( ek_0 = ek_1 \). If \( \text{Pr}[\text{ColGen}] \) is negligible, then \( \text{KEM} = \text{HU}_m^{\text{prf}} \) [PKE, H, F, H_{prf}] with \( c_1 = F(\mu, ek) \) is SCFR-CCA-secure in the quantum random oracle model.

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

**Proof.** Suppose that an adversary outputs a ciphertext \( c = (c_0, c_1) \) which is decapsulated into \( K \not\perp \) by \( dk_0 \) and \( dk_1 \), that is, \( \text{Dec}(dk_0, c) = \text{Dec}(dk_1, c) \). Let us define \( \mu_i = \text{Dec}(dk_i, c) \) for \( i \in \{0, 1\} \). We also define \( \mu_i = \mu_i' \) if \( c_0 = \text{Enc}(ek_i, \mu_i) \) and \( c_1 = F(\mu_i') \), and \( \perp \) otherwise.

We consider six cases defined as follows:

1. Case 1-1 (\( \mu_0 = \mu_1 \neq \perp \) and \( ek_0 = ek_1 \)): This case rarely occurs since \( \text{Pr}[\text{ColGen}] \) is negligible.
2. Case 1-2 (\( \mu_0 = \mu_1 \neq \perp \) and \( ek_0 \neq ek_1 \)): In this case, we have \( d = F(\mu_0, ek_0) = F(\mu_1, ek_1) \) with \( \mu_i \neq \mu_i' \) and \( ek_i \) and we succeed to find a collision for F, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 2 (\( \perp \neq \mu_0 \neq \mu_1 \neq \perp \)): In this case, the decapsulation algorithm outputs \( K = H(\mu_0) = H(\mu_1) \) and we succeed to find a collision for H and F, which is negligible for any QPT adversary (Lemma 2.3).
4. Case 3 (\( \mu_0 \neq \perp \) and \( \mu_1 \neq \perp \)): In this case, the decapsulation algorithms output \( K = H(\mu_0) = H_{prf}(s_1, c_0, c_1) \) and we find a claw \( ((s_0, c_0, c_1), (s_1, c_0, c_1)) \) of \( H_{prf} \) and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
5. Case 4 (\( \mu_0 \neq \perp \) and \( \mu_1 = \perp \)): In this case, the decapsulation algorithms output \( K = H(\mu_0) = H_{prf}(s_1, c_0, c_1) \) and we find a claw \( (\mu_0, (s_1, c_0, c_1)) \) of H and \( H_{prf} \) and H. The probability that we find such claw is negligible for any QPT adversary (Lemma 2.4).
6. Case 5 (The other cases): In this case, the decapsulation algorithms output \( K = H_{prf}(s_0, c_0, c_1) = H_{prf}(s_1, c_0, c_1) \) and we find a collision \( ((s_0, c_0, c_1), (s_1, c_0, c_1)) \) of \( H_{prf} \) if \( s_0 \neq s_1 \), which occurs with probability at least \( 1 - 1/2^l \). The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

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

I Properties of \( \text{HU}^{\text{prf}}_m \)

Next, we consider a variant of HU with implicit rejection, \( \text{HU}^{\text{prf}}_m \), which is used in Classic McEliece. Let PKE = (Gen, Enc, Dec) be a deterministic PKE scheme whose plaintext space is \( M \). Let C and \( \mathcal{K} \) be a ciphertext and key space. Let \( \mathcal{H} \) be a finite space. Let \( H, H_{prf} : M \times C \times \mathcal{H} \rightarrow \mathcal{K} \) and F: \( M \rightarrow \mathcal{H} \) be hash functions modeled by random oracles. \( \text{KEM} = (\text{Gen}, \text{Enc}, \text{Dec}) = \text{HU}^{\text{prf}}_m \) [PKE, H, F, H_{prf}] is defined as in Figure 16.
KEM\(=\)\(M\) deterministic PKE scheme whose plaintext space is \(\mathcal{M}\). Let \(C\) be a ciphertext space. Let \(\mathcal{K}\) be a key space. Let \(\mathcal{H}\) be a hash function.

Finally, we consider another variant of HU with implicit rejection, \(HU\). Properties of \(HU\).

I.1 SPR-CCA Security

In order to show the SPR-CCA security of \(HU^{L-prf}\), we first show the following theorem for indifferentiable reductions, which is obtained by mimicking that for \(U^{L-prf}\) in [BHH+19, Theorem 5].

**Theorem I.1 (\(HU_{in}^{L-prf}\)).** Let \(PKE\) be a deterministic \(PKE\). Let \(KEM_{in} = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) and \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\).\n
1. If \(KEM_{in}\) is SPR-CCA-secure, then \(KEM\) is SPR-CCA-secure also.
2. If \(KEM\) is SPR-CCA-secure, then \(KEM_{in}\) is SPR-CCA-secure also.

Since the proof is the same as that of Theorem G.1, we omit it.

We then apply the above theorem to Theorem H.2 and Theorem H.3 and obtain the following theorems:

**Theorem I.2 (Case of derandomized \(PKE\)).** Let \(PKE = \{\text{PKE}_0, \mathcal{G}\}\). Suppose that a ciphertext space \(C\) of \(PKE\) depends on the public parameter only. If \(PKE\) is strongly disjoint-simulatable with simulator \(S\), then \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) is SPR-CCA-secure, where we use the new simulator \(S' = S \times U(\mathcal{H})\).\n
**Theorem I.3 (Case of non-derandomized \(PKE\)).** Suppose that a ciphertext space \(C\) of \(PKE\) depends on the public parameter only. If \(PKE\) is strongly disjoint-simulatable, then \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) is SPR-CCA-secure, where we use the new simulator \(S' = S \times U(\mathcal{H})\).

I.2 SSMT-CCA Security

**Theorem I.4.** Suppose that a ciphertext space \(C\) of \(PKE\) depends on the public parameter only. If \(PKE\) is strongly disjoint-simulatable, then \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) is SSMT-CCA-secure.

Formally speaking, for any \(\mathcal{A}\), we have:

\[
\text{Adv}^{\text{ssmt-cca}}_{\text{KEM}, \mathcal{A}}(\kappa) \leq 2 \text{Dist}_{\text{PKE}, \mathcal{S}}(\kappa) + 4(q_{\text{prf}} + q_{\text{Dec}}) \cdot 2^{-\kappa/2}.
\]

Since the security proof is the same as that for \(HU_{in}^{L}\) (Theorem H.4), we omit it.

I.3 SCFR-CCA Security

**Theorem I.5.** If \(PKE\) is SCFR-CCA-secure (or XCFR-secure), then \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) is SCFR-CCA-secure in the quantum random oracle model.

**Theorem I.6.** Let \(\text{Col}_{\text{Gen}}\) be the event that when generating two keys \((e_k, d_k) \leftarrow \text{Gen}(\kappa)\) for \(i \in \{0, 1\}\), they collide, that is, \(e_0 = e_1\). If \(\text{Pr}[\text{Col}_{\text{Gen}}]\) is negligible, then \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\) with \(c_1 = F(\mu, e_k)\) is SCFR-CCA-secure in the quantum random oracle model.

The security proofs are the same as those for \(HU_{in}^{L}\) (Theorem H.5 and Theorem H.6) and we omit them.

J Properties of \(HU^{L}\)

Finally, we consider another variant of HU with implicit rejection, \(HU\). Let \(PKE = (\text{Gen}, \text{Enc}, \text{Dec})\) be a deterministic PKE scheme whose plaintext space is \(\mathcal{M}\). Let \(C\) and \(\mathcal{K}\) be a ciphertext and key space. Let \(\mathcal{H}\) be a hash function.

**Fig. 16.** \(KEM = HU^{L-prf}[\text{PKE}, \mathcal{H}, \mathcal{F}, \text{prf}]\)
Table 11. Summary of Games for the Proof of Theorem J.4: 'S(1^k) \setminus Enc(ek, M)' implies that the challenger generates $c_0^* \leftarrow S(1^k)$, $c_1^* \leftarrow \mathcal{H}$ and returns $\bot$ if $c_0^* \in Enc(ek, M)$.

<table>
<thead>
<tr>
<th>Game</th>
<th>$H$</th>
<th>$F$</th>
<th>$c_0^*$</th>
<th>$c_1^*$</th>
<th>$K^*$</th>
<th>Decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Game 1</td>
<td>$H$</td>
<td>$F$</td>
<td>$S(1^k)$</td>
<td>$U(H)$</td>
<td>$U(K)$</td>
<td>$H(\mu, c_0, c_1)$ \quad $H(s, c_0, c_1)$</td>
</tr>
<tr>
<td>Game 2</td>
<td>$H$</td>
<td>$F$</td>
<td>$S(1^k)$</td>
<td>$U(H)$</td>
<td>$U(K)$</td>
<td>$H(\mu, c_0, c_1)$ \quad $H(q, c_0, c_1)$ \quad statistical disjointness</td>
</tr>
<tr>
<td>Game 3</td>
<td>$H$</td>
<td>$F$</td>
<td>$S(1^k)$</td>
<td>$Enc(ek, M)$</td>
<td>$U(H)$</td>
<td>$H_s(c_0, c_1)$ \quad $H(q, c_0, c_1)$ \quad $H_g(c_0^* c_1^*)$ is hidden</td>
</tr>
<tr>
<td>Game 4</td>
<td>$H$</td>
<td>$F$</td>
<td>$S(1^k)$</td>
<td>$Enc(ek, M)$</td>
<td>$U(H)$</td>
<td>$Dec(\overline{\mathcal{H}}, (c_0^* c_1^*))$ \quad $H(s, c_0, c_1)$ \quad re-encryption check</td>
</tr>
<tr>
<td>Game 5</td>
<td>$H$</td>
<td>$F$</td>
<td>$S(1^k)$</td>
<td>$Enc(ek, M)$</td>
<td>$U(H)$</td>
<td>$Dec(\overline{\mathcal{H}}, (c_0^* c_1^*))$ \quad $H(s, c_0, c_1)$ \quad statistical disjointness</td>
</tr>
</tbody>
</table>

J.1 SPR-CCA security

In order to show the SPR-CCA security of $HU^\perp$, we use the following theorem, an adapted version of [BHH*19, Theorem 3].

Theorem J.1 ($HU^\perp \rightarrow HU^\perp$). Let PKE be a deterministic PKE. Let $KEM^\perp = HU^\perp[PKE, H, F]$ and $KEM^\perp = HU^\perp[PKE, H, F]$. If $KEM^\perp$ is SPR-CCA-secure, then $KEM^\perp$ is also SPR-CCA-secure.

Proof. Suppose that we have an adversary $\mathcal{A}$ against the SPR-CCA security of $KEM^\perp$. We construct an adversary $\mathcal{A}'$ against the SPR-CCA security of $KEM^\perp$ as follows: Given an encapsulation key $ek$, a target ciphertext $(c_0^*, c_1^*)$, and a key $K^\star$, $\mathcal{A}'$ samples a fresh seed $s$ and runs $\mathcal{A}$ on input $ek, (c_0^*, c_1^*)$, and $K^\star$. If $\mathcal{A}$ queries a ciphertext $(c_0, c_1)$ to the decapsulation oracle, then $\mathcal{A}'$ queries the ciphertext $(c_0, c_1)$ and receives $K$. If $c_0^* \neq \bot$, then it returns $K$ to $\mathcal{A}$; Otherwise, it queries $(s, c_0, c_1)$ to the random oracle $H$, receives $K$, and returns $K$ to $\mathcal{A}$. Then, $\mathcal{A}'$ outputs $b^*$ and halts, then $\mathcal{A}'$ also outputs $b^*$ and halts.

This simulation is clearly perfect and the theorem follows. $\square$

Applying the above theorem to Theorem G.2 and Theorem G.3, we obtain the following theorems:

Theorem J.2 (Case of derandomized PKE). Let PKE be $T[PKE_0, G]$. Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable with simulator $S$, then $KEM = HU^\perp[PKE, H, F]$ is SPR-CCA-secure, where we use the new simulator $S' = S \times U(H)$.

Theorem J.3 (Case of non-derandomized PKE). Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then $KEM = HU^\perp[PKE, H, F]$ is SPR-CCA-secure.

J.2 SSMT-CCA Security

Theorem J.4. Suppose that a ciphertext space $C$ of PKE depends on the public parameter only. If PKE is strongly disjoint-simulatable, then $KEM = HU^\perp[PKE, H, F]$ is SSMT-CCA-secure.

Formally speaking, for any $\mathcal{A}$, we have

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

The security proof is essentially same as that for SYX (Theorem 4.3).
Game$_0$: This game is the original game $\text{Exp}_{\text{KEM}, \mathcal{A}}^{\text{ssmt-cca}}(\kappa)$ with $b = 0$. The challenge is generated as

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

We have

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

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

$$\left|\Pr[S_1] - \Pr[S_2]\right| \leq 2(q_H + q_{\text{Dec}})\sqrt{|\mathcal{M}|}.$$ 

where $q_H$ denote the number of queries to $H$ the adversary makes.

Game$_2$: In this game, the ciphertext is set as $\perp$ if $c_0^*$ is in $\text{Enc}(ek, M)$.

The difference between two games Game$_1$ and Game$_2$ is bounded by statistical disjointness.

$$\Pr[S_1] - \Pr[S_2] \leq \text{Disj}_{\text{PKE, } S}(\kappa).$$

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

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

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

$$\left|\Pr[S_3] - \Pr[S_4]\right| \leq 2(q_H + q_{\text{Dec}})\sqrt{|\mathcal{M}|}.$$ 

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

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

Game$_6$: We finally replace how to compute $(c_0^*, c_1^*)$. In this game, the ciphertext is chosen by $S(1^*) \times U(\mathcal{H})$ as in Game$_0$.

The difference between two games Game$_5$ and Game$_6$ is bounded by statistical disjointness.

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

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

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

Summing the (in)equalities, we obtain Theorem J.4:

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

J.3 SCFR-CCA Security

Theorem J.5. If $\text{PKE}$ is SCFR-CCA-secure (or XCFR-secure) then $\text{KEM} = \text{HU}_{\text{m}}^{\text{KEM}} [\text{PKE}, H, F]$ is SCFR-CCA-secure in the quantum random oracle model.

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

We have five cases defined as follows:

1. Case 1 ($\mu_0 = \mu_1 \neq \perp$): This violates the SCFR-CCA security (or the XCFR security) of the underlying PKE.
2. Case 2 ($\bot \neq \mu_0 \neq \mu_1 \neq \bot$): In this case, the decapsulation algorithm outputs $K = H(\mu_0, c_0, c_1) = H(\mu_1, c_0, c_1)$ and we succeed to find a collision for $H$ and $F$, which is negligible for any QPT adversary (Lemma 2.3).

3. Case 3 ($\mu_0 = \bot$ and $\mu_1 \neq \bot$): In this case, the decapsulation algorithms output $K = H(s_0, c_0, c_1)$ and $H(\mu_1, c_0, c_1)$. As in the proof of Theorem E.3, we can replace $H(s_0, \cdot)$ with $H(c_0, \cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((c_0, c_1), (\mu_1, c_0, c_1))$ between $H_0$ and $H$. The probability that we find such a claw is negligible for any QPT adversary (Lemma 2.4).

4. Case 4 ($\mu_0 \neq \bot$ and $\mu_1 = \bot$): In this case, the decapsulation algorithms output $K = H(\mu_0, c_0, c_1) = H(s_1, c_0, c_1)$. This follows as Case 3.

5. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = H(s_0, c_0, c_1) = H_{\text{prf}}(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of $H$ if $s_0 \neq s_1$, which occurs with overwhelming probability $1 - 1/|M|$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. \hfill \qed

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

Theorem 6. Let ColGen be the event that when generating two keys $(ek_i, dk_i) \leftarrow \text{Gen}(1^k)$ for $i \in \{0, 1\}$, they collide, that is, $ek_0 = ek_1$. If $\text{Pr}[\text{ColGen}]$ is negligible, then $\text{KEM} = \text{HU}^F[\text{PKE}, H, F, H_{\text{prf}}]$ with $c_1 = F(\mu, ek)$ is $\text{SCFR-CCA}$-secure in the quantum random oracle model.

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

Proof. Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \bot$ by $dk_0$ and $dk_1$, that is, $\text{Dec}(dk_0, c) = \text{Dec}(dk_1, c)$. We define $\mu'_i = \text{Dec}(dk_i, c_0)$ for $i \in \{0, 1\}$. We also define $\mu_i = \mu'_i$ if $c_0 = \text{Enc}(ek_i, \mu'_i)$ and $c_1 = F(\mu'_i, ek_i)$, and $\bot$ otherwise.

We consider six cases defined as follows:

1. Case 1-1 ($\mu_0 = \mu_1 \neq \bot$ and $ek_0 = ek_1$): This case rarely occurs since $\text{Pr}[\text{ColGen}]$ is negligible.

2. Case 1-2 ($\mu_0 = \mu_1 \neq \bot$ and $ek_0 \neq ek_1$): In this case, we have $d = F(\mu_0, c_0) = F(\mu_1, c_1)$ with $(\mu'_0, ek_0) \neq (\mu'_1, ek_1)$ and we succeed to find a collision for $F$, which is negligible for any QPT adversary (Lemma 2.3).

3. Case 2 (\bot \neq \mu_0 \neq \mu_1 \neq \bot): In this case, the decapsulation algorithm outputs $K = H(\mu_0, c_0, c_1) = H(\mu_1, c_0, c_1)$ and we succeed to find a collision for $H$ and $F$, which is negligible for any QPT adversary (Lemma 2.3).

4. Case 3 ($\mu_0 = \bot$ and $\mu_1 \neq \bot$): In this case, the decapsulation algorithms output $K = H(s_0, c_0, c_1)$ and $H(\mu_1, c_0, c_1)$. As in the proof of Theorem E.3, we can replace $H(s_0, \cdot)$ with $H(c_0, \cdot)$ by introducing negligible error (Lemma 2.2). After that, we find a claw $((c_0, c_1), (\mu_1, c_0, c_1))$ between $H_0$ and $H$. The probability that we find such a claw is negligible for any QPT adversary (Lemma 2.4).

5. Case 4 ($\mu_0 \neq \bot$ and $\mu_1 = \bot$): In this case, the decapsulation algorithms output $K = H(\mu_0, c_0, c_1) = H(s_1, c_0, c_1)$. This follows as Case 3.

6. Case 5 (The other cases): In this case, the decapsulation algorithms output $K = H(s_0, c_0, c_1) = H(s_1, c_0, c_1)$ and we find a collision $((s_0, c_0, c_1), (s_1, c_0, c_1))$ of $H$ if $s_0 \neq s_1$, which occurs with overwhelming probability $1 - 1/|M|$. The probability that we find such collision is negligible for any QPT adversary (Lemma 2.3).

We conclude that the advantage of the adversary is negligible in any cases. \hfill \qed

K Classic McEliece

We briefly review Classic McEliece [ABC*20] in subsection K.1, discuss the security properties of the underlying DPKE, CM-DPKE, in subsection K.2, and discuss the security properties of Classic McEliece in subsection K.3. We want to show that, under appropriate assumptions, Classic McEliece is ANON-CCA-secure in the QROM, and Classic McEliece leads to ANON-CCA-secure hybrid PKE in the QROM. (Unfortunately, Classic McEliece is not collision-free [GMP21a].) In order to do so, we show that the underlying CM-DPKE of Classic McEliece is strongly disjoint-simulatable under appropriate assumptions in subsection K.2. Since Classic McEliece is obtained by applying HU^F-\text{prf} to CM-DPKE, this strong disjoint-simulatability implies that Classic McEliece is SPR-CCA-secure and SMT-CCA-secure in the QROM under those assumptions. Those three properties lead to the anonymity of Classic McEliece and hybrid PKE in the QROM as we wanted. We also discuss a modification of Classic McEliece in order to salvage collision-freeness.
Table 12. Parameter sets of Classic McEliece in Round 3. Note that \( q = 2^m \) and \( k = n - mt \). (We omit the semi-systematic forms.)

<table>
<thead>
<tr>
<th>Parameter sets</th>
<th>( m )</th>
<th>( n )</th>
<th>( t )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>kem/mceliece348864</td>
<td>12</td>
<td>3488</td>
<td>64</td>
<td>2720</td>
</tr>
<tr>
<td>kem/mceliece468896</td>
<td>13</td>
<td>4688</td>
<td>96</td>
<td>3360</td>
</tr>
<tr>
<td>kem/mceliece6688128</td>
<td>13</td>
<td>6688</td>
<td>128</td>
<td>5024</td>
</tr>
<tr>
<td>kem/mceliece6960119</td>
<td>13</td>
<td>6960</td>
<td>119</td>
<td>5413</td>
</tr>
<tr>
<td>kem/mceliece8192128</td>
<td>13</td>
<td>8192</td>
<td>128</td>
<td>6528</td>
</tr>
</tbody>
</table>

K.1 Review of Classic McEliece

Classic McEliece [ABC+20] is a KEM scheme based on the Niederreiter PKE, in which a public key is a scrambled parity-check matrix, a plaintext is an error vector, and a ciphertext is a syndrome. See Table 12 for concrete parameter values (we omit semi-systematic ones).

Let \( m, n, t, k, q \) be positive integers with \( q = 2^m \) and \( k = n - mt \). Define \( S = \{ e \in \mathbb{F}_q^m : \text{HW}(e) = t \} \), which is a plaintext space. Let \( I_{n-k} \) be the identity matrix of dimension \( n - k \). The underlying deterministic PKE of Classic McEliece, which we call CM-DPKE, is summarized as follows, where we only consider the systematic form and omit the details for the semi-systematic form:

- **Gen(1\(^t\))**: Choose a monic irreducible polynomial \( g \in \mathbb{F}_q[x] \) of degree \( t \) and distinct \( \alpha_1, \ldots, \alpha_n \leftarrow \mathbb{F}_q \).

- Compute a parity-check matrix \( H \in \mathbb{F}_2^{n \times k} \) of the Goppa code generated by \( g \) and \( \alpha_1, \ldots, \alpha_n \). Reduce \( H \) to systematic form \([I_{n-k} | T]\). (If this fails, return \( \perp \).)

- Output \( e_k := T \in \mathbb{F}_2^{(n-k) \times k} \) and \( dk := (T, I) \), where \( I := (g, \alpha_1, \ldots, \alpha_n) \). We note that, using \( I \), one can correct an error up to \( t \), because the minimum distance of the Goppa code is at least \( 2t + 1 \) by design.

- **Enc(ek)**: Define \( H_e := [I_{n-k} | T] \in \mathbb{F}_2^{(n-k) \times n} \). Compute \( c := H_e \cdot e \in \mathbb{F}_2^m \). Output \( c \).

- **Dec(dk, c)**: Extend \( c \) to \( v := (c, 0, \ldots, 0) \in \mathbb{F}_2^n \). Find the unique codeword \( \hat{c} \) in the Goppa code defined by \( I \) that satisfies \( \text{HW}(\hat{c} - v) \leq t \). Set \( e := v + \hat{c} \). If \( \text{HW}(e) = t \) and \( c = H_e \), then return \( e \). Otherwise, return \( \perp \).

Classic McEliece applies \( HU^{\perp-\text{prf}} \) to CM-DPKE, where \( H(\mu, c_0, c_1) = \text{SHAKE256}_256(0x01, \mu || c_0 || c_1) \), \( H_{\text{prf}}(s, c_0, c_1) = \text{SHAKE256}_256(0x00, s || c_0 || c_1) \), and \( F(e) = \text{SHAKE256}_256(0x02, e) \), and is defined in Figure 18.

![Table 12. Parameter sets of Classic McEliece](image)

**Fig. 18.** Classic McEliece

K.2 Properties of CM-DPKE

It is known that the Niederreiter PKE is pseudorandom under appropriate assumptions. In order to adapt the argument, we use the following assumptions:

**Definition K.1.** Fix the parameter set. We define a random key-generation algorithm \( \text{RandGen}(pp) \) as follows:

Choose \( \hat{H} \leftarrow U(\mathbb{F}_2^{n \times k}) \), reduce \( \hat{H} \) to systematic form \([I_{n-k} | \hat{T}]\) (if this fails, resample), and output \( \hat{T} \in \mathbb{F}_2^{(n-k) \times k} \).

- **The modified PR-Key assumption**: It is computationally hard to distinguish \( T \) and \( \hat{T} \), where \( (T, sk) \leftarrow \text{Gen}(1^t) \) and \( \hat{T} \leftarrow \text{RandGen}(pp) \).

- **The modified Decisional Syndrome Decoding assumption**: It is computationally hard to distinguish \([\hat{T}, I_{n-k} | \hat{T}] \cdot e\) from \((\hat{T}, u)\) with \( \hat{T} \leftarrow \text{RandGen}(pp) \), \( e \leftarrow \text{FixedWeight()} \), and \( u \leftarrow U(\mathbb{F}_2^{n-k}) \).

---

50
Security: Assuming the modified PR-key assumption and the modified Decisional Syndrome Decoding assumption, it is easy to show that CM-DPKE is ciphertext-indistinguishable in the sense of disjoint simulatability as the case of NTRU-DPKE. Grubbs et al. observed that for any public key \( T \), we have that, under those assumptions, Classic McEliece is SPR-CCA-secure in the QROM.

**Theorem K.3.** Properties of Classic McEliece

Combining CM-DPKE’s strong disjoint-simulatability with previous theorems on HU, we obtain the following theorems.

**Theorem K.1.** Suppose that the modified PR-key assumption and the modified Decisional Syndrome Decoding assumption hold. Then, Classic McEliece is SP-DPKE-secure in the QROM.

**Proof.** Suppose that an adversary outputs a ciphertext \( \epsilon \) for all parameter sets of Classic McEliece. Grubbs et al. observed that for any public key \( T \), \( \epsilon \) is a valid ciphertext of plaintext \( \epsilon \) since \( H : \epsilon = I_{n-k} T \). Hence, CM-DPKE and Classic McEliece is not collision free.

**Theorem K.2.** Suppose that the modified PR-key assumption and the modified Decisional Syndrome Decoding assumption hold. NTRU-DPKE is strongly disjoint-simulatable (Lemma K.1). In addition, CM-DPKE is perfectly correct. Applying Theorem K.3 and Theorem K.4, we obtain the theorem.

**Theorem K.3.** Let \( \text{ColGen} \) be the event that when generating two keys \( (e_k, d_k) \leftarrow \text{Gen}(1^n) \) for \( i \in \{0, 1\} \), they collide, that is, \( e_k = e_k \). If \( \Pr[\text{ColGen}] \) is negligible, then the modified Classic McEliece is SPR-CCA-secure in the QROM.

**Proof.** Suppose that an adversary outputs a ciphertext \( c = (c_0, c_1) \) which is decapsulated into \( K \neq \perp \) by \( d_k \) and \( d_k \), that is, \( \text{Dec}(d_k, c) = \text{Dec}(d_k, c) \). Let us define \( e'_i = \text{Dec}(d_k, c) \) for \( i \in \{0, 1\} \). We also define \( e_i = e'_i \) if \( e_0 = \text{Enc}(e_k, \phi) \) and \( e_1 = F(e'_i, \text{Hash}(e_k)) \). We consider seven cases defined as follows:

1. Case 1-1: \( e_i = e_i \neq \perp \) and \( e_0 = e_k \): This case rarely occurs since \( \Pr[\text{ColGen}] \) is negligible.
2. Case 1-2: \( e_i = e_i \neq \perp \), \( e_0 \neq e_k \), and \( \text{Hash}(e_0) = \text{Hash}(e_k) \): In this case, we have \( \text{Hash}(e_0) = \text{Hash}(e_k) \) with \( e_0 \neq e_k \) and we succeed to find a collision for Hash, which is negligible for any QPT adversary (Lemma 2.3).
3. Case 1-3: \( e_i = e_i \neq \perp \), \( e_0 \neq e_k \), and \( \text{Hash}(e_0) \neq \text{Hash}(e_k) \): In this case, we have \( d = F(e_0, \text{Hash}(e_0)) = F(e_1, \text{Hash}(e_k)) \) with \( e_0 \neq e_k \neq e_k \) and we succeed to find a collision for \( F \), which is negligible for any QPT adversary (Lemma 2.3).
4. Case 2: \( \perp \neq e_0 \neq e_1 \): In this case, the decapsulation algorithm outputs \( K = H(e_0) = H(e_1) \) and we succeed to find a collision for \( H \), which is negligible for any QPT adversary (Lemma 2.3).
We next consider an intermediate PKE scheme. FO
Kyber applies a variant of the FO transform with implicit rejection, denoted by
\[ H(\ell, c) \]
where
\[ H \]

Review of Kyber in Round 3: Kyber [SAB+20] is a KEM scheme based on the Module LWE problem. We briefly review Kyber.

The underlying PKE scheme of Kyber, which we call Kyber-PKE, is summarized as follows:

- **Gen**: The key generation algorithm outputs \( ek \) and \( dk \).
- **Enc**: The encryption algorithm is probabilistic. Taking \( \mu \in \{0, 1\}^{256} \), it outputs \( c \).
- **Dec**: The decryption algorithm is deterministic and outputs \( \mu' \in \{0, 1\}^{256} \).

We next consider an intermediate PKE scheme \( \text{PKE}_0 = (\text{Gen}_0, \text{Enc}_0, \text{Dec}_0) \) where the encryption algorithm uses pseudorandomness, which we call Kyber-PKE-PRG:

- **Gen**: \( \text{Gen}(\text{pp}) = \text{Gen}(\text{pp}) \):
- **Enc**: \( \text{Enc}_0(ek, \mu; r) \): It uses \( \rho_i = \text{SHAKE256}_X(r, i) \) for \( i = 0, 1, \ldots \) to sample randomness \( \rho \) of \( \text{Enc}(ek, \mu) \). It then outputs \( c = \text{Enc}(ek, \mu; r) \).
- **Dec**: \( \text{Dec}_0(dk, c) = \text{Dec}(dk, c) \):

Kyber applies a variant of the FO transform with implicit rejection, denoted by \( FO^\ell \), to Kyber-PKE-PRG, where \( H' = \text{SHA3-256} \), \( G(\mu, h) = \text{SHA3-512} \), and \( H = \text{SHAKE256}_X \) with unspecified output bits \( X \), and is defined as in Figure 19.

<table>
<thead>
<tr>
<th>\text{Gen}(1^k)</th>
<th>\text{Enc}(ek)</th>
<th>\text{Dec}(dk, c), where ( \overline{dk} = (dk, ek, h, s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (ek, dk) \leftarrow \text{Gen}_0(1^k) )</td>
<td>( \mu \leftarrow {0, 1}^{256} )</td>
<td>( \mu' \leftarrow \text{Dec}_0(dk, c) )</td>
</tr>
<tr>
<td>( h \leftarrow H'(ek) )</td>
<td>( \mu := H'(') )</td>
<td>( (K', r') := G(\mu', h) )</td>
</tr>
<tr>
<td>( s \leftarrow {0, 1}^{256} )</td>
<td>( (K, r) := G(\mu, H'(ek)) )</td>
<td>( c' := \text{Enc}_0(ek, \mu', r') )</td>
</tr>
<tr>
<td>( \overline{dk} := (dk, ek, h, s) )</td>
<td>( c := \text{Enc}_0(ek, \mu; r) )</td>
<td>if ( c \neq c' ), then return ( K := H(s, H'(c)) )</td>
</tr>
<tr>
<td>return ( (ek, \overline{dk}) )</td>
<td>( K := H(K', H'(c)) )</td>
<td>else return ( K := H(k', H'(c)) )</td>
</tr>
</tbody>
</table>

Fig. 19. Kyber
Security: Grubbs et al. [GMP21a] pointed out there are technical barriers. At first, a pre-key $\bar{K}$ and a randomness $r$ is generated by $G(\mu, H'(ek))$. We can treat is as $\bar{K} = G\bar{0}(\mu, H'(ek))$ and $r = G1(\mu, H'(ek))$, where $G0(x)$ and $G1(x)$ are defined as the first and last 256-bits of $G = \text{SHA3-512}$. Using this notion, we compute $K = H(G\bar{0}(\mu, H'(ek)), H'(c))$. Grubbs et al. solved the problem on nested random oracles on $\mu$ by letting $G_\tau(\mu) := G\bar{0}(\mu, H'(ek)) : \{0,1\}^{256} \rightarrow \{0,1\}^{256}$ and simulating $G_\tau$ by a random polynomial over $\text{GF}(2^{512})$ of degree $2q_G + 1$ as in [TU16, HIK17]. Grubbs et al. succeeded to show its IND-CCA-security if $K$ was computed as $H(G\bar{0}(\mu), c)$ as in FO$^L$'s IND-CCA-security as open problem. We also left it here.

M Saber

Review of Saber: Saber [DKR20] is a KEM scheme based on the Module LWR problem. We briefly review Saber.

The underlying PKE scheme of Saber, which we call Saber-PKE, is summarized as follows:
- $\text{Gen}(pp)$: The key-generation algorithm outputs $ek$ and $dk$.
- $\text{Enc}(ek, \mu; r)$: The encryption algorithm is probabilistic. Taking $\mu \in \{0,1\}^{256}$, it outputs $c$.
- $\text{Dec}(dk, c)$: The decryption algorithm is deterministic and outputs $\mu' \in \{0,1\}^{256}$.

We next consider an intermediate PKE scheme $\text{Saber}_0 = (\text{Gen}_0, \text{Enc}_0, \text{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call Saber-PKE-PRG:
- $\text{Gen}_0(pp) = \text{Gen}(pp)$;
- $\text{Enc}_0(ek, \mu; r)$: It uses $\text{SHAKE128}_\chi(r)$ to sample randomness $\rho$ of $\text{Enc}(ek, \mu)$. It then outputs $c \coloneqq \text{Enc}(ek, \mu; \rho)$.
- $\text{Dec}_0(dk, c) = \text{Dec}(dk, c)$:

Saber applies the same variant of the FO transform with implicit rejection as Kyber to Saber-PKE-PRG, where $H' = \text{SHA3-256}, G(\mu, h) = \text{SHA3-512}$, and $H = \text{SHA3-256}$, and is defined as in Figure 20.

<table>
<thead>
<tr>
<th>$\text{Gen}(1^s)$</th>
<th>$\text{Enc}(ek)$</th>
<th>$\text{Dec}(dk, c)$, where $\overline{dk} = (dk, ek, h, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(ek, dk) \leftarrow \text{Gen}_0(1^s)$</td>
<td>$\mu \leftarrow {0,1}^{256}$</td>
<td>$\mu' \coloneqq \text{Dec}_0(dk, c)$</td>
</tr>
<tr>
<td>$h \leftarrow H'(ek)$</td>
<td>$\mu \coloneqq H'(\mu)$</td>
<td>$(K, r') \coloneqq G(\mu', h)$</td>
</tr>
<tr>
<td>$s \leftarrow {0,1}^{256}$</td>
<td>$(\bar{K}, r) \coloneqq G(\mu, H'(ek))$</td>
<td>$c' \coloneqq \text{Enc}_0(ek, \mu'; r')$</td>
</tr>
<tr>
<td>$\overline{dk} \coloneq (dk, ek, h, s)$</td>
<td>$c \coloneqq \text{Enc}_0(ek, \mu; r)$</td>
<td>if $c \neq c'$, then return $K \coloneq H(s, H'(c))$</td>
</tr>
<tr>
<td>return $(ek, \overline{dk})$</td>
<td>$K \coloneq H(\bar{K}, H'(c))$</td>
<td>else return $K \coloneq H(\bar{K}', H'(c'))$</td>
</tr>
</tbody>
</table>

Fig. 20. Saber

Security: Grubbs et al. [GMP21a] wrote Saber uses FO$^L$ as defined in [DKR20, Section 2.5]. However, the specification uses FO$^U$ [DKR20, Section 8.5]. Thus, Saber lacks the IND-CCA-security proof in the QROM as Kyber. We also left proving the IND-CCA security of Saber in the QROM as an open problem. It might be interesting to study anonymity and robustness in the ROM.

N BIKE

We briefly review BIKE [ABB20] in subsection N.1, discuss the security properties of the underlying PKE, BIKE-PKE, and its derandomized version, BIKE-DFKE, in subsection N.2, and discuss the security properties of BIKE in subsection N.3. We want to show that, under appropriate assumptions, BIKE is ANON-CCA-secure in the QROM, and BIKE leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM.

In order to do so, we show that the underlying BIKE-DFKE of BIKE is strongly disjoint-simulatable under appropriate assumptions and XCFR-secure in subsection N.2. BIKE is obtained by applying U$^L$ to BIKE-DFKE, and the former implies that BIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM under those assumptions and the latter implies that BIKE is SCFR-CCA-secure in the QROM. Those three properties lead to the anonymity of BIKE and hybrid PKE in the QROM as we wanted.
N.1 Review of BIKE

BIKE in round 3 [ABB⁺20] is a KEM scheme based on QC-MDPC [MTSB13], which is a variant of the McEliece PKE upon a code with quasi-cyclic (QC) moderate density parity-check (MDPC) matrix. BIKE can be considered as the Niederreiter PKE scheme upon a code with the QC-MDPC matrix. Let \( \mathcal{R} := \mathbb{F}_{2}[x]/(x^r - 1) \). Let \( \mathcal{H}_w := \{(h_0, h_1) \in \mathcal{R}^2 : \text{HW}(h_0) = \text{HW}(e_1) = w/2 \} \). Let \( \mathcal{E}_r := \{(e_0, e_1) \in \mathcal{R}^2 : \text{HW}(e_0, e_1) = r \} \). For concrete values of \( r, w, \) and \( t \), see Table 13.

<table>
<thead>
<tr>
<th>Parameter sets</th>
<th>( r )</th>
<th>( w )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIKE-1</td>
<td>12,323</td>
<td>142,134</td>
<td></td>
</tr>
<tr>
<td>BIKE-2</td>
<td>24,659</td>
<td>206,199</td>
<td></td>
</tr>
<tr>
<td>BIKE-3</td>
<td>40,973</td>
<td>274,264</td>
<td></td>
</tr>
</tbody>
</table>

The underlying CPA-secure PKE scheme of BIKE, which we call BIKE-PKE, is summarized as follows:

- \( \text{Gen}_{pp} : dk \leftarrow \mathcal{H}_w \). Output \( ek = h := h_1 \cdot h_0^{-1} \in \mathcal{R} \) and \( dk \).
- \( \text{Enc}_\mathcal{E}_r(e_k, \mu) \in \{0, 1\}^{256} ; r \): Sample \((e_0, e_1) \leftarrow \mathcal{E}_r \) by using the randomness \( r \). Compute \( u := e_0 + e_1 h \in \mathcal{R} \) and \( v := \mu \oplus L(e_0, e_1) \) and output \( c := (u, v) \).
- \( \text{Dec}_{\mathcal{H}_w}(dk, (u, v)) \): Compute \((e_0, e_1) \leftarrow \text{decode}(u h_0, (h_0, h_1)) \), where decode is a decoder of the QC-MDPC code with parity-check matrix generated by \( h_0 \) and \( h_1 \). Output \( \mu' := v \oplus L(e_0, e_1) \), where \( L = \text{SHA-384} \).

Notice that \( uh_0 = e_0 h_0 + e_1 h_1 \), which is the syndrome of \((e_0, e_1)\) with the parity-check matrix generated by \( h_0 \) and \( h_1 \).

BIKE applies a variant of the FO transform with implicit rejection, \( \text{FO}^\perp := \mathcal{U}^\perp \circ T \), to BIKE-PKE PKE, where \( G = \text{SHAKE256} \) and \( H = \text{SHA-384} \), and is defined as in Figure 21.

Recall that \( \text{FO}^\perp \) is \( \mathcal{U}^\perp \circ T \). In what follows, we first study BIKE-PKE’s properties and then study BIKE-DPKE’s properties, where BIKE-DPKE is obtained by derandomizing BIKE-PKE with transform \( T \).

N.2 Properties of BIKE-PKE and BIKE-DPKE

Although we can invoke theorems on \( \text{FO}^\perp \) by Grubbs et al. [GMP21a] to show BIKE’s anonymity and collision-freeness, we can show BIKE’s anonymity through another pass.

Assumptions: For \( b \in \{0, 1\} \), define the finite set \( \mathcal{F}_b := \{h \in \mathcal{R} : \text{HW}(h) \equiv b \ (\text{mod} \ 2) \} \). That is, a set of all binary vectors of length \( r \) and parity \( b \). We suppose that \( w \) is even and \( w/2 \) is odd, which hold for all parameter sets of BIKE.

Definition N.1 (The 2-Decisional Quasi-Cyclic Code-Finding (2-DQCCF) assumption [ABB⁺20]). For any (QIPPT adversary, it is hard to distinguish the following two distributions:

- \( h := h_1 \cdot h_0^{-1} \), where \((h_0, h_1) \leftarrow \mathcal{H}_w \).
- \( h \leftarrow \mathcal{F}_1 \).

 Definition N.2 (The 2-Computational Quasi-Cyclic Syndrome Decoding (2-CQCSD) assumption \([ABB^*20]\)).
 For any \((Q)PPT\) adversary, given \( (h, u : = h_1 + e_0) \), where \( h \leftarrow \mathcal{F}_1 \) and \( (e_0, e_1) \leftarrow \mathcal{E}_i \), it is hard to find \((e'_0, e'_1) \in \mathcal{E}_i \) with \( u = h'_1 + e'_0 \).

 Definition N.3 (The 2-Decisional Quasi-Cyclic Syndrome Decoding (2-DQCSD) assumption \([ABB^*20]\)).
 For any \((Q)PPT\) adversary, it is hard to distinguish the following two distributions:
- \( \langle h, u : = h'_1 + e'_0 \rangle \), where \( h \leftarrow \mathcal{F}_1 \) and \( (e_0, e_1) \leftarrow \mathcal{E}_i \),
- \( \langle h, u \rangle \), where \( h \leftarrow \mathcal{F}_1 \) and \( u \leftarrow \mathcal{T}_i \mod 2 \).

 BIKE-Simple: Before showing the security, we consider the following deterministic PKE scheme, which we call BIKE-Simple:
- \( \text{Gen}(pp) : dk \leftarrow \mathcal{H}_w \). Output \( ek = h \leftarrow h_1 \cdot h_0^{-1} \in \mathcal{R} \) and \( dk \).
- \( \text{Enc}(ek, (e_0, e_1)) \leftarrow \mathcal{E}_i \). Compute \( u \leftarrow e_0 + e_1h \in \mathcal{R} \) and output \( u \).
- \( \text{Dec}(dk, u) \): Output \( (e_0, e_1) \leftarrow \text{decode}(u, (h_0, h_1)) \).

 The proposers showed that this scheme is OW-CPA-secure using appropriate assumptions as follows:

 Lemma N.1 \([ABB^*20, \text{Theorem 1}]\). If the 2-DQCCF and 2-CQCSD assumptions hold, then BIKE-Simple is OW-CPA-secure.

 Remark N.1. It is easy to show BIKE-Simple’s disjoint simulatability: Let \( \mathcal{F}_i \) be a ciphertext space. We define the simulator as sampling \( u \leftarrow U(\mathcal{F}_i) \). Statistical disjointness follows from the fact that \( |\mathcal{F}_i| = 2^t/2 \gg \mathcal{L} = |\mathcal{E}_i| \geq |\text{Enc}(ek, \mathcal{E}_i)| \). We can show ciphertext indistinguishability by using the 2-DQCCF and 2-DQCSD assumptions as we showed ciphertext indistinguishability of NTRU-DPKE and CM-DPKE.

 Remark N.2. Applying SXY and assuming \( \delta \) is negligible, we can obtain a tightly CCA-secure KEM scheme with shorter ciphertext, which leads to anonymous, robust hybrid PKE.

 Security of BIKE-PKE: We next show that BIKE-PKE is ciphertext-indistinguishable in the QROM.

 Lemma N.2. Suppose that the 2-DQCCF and 2-DQCSD assumptions hold. Then BIKE-PKE is ciphertext-indistinguishable in the QROM with a simulator that outputs \( u \leftarrow \mathcal{T}_i \mod 2 \) and \( v \leftarrow \mathbb{F}_2^{256} \).

 Proof (Proof Sketch). We consider four games \( \text{Game}_i \) for \( i = 0, 1, \ldots, 4 \) defined as follows:
- \( \text{Game}_0 \): In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( (h_0, h_1) \leftarrow \mathcal{H}_w \) and \( h \leftarrow h_1 \cdot h_0^{-1} \).
  - Encryption: \( \mu \leftarrow \mathbb{F}_2^{256} \), \( (e_0, e_1) \leftarrow \mathcal{E}_i \). Compute \( u \leftarrow e_0 + he_1 \) and \( v := \mu \oplus \mathcal{L}(e_0, e_1) \); return \( c = (u, v) \).
- \( \text{Game}_1 \): In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( (h_0, h_1) \leftarrow \mathcal{H}_w \) and \( h \leftarrow h_1 \cdot h_0^{-1} \).
  - Encryption: \( (e_0, e_1) \leftarrow \mathcal{E}_i \). Compute \( u \leftarrow e_0 + he_1 \); return \( c = (u, v) \).
- \( \text{Game}_2 \): In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( h \leftarrow \mathcal{T}_i \).
  - Encryption: \( (e_0, e_1) \leftarrow \mathcal{E}_i \). Compute \( u \leftarrow e_0 + he_1 \); return \( c = (u, v) \).
- \( \text{Game}_3 \): In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( h \leftarrow \mathcal{T}_i \).
  - Encryption: \( u \leftarrow \mathcal{T}_i \mod 2 \); return \( c = (u, v) \).
- \( \text{Game}_4 \): In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( (h_0, h_1) \leftarrow \mathcal{H}_w \) and \( h \leftarrow h_1 \cdot h_0^{-1} \).
  - Encryption: \( u \leftarrow \mathcal{T}_i \mod 2 \); return \( c = (u, v) \).

\( \text{Game}_0 \) and \( \text{Game}_1 \) are equivalent, since \( \mu \) in \( \text{Game}_0 \) and \( v \) in \( \text{Game}_1 \) is chosen uniformly at random. \( \text{Game}_1 \) and \( \text{Game}_2 \) are computationally indistinguishable under the 2-DQCCF assumption. \( \text{Game}_2 \) and \( \text{Game}_3 \) are computationally indistinguishable under the 2-DQCSD assumption. \( \text{Game}_3 \) and \( \text{Game}_4 \) are computationally indistinguishable under the 2-DQCCF assumption. Summing up those (in)equalities, we obtain the lemma.

\( \square \)

We next consider BIKE-PKE is IND-CPA-secure in the QROM. The proposers showed the security in the ROM as follows:

 Lemma N.3 \([ABB^*20, \text{Theorem 2}]\). If the 2-DQCCF and 2-DQCSD assumptions hold, then BIKE-PKE is IND-CPA-secure in the QROM.

 Unfortunately, applying their idea directly to the QROM setting, the security proof becomes loose since it will involve the OZK lemma \( \text{Corollary A.1} \). We here show the IND-CPA security of BIKE-PKE in the QROM tightly using the idea of \([SXY18]\).
Lemma N.4. Assume that the 2-DQCCF and 2-DQCSD assumptions hold and BIKE-PKE is $\delta$-correct with negligible $\delta$. Then, BIKE-PKE is IND-CPA-secure (and OW-CPA-secure) in the QROM.

Proof (Proof Sketch). We consider Game$_{b,i}$ for $b \in \{0, 1\}$ and $i = 0, \ldots, 4$ defined as follows:

- Game$_{0,b}$: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h \leftarrow h_1 \cdot h_0^{-1}$.
  - Encryption given $\mu_0$ and $\mu_1$: $(e_0, e_1) \leftarrow \mathcal{E}_i$; compute $u \leftarrow e_0 + he_1$, $k \leftarrow L(e_0, e_1)$, and $v \leftarrow \mu_b \oplus k$; return $c = (u, v)$.

- Game$_{1,b}$: In this game, we use another random oracle $L_q : \mathcal{R} \rightarrow \{0, 1\}^{256}$ and define $L(e_0, e_1) = L_q(he_0 + e_1)$.
  - An encryption key and a target ciphertext is computed as follows:
    - Key generation: $(h_0, h_1) \leftarrow \mathcal{H}_w$ and $h \leftarrow h_1 \cdot h_0^{-1}$.
    - Encryption given $\mu_0$ and $\mu_1$: $(e_0, e_1) \leftarrow \mathcal{E}_i$; compute $u \leftarrow e_0 + he_1$, $k \leftarrow L_q(u)$, and $v \leftarrow \mu_b \oplus k$; return $c = (u, v)$.

- Game$_{2,b}$: In this game, we use random $h$. An encryption key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_i$.
  - Encryption given $\mu_0$ and $\mu_1$: $(e_0, e_1) \leftarrow \mathcal{E}_i$; compute $u \leftarrow e_0 + he_1$, $k \leftarrow L_q(u)$, and $v \leftarrow \mu_b \oplus k$; return $c = (u, v)$.

- Game$_{3,b}$: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_i$.
  - Encryption given $\mu_0$ and $\mu_1$: $u \leftarrow \mathcal{F}_i \mod 2^i$; compute $k \leftarrow L_q(u)$, and $v \leftarrow \mu_b \oplus k$; return $c = (u, v)$.

- Game$_{4,b}$: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: $h \leftarrow \mathcal{F}_i$.
  - Encryption given $\mu_0$ and $\mu_1$: $u \leftarrow \mathcal{F}_i \mod 2^i$; $k \leftarrow \{0, 1\}^{256}$; compute $v \leftarrow \mu_b \oplus k$; return $c = (u, v)$.

Game$_{0,b}$ and Game$_{1,b}$ are equivalent if the mapping $(e_0, e_1) \mapsto he_0 + e_1$ is injective, which is satisfied if a key pair is accurate. Game$_{1,b}$ and Game$_{2,b}$ are computationally indistinguishable under the 2-DQCCF assumption. Game$_{2,b}$ and Game$_{3,b}$ are computationally indistinguishable under the 2-DQCSD assumption. Game$_{3,b}$ and Game$_{4,b}$ are equivalent if $u$ is outside of the image of the mapping $(e_0, e_1) \mapsto e_0 + e_1 h$, which occurs with overwhelming probability. Game$_{4,0}$ and Game$_{4,1}$ are equivalent since $k$ is uniformly at random. Summing up those (in)equalities, we obtain the lemma.

Remark N.3. We can replace the term $\delta$ with the probability that the mapping $(e_0, e_1) \mapsto e_0 + e_1 h$ is injective for random $h \leftarrow \mathcal{F}_i$.

Security of BIKE-DPKE: We then consider BIKE-DPKE obtained by applying $T$ to BIKE-PKE.

Lemma N.5. Assume that the 2-DQCCF and 2-DQCSD assumptions hold. Then, BIKE-DPKE is strongly disjoint-simulatable.

Proof. Statistical disjointness follows from the fact that $|S(1^t)| \approx 2^t / 2 \cdot 2^{256}$ and $|\text{Enc}(ek, M)| \leq 2^{256}$. We can show ciphertext indistinguishability by invoking Theorem D.1 since BIKE-PKE is ciphertext-indistinguishable (Lemma N.2) and oneway (Lemma N.4).

We next consider BIKE-DPKE’s XCFR-security:

Lemma N.6. Let $\epsilon_0$ be a probability that $h_0 - h_1 \notin \mathbb{R}^*$ holds for two randomly generated keys $h_0$ and $h_1$. Let $\epsilon_0$ be a probability that an efficient adversary finds $\mu$ such that $e_1 = 0$ where $(e_0, e_1) \leftarrow \mathcal{E}_i(G(\mu))$. Suppose that $\epsilon_\mu$ and $\epsilon_0$ is negligible. Then, BIKE-DPKE is XCFR-secure.

Proof (Proof sketch). Let us consider $e_k = h_1$ and $dk_i = (h_0, h_1)$ for $i \in \{0, 1\}$. If the adversary outputs $c = (u, v)$, it should be decrypted into $\mu$ by using $dk_0$ and $dk_1$, respectively. Let $(e_0, e_1) \leftarrow \mathcal{E}_i(G(\mu))$. We have $u = e_0 + e_1 h_0 = e_0 + e_1 h_1$ in the re-encryption check. This implies $(h_0 - h_1) \cdot e_1 = 0 \in \mathcal{R}$. If $e_1 \neq 0$ and $h_0 - h_1 \in \mathbb{R}^*$, then this leads a contradiction. Thus, the lemma holds.

N.3 Properties of BIKE

Combining BIKE-DPKE’s strong disjoint-simulatability and XCFR security with previous theorems on $U^L$, we obtain the following theorems.

Theorem N.1. Suppose that the 2-DQCCF and 2-DQCSD assumptions hold and BIKE-DPKE is $\delta$-correct with negligible $\delta$. Then, BIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM.

Proof. Under the 2-DQCCF and 2-DQCSD assumptions, BIKE-DPKE is strongly disjoint-simulatable (Lemma N.5). Applying Theorem E.2 and Theorem E.3, we obtain the theorem.
Theorem N.2. Let $\epsilon_n$ be a probability that $h_0 - h_1 \not\in R^*$ holds for two randomly generated keys $h_0$ and $h_1$. Let $\epsilon_0$ be a probability that an efficient adversary finds $\mu$ such that $\epsilon_1 = 0$ where $(e_0, e_1) := E_t(G(\mu))$. Suppose that and $\epsilon := \epsilon_n + \epsilon_0$ is negligible. Then, BIKE is SCFR-CCA-secure in the QROM.

Proof. Under the hypothesis, BIKE-DPKE is XCFR-secure (Lemma N.6). Applying Theorem E.4, we have that BIKE is SCFR-CCA-secure in the QROM. \qed

Theorem N.3. Suppose that the 2-DQCCF and 2-DQCSD assumptions hold and BIKE-DPKE is $\delta$-correct with negligible $\delta$. Then, BIKE is ANON-CCA-secure in the QROM.

Proof. Due to Theorem N.1, under the hypothesis, BIKE is SPR-CCA-secure in the QROM. Thus, applying Theorem 2.5, we have that, under those assumptions, BIKE is ANON-CCA-secure in the QROM. \qed

Theorem N.4. Let $\epsilon_n$ be a probability that $h_0 - h_1 \not\in R^*$ holds for two randomly generated keys $h_0$ and $h_1$. Let $\epsilon_0$ be a probability that an efficient adversary finds $\mu$ such that $\epsilon_1 = 0$ where $(e_0, e_1) := E_t(G(\mu))$. Suppose that and $\epsilon := \epsilon_n + \epsilon_0$ is negligible. Suppose that the 2-DQCCF and 2-DQCSD assumptions hold and BIKE-DPKE is $\delta$-correct with negligible $\delta$. Then, BIKE leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM, combined with SPR-orCCA-secure and FROB-secure DEM.

Proof. Due to Theorem N.1, under the 2-DQCCF and 2-DQCSD assumptions and the assumption on the correctness, BIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM. Thus, combining BIKE with SPR-orCCA-secure DEM, we obtain a SPR-CCA-secure hybrid PKE in the QROM (Theorem 3.2). Moreover, BIKE is SCFR-CCA-secure in the QROM (Theorem N.2) under the hypothesis on $\epsilon$. Thus, if DEM is FROB-secure, then the hybrid PKE is SROB-CCA-secure (Theorem 2.2). \qed

O FrodoKEM

Review of FrodoKEM: FrodoKEM [NAB+20] is an LWE-based KEM scheme in the alternates candidates. The underlying PKE scheme of FrodoKEM, which we call FrodoKEM-PKE, is summarized as follows:
- $Gen(\mathbb{P})$: The key-generation algorithm outputs $ek$ and $dk$.
- $Enc(ek, \mu; r)$: The encryption algorithm is probabilistic. Taking $\mu \in \{0, 1\}^k$, it outputs $c$.
- $Dec(dk, c)$: The decryption algorithm is deterministic and outputs $\mu' \in \{0, 1\}^k$.

We next consider an intermediate PKE scheme $PKE_0 = (Gen_0, Enc_0, Dec_0)$ where the encryption algorithm uses pseudorandomness, which we call FrodoKEM-PKE-PRG:
- $Gen_0(\mathbb{P}) = Gen(\mathbb{P})$.
- $Enc_0(ek, \mu; r)$: It uses $\rho = \text{SHAKE}_128_{\mathbb{R}}(8\times96\lceil r \rceil)$ to sample randomness $\rho$. It then outputs $c := Enc(ek, \mu; \rho)$.
- $Dec_0(dk, c) = Dec(dk, c)$.

FrodoKEM applies a variant of the FO transform with implicit rejection to FrodoKEM-PKE-PRG, where $H'$, $G$, and $H$ are SHAKE 128 or SHAKE256, and is defined as in Figure 22: We can treat them as different random oracles because their input length differ.

<table>
<thead>
<tr>
<th>$Gen(\mathbb{P})$</th>
<th>$Enc(ek)$</th>
<th>$Dec(dk, c)$, where $\overline{dk} = (dk, ek, h, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(ek, dk) \leftarrow Gen(\mathbb{P})$</td>
<td>$\mu \leftarrow {0, 1}^k$</td>
<td>$\mu' := Dec_0(dk, c)$</td>
</tr>
<tr>
<td>$h \leftarrow H'(ek)$</td>
<td>$(\bar{K}, r) := G(\mu, H'(ek))$</td>
<td>$(\bar{K}', r') := G(\mu', h)$</td>
</tr>
<tr>
<td>$s \leftarrow {0, 1}^k$</td>
<td>$c := Enc_0(ek, \mu; r)$</td>
<td>$c' := Enc_0(ek, \mu'; r')$</td>
</tr>
<tr>
<td>$\overline{dk} := (dk, ek, h, s)$</td>
<td>$K := H(\bar{K}, c)$</td>
<td>if $c \neq c'$, then return $K := H(s, c)$</td>
</tr>
<tr>
<td>return $(ek, \overline{dk})$</td>
<td>return $(c, K)$</td>
<td>else return $K := H(\bar{K}', c)$</td>
</tr>
</tbody>
</table>

Fig. 22. FrodoKEM

Security: Grubbs et al. [GMP21a] fortunately show the security of the variant of the FO transform. Thus, we can apply their result to FrodoKEM.
P HQC

We briefly review HQC [AAB⁺20] in subsection P.1, discuss the security properties of the underlying PKE, HQC-PKE, and its derandomized version, HQC-DPKE, in subsection P.2, and discuss the security properties of HQC in subsection P.3. We want to show that, under appropriate assumptions, HQC is ANON-CCA-secure in the QROM, and HQC leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM. In order to do so, we show that the underlying HQC-DPKE of HQC-128/196 is strongly disjoint-simulatable under appropriate assumptions in subsection P.2. Unfortunately, we find that HQC-256 is not anonymous. HQC is obtained by applying HU to HQC-DPKE, and the strong disjoint simulatability implies that HQC-128/196 is SPR-CCA-secure and SMT-CCA-secure in the QROM under those assumptions. We directly prove that HQC is SROB-CCA-secure in the QROM under an appropriate assumption. Those three properties lead to the anonymity and robustness of HQC-128/196 and hybrid PKE in the QROM as we wanted.

P.1 Review of HQC

HQC [AAB⁺20] is another code-based KEM scheme in the alternate candidates.

Let $\mathcal{R} := \mathbb{F}_2[x]/(x^r - 1)$. Let $C$ be a decodable $[n_1, n_2, k]$ code generated by $G \in \mathbb{F}_2^{k \times n_1 n_2}$, where $n_1 n_2 \leq r$.

Let decode be a decoder algorithm which corrects an error up to $\delta$. Let $S_w := \{ x \in \mathcal{R} | \text{HW}(x) = w \}$. For a polynomial $A = \sum_i a_i x^i \in \mathcal{R}$, we define $\text{trunc}(A, l) = (a_0, \ldots, a_{l-1}) \in \mathbb{F}_2^l$. For concrete values, see Table 14.

Table 14. Parameter sets of HQC in Round 3.

<table>
<thead>
<tr>
<th>parameter sets</th>
<th>$r$</th>
<th>$n_1$</th>
<th>$k_1$</th>
<th>$n_2$</th>
<th>$k_2$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$w$</th>
<th>$w_e$</th>
<th>$w_r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>hqc-128</td>
<td>17</td>
<td>669</td>
<td>46</td>
<td>16</td>
<td>31</td>
<td>384</td>
<td>8</td>
<td>192</td>
<td>66</td>
<td>75</td>
</tr>
<tr>
<td>hqc-192</td>
<td>35</td>
<td>851</td>
<td>56</td>
<td>24</td>
<td>32</td>
<td>640</td>
<td>8</td>
<td>320</td>
<td>100</td>
<td>114</td>
</tr>
<tr>
<td>hqc-256</td>
<td>57</td>
<td>637</td>
<td>90</td>
<td>32</td>
<td>59</td>
<td>640</td>
<td>8</td>
<td>320</td>
<td>131</td>
<td>149</td>
</tr>
</tbody>
</table>

The underlying PKE scheme of HQC, which we call HQC-PKE, is summarized as follows:

- $\text{Gen(pp)}$: $h_0 \leftarrow \mathcal{R}, (x, y) \leftarrow S_w^d$. Compute $h_1 := x + h_0 y$. Output $dk := (x, y)$ and $ek := (h_0, h_1)$.
- $\text{Enc(ek, } \mu \in \mathbb{F}_2^k; (e, f, t) \in S_w x S_w \times S_w)$: Output:
  $$c = (a, v) := (h_0 t + f, \text{trunc}(h_1 t + e, n_1 n_2) \oplus \mu G) \in \mathcal{R} \times \mathbb{F}_2^{n_1 n_2}.$$  
- $\text{Dec}(dk, (a, v))$: Compute $a := v \oplus \text{trunc}(u_x, n_1 n_2) \in \mathbb{F}_2^{n_1 n_2}$ and output $\text{decode}(a)$.

We next consider an intermediate PKE scheme $\text{PKE}_0 = (\text{Gen}_0, \text{Enc}_0, \text{Dec}_0)$ where the encryption algorithm uses pseudorandomness, which we call HQC-PKE-PRG:

- $\text{Gen}_0(pp) = \text{Gen(pp)}$
- $\text{Enc}_0(ek, \mu; r)$: Use $\rho = \text{SHAKE256}(r, 0x00)$ to sample $(e, f, t) \in S_w \times S_w \times S_w$. Output $(a, v) := \text{Enc}(ek, \mu; (e, f, t))$
- $\text{Dec}_0(\mu, (a, v)) = \text{Dec}(dk, (a, v))$

HQC applies a variant of the FO transform with explicit rejection $\text{HFO}^1 = \text{HU} \circ T$ to HQC-PKE-PRG $\text{PKE}_0$, where $G(\mu) = \text{SHAKE256512}(\mu, 0x03), F(\mu) = \text{SHAKE256512}(\mu, 0x04)$, and $H(\mu, (c_0, c_1)) = \text{SHAKE256512}(\mu, 0x05)$. We can treat them as different random oracles because their input length differ.

Recall that $\text{HFO}^1$ is $\text{HU} \circ T$. In what follows, we first study HQC-PKE’s and HQC-PKE-PRG’s properties and then study HQC-DPKE’s properties, where HQC-DPKE is obtained by derandomizing HQC-PKE-PRG with transform $T$.

P.2 Properties of HQC-PKE

Grubbs et al. [GMP21a] showed properties of a variant of $\text{HFO}^1$, in which $c_1 = f(\mu, c_0)$ instead of $c_1 = f(\mu)$. We here show HQC’s anonymity directly by using properties of $\text{HFO}^1 = \text{HU} \circ T$.
For $b \in \{0, 1\}$, define the finite set $\mathcal{T}_b := \{ h \in \mathcal{R} : h(1) \equiv b \pmod{2}\}$, that is, a set of all binary vectors of length $r$ and parity $b$. Similarly, for $b, b_0, b_1 \in \{0, 1\}$, we define the sets

\[
\mathcal{T}_{b_0, b_1}^{2,3} := \{ H = [1, h] \in \mathcal{R}^2 : h \in \mathcal{T}_b, h_0 \in \mathcal{T}_{b_0} \wedge h_1 \in \mathcal{T}_{b_1} \}.
\]

**Definition P.1** (The 2-Decisional Quasi-Cyclic Syndrome Decoding (2-DQCSD) assumption [AAB\textsuperscript{20}]). Fix $b \in \{0, 1\}$, $w$, and $b' = (1 + b)w \mod 2$. For any (Q)PPT adversary, it is hard to distinguish the following two distributions:

- $(H, H \cdot (x, y))$, where $H \leftarrow \mathcal{T}_b^{1,2}$ and $(x_1, x_2) \leftarrow \mathcal{S}_w^2$.
- $(H, z)$, where $H \leftarrow \mathcal{T}_b^{1,2}$ and $y \leftarrow \mathcal{T}_b'$.  

**Definition P.2** (The 3-Decisional Quasi-Cyclic Syndrome Decoding (3-DQCSD) assumption [AAB\textsuperscript{20}]). Fix $b_0, b_1 \in \{0, 1\}$, and $w$. Let $b'_0 := (1 + b_0)w \mod 2$ and $b'_1 := (1 + b_1)w \mod 2$. For any (Q)PPT adversary, it is hard to distinguish the following two distributions:

- $(H, H \cdot (x_0, x_1, x_2))$, where $H \leftarrow \mathcal{T}_{b_0, b_1}^{2,3}$ and $(x_0, x_1, x_2) \leftarrow \mathcal{S}_w^3$.
- $(H, (z_0, z_1))$, where $H \leftarrow \mathcal{T}_{b_0, b_1}^{2,3}$, $z_0 \leftarrow \mathcal{T}_{b'_0}$, and $z_1 \leftarrow \mathcal{T}_{b'_1}$.

For collision-freeness, we define the following new assumption:

**Definition P.3** (The 3-Computational Quasi-Cyclic Codeword Finding (3-CQCCF) assumption). For any (Q)PPT adversary, given $(1, h, h')$ where $h, h' \in \mathcal{R}$, it is hard to find a non-zero codeword $(f, t, t')$ whose Hamming weight is at most $4w_r$.

**Security of HQC-PKE**: Using those assumptions, the proposers showed the IND-CPA security of HQC-PKE:

**Lemma P.1** ([AAB\textsuperscript{20}, Theorem 5.1], adapted). Assume that the 2-DQCSD and 3-DQCSD assumptions hold. Then, HQC-PKE is IND-CPA-secure (and OW-CPA-secure).

By mimicking their proof, we can show that it is ciphertext-indistinguishable as follows:

**Lemma P.2.** Assume that the 2-DQCSD and 3-DQCSD assumptions hold. Then, HQC-PKE is ciphertext-indistinguishable with a simulator that outputs $u \leftarrow \mathcal{T}_{b_0}$ and $v \leftarrow \mathcal{P}^{r_{c_1}}$, where $b_0 := (1 + h_0(1))w_r \mod 2$.

**Proof (Proof Sketch).** In what follows, we define the parity of $h_1$ as $b := (1 + h_0(1))w \mod 2$, the parity of $u$ as $b := (1 + h_0(1))w_r \mod 2$, and the parity of $v$ as $b := w_r \mod 2$. We consider games Game\textsubscript{0} for $i = 0, \ldots, 4$ defined as follows:

- **Game\textsubscript{0}**: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}, x, y \leftarrow \mathcal{S}_{w_r}$, and $h_1 := x \cdot y$.
  - Encryption: $\mu \leftarrow \mathcal{P}^k$, $e \leftarrow \mathcal{S}_{w_r}$, and compute $u := h_0t + f$ and $v := \text{trunc}(h_1t + e, n_1n_2) \in \mu G$.

- **Game\textsubscript{1}**: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: $h_0 \leftarrow \mathcal{R}, h_1_t \leftarrow \mathcal{T}_b$.
  - Encryption: $\mu \leftarrow \mathcal{P}^k$, $e \leftarrow \mathcal{S}_{w_r}$, and compute $u := h_0t + f$ and $v := \text{trunc}(h_1^tt + e, n_1n_2) \in \mu G$.
- Game2: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( h_0 \rightarrow R, h_1^{*} \rightarrow F_B \).
  - Encryption: \( \mu \rightarrow \mathbb{Z}_2^k, e \rightarrow F_{w_e}, t, f \rightarrow F_{w_t}, \) and compute \( u := h_0 t + f \) and \( v := \text{trunc}(h_1^{*} t + e, n, n_2) \oplus G \).

- Game3: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( h_0 \rightarrow R, h_1^{*} \rightarrow F_2.B \).
  - Encryption: \( u \rightarrow F_B, \) and \( v \rightarrow \mathbb{Z}_2^{n_2} \).

- Game4: In this game, an encryption key and a target ciphertext is computed as follows:
  - Key generation: \( h_0 \rightarrow R, x, y \rightarrow S_w, \) and \( h_1 := x + h_0 y. \)
  - Encryption: \( u \rightarrow F_B, \) and \( v \rightarrow \mathbb{Z}_2^{n_2} \).

Game2 and Game3 are computationally indistinguishable under the 2-DQCSD assumption. Game4 and Game2 are computationally indistinguishable under the 3-DQCSD assumption. Game2 and Game3 are statistically indistinguishable, because \( \text{trunc} \) truncates \( r - n_1 n_2 \) bits of \( y := h_1^{*} t + e \) in Game2 and thus, \( \text{trunc}(\tilde{y}, n_1 n_2) \)'s distribution is statistically close to the uniform distribution over \( \mathbb{Z}_2^{n_2} \). Game3 and Game4 are computationally indistinguishable under the 2-DQCSD assumption. Summing up those (in)equalities, we obtain the lemma.

\[ \square \]

We notice that HQC-128/196 are strongly pseudorandom, while HQC-256 is not strong:

**Corollary P.1.** HQC-128 and HQC-196 are strongly ciphertext-indistinguishable, while HQC-256 is not.

**Proof.** Let us compute the parity of \( h_1, b := (1 + h_0(1)) w \mod 2, \) and the parity of \( u, b_0 := (1 + h_0(1)) w_r \mod 2. \) According to Table 14, we obtain that the parity \( b \) of \( h_1 \) is 0, 0, 1–\( h_0 \) and the parity \( b_0 \) of \( u = 1, 0, h_0(1) \), for HQC-128/192/256, respectively. Thus, the simulator for HQC-128/192 can ignore the encryption key \( (h_0, h_1) \) and we can say that HQC-128/196 are strongly ciphertext-indistinguishable. However, the simulator for HQC-256 depends on \( h_0(1) \) and HQC-256 is not strongly ciphertext-indistinguishable. Indeed, the parity of \( u \) leaks the information of \( h_0 \) of the encryption key for HQC-256.

\[ \square \]

**Security of HQC-PKE-PRG:** We next consider HQC-PKE-PRG, whose encryption algorithm uses a PRG SHAKE256(·, 0x02) instead of true randomness. The IND-CPA security and ciphertext indistinguishability of HQC-PKE-PRG follows from PRG’s quantum security tightly.

**Lemma P.3.** Assume that the 2-DQCSD and 3-DQCSD assumptions hold and SHAKE256(·, 0x02) is quantumly-secure PRG. Then, HQC-PKE-PRG is ciphertext-indistinguishable and IND-CPA-secure (and OW-CPA-secure). In addition, HQC-PKE-PRG for HQC-128/196 is strongly ciphertext-indistinguishable.

**Security of HQC-DPKE:** We then consider HQC-DPKE obtained by derandomizing HQC-PKE-PRG by \( T \).

**Lemma P.4.** Assume that the 2-DQCSD and 3-DQCSD assumptions hold and SHAKE256(·, 0x02) is quantumly-secure PRG. Then, HQC-DPKE is disjoint-simulatable. Especially, HQC-DPKE for HQC-128/196 is strongly disjoint-simulatable.

**Proof.** Statistical disjointness follows from the fact that \(|S(1^*)| \approx 2^k / 2 \cdot 2^{n_1 n_2} \) and \(|\text{Enc}'(ek, M)| \leq 2^k \). We can show ciphertext indistinguishability by invoking Theorem D.1 since HQC-PKE-PRG is ciphertext indistinguishable and OW-CPA-secure (Lemma P.3). Strong disjoint simulatability for HQC-128/196 follows from Lemma P.3.

\[ \square \]

**P.3 Properties of HQC**

Combining HQC-DPKE’s strong disjoint-simulatability with previous theorems on HU², we obtain the following theorems.

**Theorem P.1.** Assume that the 2-DQCSD and 3-DQCSD assumptions hold and SHAKE256(·, 0x02) is quantumly-secure PRG. Then, HQC-128/106 is SPR-CCA-secure in the QROM. It is also \( 1/2^{512} \)-sparse in the QROM.

**Proof.** Under the 2-DQCSD and 3-DQCSD assumptions and quantum security of SHAKE256(·, 0x02), HQC-DPKE for HQC-128/196 is strongly disjoint-simulatable (Lemma P.4). Applying Theorem G.2, we obtain the SPR-CCA security in the QROM. In addition, using the fact that \( F(\cdot) = \text{SHAKE256}_{512}(\cdot, 0x04) \)'s range is \( \{0, 1\}^{512} \) and applying Theorem G.4, we obtain \( 1/2^{512} \)-sparseness in the QROM.

\[ \square \]

**Theorem P.2.** Suppose that the 2-DQCSD and 3-DQCSD assumptions hold and SHAKE256(·, 0x02) is quantumly-secure PRG. Then, HQC-128/196 is ANON-CCA-secure in the QROM.
Proof. Due to Theorem P.1, under the hypothesis, HQC-128/196 is SPR-CCA-secure in the QROM. Thus, applying Theorem 2.5, we have that, under those assumptions, HQC-128/196 is ANON-CCA-secure in the QROM. □

We next consider HQC’s SROB-CCA security.

**Theorem P.3.** Suppose that the 3-CQCCF assumption holds. Then, HQC is SROB-CCA-secure.

Proof (Proof sketch): Given $(1, h_{0,0}, h_{1,0})$ with $h_{0,0}, h_{1,0} \sim \mathcal{R}$, we generate decryption keys and encryption keys $ek_i = (h_{i,0}, h_{i,1})$ and $dk_i = (x_i, y_i)$ for $i \in \mathbb{Z}_0$. We give them to an adversary against SROB-CCA security of KEM. Suppose that the adversary outputs $c = (u, v)$ and the adversary wins. If so, it should be decapsulated into $K_0 \neq \perp$ and $K_1 \neq \perp$. Thus, $c$ should be decrypted into $\mu_0$ and $\mu_1$ by using $dk_0$ and $dk_1$, respectively. In re-encryption check, we have $(e_0, f_0, t_0) \leftarrow \text{SHAKE256}(G(\mu_0), 0x02)$ and $(e_1, f_1, t_1) \leftarrow \text{SHAKE256}(G(\mu_1), 0x02)$, and $u = h_{0,0} \cdot f_0 + h_{1,0} \cdot f_1 + f_1$. This implies $(1, h_{0,0}, h_{1,0}) \cdot (f_0 + f_1, t_0, t_1) = \delta$ and $(f_0 + f_1, t_0, t_1)$ is the solution of the 3-CQCCF problem. □

We finally consider the anonymity and robustness of the hybrid PKE using HQC as KEM.

**Theorem P.4.** Suppose that the 2-DQCSD, 3-DQCSD, and 3-CQCCF assumptions hold, SHAKE256(·, 0x02) is quantum-secure PRG, and HQC-DPKE is $\delta$-correct with negligible $\delta$. In addition, we assume that $1/2^{512}$ is negligible. Then, HQC-128/196 leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM, combined with SPR-\text{cRCCA}-secure and INT-\text{CTXT}-secure DEM.

Proof. Due to Theorem P.1, under the 2-DQCSD and 3-DQCSD assumptions and the assumptions on the quantum security of SHAKE256(·, 0x02) and the correctness, HQC-128/196 is SPR-\text{cRCCA}-secure and SSMT-CCA-secure in the QROM. Thus, combining HQC-128/196 with SPR-\text{cRCCA}-secure and INT-\text{CTXT}-secure DEM, we obtain a SPR-\text{cRCCA}-secure hybrid PKE in the QROM (Theorem 3.1). Moreover, HQC is SROB-CCA-secure in the QROM (Theorem P.3) under the 3-CQCCF assumption. Thus, the hybrid PKE is SROB-CCA-secure under the same assumption (Theorem 2.1). □

**Q Streamlined NTRU Prime**

**Review of Streamlined NTRU Prime:** Streamlined NTRU Prime is one of two KEMs in NTRU Prime [BBC\textsuperscript{+}20]. We briefly review Streamlined NTRU Prime. The underlying CPA-secure PKE scheme, which is called as Streamlined NTRU Prime Core\cite{20}, is summarized as follows:

- \texttt{Gen}(pp): The key-generation algorithm outputs $ek$ and $dk$.
- \texttt{Enc}(ek, $\mu$): The encryption algorithm is deterministic. Taking $\mu \in \mathcal{M}$, it outputs $c$.
- \texttt{Dec}(dk, $c$): The decryption algorithm is deterministic and outputs $\mu \in \mathcal{M}$ or special $\mu_{\text{invalid}} \in \mathcal{M}$.

Streamlined NTRU Prime [BBC\textsuperscript{+}20] applies $H_{\text{Haro}}$ to Streamlined NTRU Prime Core, where $H(\mu, c) = \text{SHA512}_{256}(\mu, c)$, $H_{\text{prf}}(s, c) = \text{SHA512}_{256}(0x00, \text{SHA512}_{256}(0x03, c, e), c)$, $F(\mu, ek) = \text{SHA512}_{256}(0x02, \text{SHA512}_{256}(0x03, \mu), \text{SHA512}_{256}(0x04, ek))$, and is defined as in Figure 24.

![Fig. 24. Streamlined NTRU Prime](image-url)

---

<table>
<thead>
<tr>
<th>Gen$(1^n)$</th>
<th>Enc$(ek)$</th>
<th>Dec$(\overline{dk}, (c_0, c_1))$, where $\overline{dk} = (dk, ek, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(ek, dk) \leftarrow \text{Gen}(1^n)$</td>
<td>$\mu \leftarrow \mathcal{M}$</td>
<td>$\mu' \leftarrow \text{Dec}(dk, c_0)$</td>
</tr>
<tr>
<td>$s \leftarrow {0, 1}^l$</td>
<td>$c_0 \leftarrow \text{Enc}(ek, \mu)$</td>
<td>if $\mu' \neq \perp$, then return $K \leftarrow H_{\text{prf}}(s, c_0, c_1)$</td>
</tr>
<tr>
<td>$\overline{dk} := (dk, ek, s)$</td>
<td>$c_1 := F(\mu, ek)$</td>
<td>$c_0' := \text{Enc}(ek, \mu')$</td>
</tr>
<tr>
<td>return $(ek, \overline{dk})$</td>
<td>$K := H(\mu, c_0, c_1)$</td>
<td>$c_1' := F(\mu', ek)$</td>
</tr>
<tr>
<td></td>
<td>return $((c_0, c_1), K)$</td>
<td>if $(c_0, c_1) = (c_0', c_1')$, then return $K := H(\mu', c_0, c_1)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>else return $K := H_{\text{prf}}(s, c_0, c_1)$</td>
</tr>
</tbody>
</table>
Security: We found that Streamlined NTRU Prime has a problem. For simplicity, let $H_3(x) = \text{SHA512}_{256}(\emptyset \parallel 01 \parallel x)$ as in [BBC+20]. Using this notation, we have
\begin{itemize}
  \item $H(\mu, c) = H_1(H_3(\mu))c$
  \item $H_{\text{prf}}(s, c) = H_0(H_3(s))c$
  \item $F(\mu, ek) = H_2(H_3(\mu))H_4(ek)$.
\end{itemize}
Using them, the conversion of Streamlined NTRU Prime is summarized as in Figure 25.

We can assume $H_3$ as random oracles. The IND-CCA security proof in the ROM is straightforward because, intuitively speaking, the adversary cannot distinguish real $K$ with random one unless it asks $\mu$ of $c_0$ to $H_3$ and the simulation of decapsulation is done by the list of queries to the random oracles. Unfortunately, we have a technical obstacle for the IND-CCA security in the QROM.

We have tried to show its security via an intermediate transform $\text{HU}^{\text{prf},r}_{\mu}[\text{PKE}, H_2, H_3, H_4]$, in which $K = H_3(\mu)$ and the decapsulation algorithm returns $\perp$ for an invalid ciphertext. If this was secure, then we can convert the security proof of $\text{HU}^{\text{prf},r}_{\mu}$ into that of $\text{HU}^{\text{prf},r}_{\mu}$ using a simple reduction. Unfortunately, $\text{HU}^{\text{prf},r}_{\mu}$ is not IND-CPA-secure because $c_1 = H_2(H_3(\mu), H_4(ek)) = H_2(K, H_4(ek))$ and we can check if $K$ is real by checking if $c_1 = H_2(K, H_4(ek))$ or not.

If $H_3$ is length-preserving, we could use the technique by Grubbs et al. [GMP21a] for QROM security proof. Unfortunately, $\mu$ is longer than 256-bits and this is not length-preserving.

If $F$ is not nested on $\mu$, we can prove the security as follows: We first consider $\text{HU}^{\mu}_{\mu}[\text{PKE}, H_3, F]$, which is SPR-CCA-secure if PKE is strongly disjoint-simulatable. We then consider an indifferentiable reduction defined as follows: if $K \neq \perp$, then we rewrite the decapsulation result as $H_2(K)(c)$; if $K = \perp$, then we rewrite the decapsulation result as $H_0(H_3(s))c$. It is easy to verify that $\text{HU}^{\mu}_{\mu}[\text{PKE}, H, F]_{\text{prf}}$ is SPR-CCA-secure if $\text{HU}^{\mu}_{\mu}[\text{PKE}, H_3, F]$ is SPR-CCA-secure.

Bernstein [Ber21] suggests to use the domain extension of quantum random oracles in [Zha19, Section 5], which is shown quantumly indifferentiable. Let $C^{H_1,H_2}(x,y) = H_1(H_2(x), y)$. Roughly speaking, we say $C^{H_1,H_2}$ is indifferentiable if any efficient adversary cannot distinguish oracles $H_1$, $H_2$, $C^{H_1,H_2}$ from $\text{Sim}^{H_1,H_2}$, where $\text{Sim}$ queries to $H$ and simulates $H_1$ and $H_2$. We did not check the detail and leave to show the IND-CCA security (and anonymity) in the QROM as an open problem.

It might be interesting to study anonymity and robustness in the ROM.

R NTRU LPRime

NTRU LPRime is the other KEM in NTRU Prime [BBC+20]. We briefly review NTRU LPRime [BBC+20] in subsection R.1, discuss the security properties of the underlying PKEs, NTRU LPRime Core and NTRU LPRime Expand, and its derandomized version, NTRU LPRime DPKE, in subsection R.2, and discuss the security properties of NTRU LPRime in subsection R.3. We want to show that, under appropriate assumptions, NTRU LPRime is ANON-CCA-secure in the QROM, and NTRU LPRime leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM. In order to do so, we show that the underlying NTRU LPRime DPKE is strongly disjoint-simulatable under appropriate assumptions in subsection R.2. NTRU LPRime is obtained by applying a variant of $\text{HU}^{\text{prf},r}_{\mu}$ to NTRU LPRime DPKE, and the strong disjoint simulatability implies that NTRU LPRime is SPR-CCA-secure and SSMT-CCA-secure in the QROM under those assumptions. We directly prove that NTRU LPRime is SFCR-CCA-secure in the QROM under an appropriate assumption. Those three properties lead to the anonymity of NTRU LPRime and hybrid PKE in the QROM as we wanted.

<table>
<thead>
<tr>
<th>Gen($1^k$)</th>
<th>Enc(ek)</th>
<th>Dec($\bar{d}k, (c_0, c_1)$), where $\bar{d}k = (dk, ek, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(ek, dk) \leftarrow \text{Gen}(1^k)$</td>
<td>$\mu \leftarrow M$</td>
<td>$\mu' \leftarrow \text{Dec}(dk, c_0)$</td>
</tr>
<tr>
<td>$s \leftarrow {0, 1}^t$</td>
<td>$c_0 \leftarrow \text{Enc}(\mu, c_0)$</td>
<td>if $\mu' = \perp$, then return $K \leftarrow H_0(H_3(s), c_0, c_1)$</td>
</tr>
<tr>
<td>$\bar{d}k \leftarrow (dk, ek, s)$</td>
<td>$c_1 \leftarrow H_2(H_3(\mu), H_4(ek))$</td>
<td>$c_0' \leftarrow \text{Enc}(\mu', c_0')$</td>
</tr>
<tr>
<td>return $(ek, \bar{d}k)$</td>
<td>$K_1 \leftarrow H_1(H_3(\mu), c_0, c_1)$</td>
<td>if $(c_0, c_1) \neq (c_0', c_1')$, then return $K \leftarrow H_0(H_3(s), c_0, c_1)$</td>
</tr>
</tbody>
</table>

Fig. 25. KEM = HU$^{\text{prf},r}_{\mu}[\text{PKE}, H_0, H_1, H_2, H_3, H_4]$. 62
R.1 Review of NTRU LPrime

NTRU LPrime has parameter sets \( p, q, w, \delta, \tau_0, \tau_1, \tau_2, \) and \( \tau_3 \). We note that \( q = 6q' + 1 \) for some \( q' \) and \( q \geq 16w + 2\delta + 3 \). For concrete values, see Table 15.

Let \( R \coloneqq \mathbb{Z}[x]/(x^p - x - 1) \) and \( R_q \coloneqq \mathbb{Z}_q[x]/(x^p - x - 1) \). Let \( S \coloneqq \{ a = \sum_{i=0}^{p-1} a_i x^i \in R \mid a_i \in \{-1, 0, +1\}, \text{HW}(a) = w \} \), a set of “short” polynomials.

For \( a \in \mathbb{Z}_q \), define \( \text{Round}(a) = 3 \cdot [a/3] \). For a polynomial \( A = \sum_{i} a_i x^i \in R_q \), we define \( \text{trunc}(A, I) = (a_0, \ldots, a_{I-1}) \in \mathbb{Z}_q^I \). For \( C \in \{0, q\} \), define \( \text{Top}(C) = (\lfloor (\tau_1(C + \tau_0 + 2^{13})/2^{15} \rfloor \right) \). For \( T \in \{0, 16\} \), define \( \text{Right}(T) = \tau_1 T - \tau_2 \in \mathbb{Z}_q \). For \( a \in \mathbb{Z} \), define \( \text{Sign}(a) = 1 \) if \( a < 0 \), otherwise.

The underlying CPA-secure PKE scheme ‘NTRU LPrime Core’ is defined as follows:

- \( \text{Gen}(pp) \): Generate \( A \leftarrow R_q \) and \( dk \leftarrow S \). Compute \( B \leftarrow \text{Round}(A \cdot dk) \). Output \( ek := (A, B) \) and \( dk \).
- \( \text{Enc}(ek, \mu \in \{0, 1\}^{256}) \): Choose \( t \leftarrow S \) and output \((U, V) := (\text{Round}(t \cdot A), \text{Top}(\text{trunc}(t \cdot B, 256) + \mu(q - 1)/2)). \)
- \( \text{Dec}(dk, (U, V)) \): Compute \( r := \text{Right}(V) + \text{trunc}(dk \cdot U, 256) + (4w + 1) \cdot 1_{256} \in \mathbb{Z}^{256} \) and outputs \( \mu := \text{Sign}(r \mod^q) \).

NTRU LPrime Core is perfectly correct.

We next consider an intermediate PKE scheme \( \text{PKE}_0 := (\text{Gen}_0, \text{Enc}_0, \text{Dec}_0) \) where the encryption algorithm uses pseudorandomness, which is called as ‘NTRU LPrime Expand’:

- \( \text{Gen}_0(pp) = \text{Gen}(pp) \):
- \( \text{Enc}_0(ek, \mu; r) \): Use \( r = \text{AES256-CTR}(r) \) to sample \( t \leftarrow S \). Output \((U, V) := \text{Enc}(ek, \mu; t) \).
- \( \text{Dec}_0(dk, (U, V)) \):

NTRU LPrime applies a variant of \( \text{HFO}_{\text{LPrime}} \) to NTRU LPrime Expand \( \text{PKE}_0 \), where \( G(\mu) = \text{SHA512}_{256}(0x05, \mu) \), \( H(\mu, c) = \text{SHA512}_{256}(0x01, \mu, c) \), \( H_{\text{pref}}(s, c) = \text{SHA512}_{256}(0x00, s, c) \), \( F(\mu, H'(ek)) = \text{SHA512}_{256}(0x02, \mu, \text{SHA512}_{256}(0x04, ek)) \) and is defined as in Figure 26.

<table>
<thead>
<tr>
<th>Table 15. Parameter sets of ntrulpr of NTRU Prime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter sets</td>
</tr>
<tr>
<td>ntrulpr653</td>
</tr>
<tr>
<td>ntrulpr761</td>
</tr>
<tr>
<td>ntrulpr857</td>
</tr>
<tr>
<td>ntrulpr953</td>
</tr>
<tr>
<td>ntrulpr1013</td>
</tr>
<tr>
<td>ntrulpr1277</td>
</tr>
</tbody>
</table>

Fig. 26. NTRU LPrime

When \( q = 6q' + 1 \), \( \text{Round}([- (q - 1)/2, (q - 1)/2]) \in [-(q - 1)/2, (q - 1)/2].\)
R.2 Properties of NTRU LPrime Core, NTRU LPrime Expand, and NTRU LPrime DPKE

Security of NTRU LPrime Core and NTRU LPrime Expand: We directly assume that NTRU LPrime Core is ciphertext-indistinguishable with simulator $S$ that samples $a \leftarrow \mathcal{R}$, computes $U \leftarrow \text{Round}(a)$, samples $V \leftarrow (\mathbb{Z}/162)^{256}$, and outputs $(U, V)$. Moreover, we assume that NTRU LPrime Core is IND-CPA-secure (and OW-CPA-secure). The IND-CPA security and ciphertext indistinguishability of NTRU LPrime Expand follows from PRG’s quantum security tightly.

**Lemma R.1.** Assume that NTRU LPrime Core is ciphertext-indistinguishable with simulator $S$ and is IND-CPA-secure, and AES256-CTR is quantumly-secure PRG. Then, NTRU LPrime Expand is strongly ciphertext-indistinguishable and IND-CPA-secure (and OW-CPA-secure).

*Proof.* Let $\mu$ denote $\text{Enc}(ek, M)$, and $\mu^*$ denote $\text{Enc}^*(ek, M)$, where $\text{Enc}^*$ is the deterministic encryption function. Then, we have $\Pr[\text{Guess}(\mu) = 1] = \Pr[\text{Guess}(\mu^*) = 1]$, which is negligible for any QPT adversary (Lemma 2.3).

R.3 Properties of NTRU LPrime

PKE $\prime \leftarrow T[PKE_0, G]$ is strongly disjoint-simulatable. Recall that $\text{HFO}_{L-prf}$ is $\text{HUF}_{L-prf} \circ T$. Applying $\text{HUF}_{L-prf}$ to PKE $\prime = T[PKE_0, G]$, we obtain KEM $= \text{HUF}_{L-prf}[\text{PKE}^*, H, F]$. After applying our theorems, we summarize the security properties of NTRU LPrime as follows:

- Assume that the underlying DPKE of NTRU LPrime PKE $\prime$ is strongly disjoint-simulatable with simulator that samples $a \leftarrow \mathcal{R}$, computes $U \leftarrow \text{Round}(a)$, samples $V \leftarrow (\mathbb{Z}/162)^{256}$, and outputs $(U, V)$.
- Then, NTRU LPrime is SPR-CCA-secure and SSMT-CCA-secure in the QROM.
- NTRU LPrime is SCFR-CCA-secure if the colliding probability of $ek$ is negligible since $F$ takes $\mu$ and $ek$ as input.
- NTRU LPrime is ANON-CCA-secure.
- NTRU LPrime leads to ANON-CCA-secure, SROB-CCA-secure hybrid PKE.

Combining NTRU LPrime DPKE’s strong disjoint-simulatability with previous theorems on $\text{HUF}_{L-prf}$, we obtain the following theorem.

**Theorem R.1.** Suppose that NTRU LPrime Core is ciphertext-indistinguishable with simulator $S$ and is IND-CPA-secure, and AES256-CTR is quantumly-secure PRG. Then, NTRU LPrime is SPR-CCA-secure and SSMT-CCA-secure in the QROM.

*Proof.* Suppose that NTRU LPrime Core is ciphertext-indistinguishable with simulator $S$ and is IND-CPA-secure, NTRU LPrime DPKE is strongly disjoint-simulatable (Lemma R.2). Applying Theorem I.2 and Theorem I.4, we obtain the theorem.

Next, we directly prove the SCFR-CCA security in the QROM. The proof is very similar to that for the modified Classic McEliece (Theorem R.3).

**Theorem R.2.** Let $\text{Col}_{\text{Gen}_i}$ be the event that when generating two keys $(ek_i, dk_i) \leftarrow \text{Gen}_0(1^n)$ for $i \in \{0, 1\}$, they collide, that is, $ek_0 = ek_1$. If $\Pr[\text{Col}_{\text{Gen}_i}]$ is negligible, then NTRU LPrime is SCFR-CCA-secure in the QROM.

*Proof.* Suppose that an adversary outputs a ciphertext $c = (c_0, c_1)$ which is decapsulated into $K \neq \bot$ by $\text{dk}_0$ and $\text{dk}_1$, that is, $\text{Dec}(\text{dk}_0, c) = \text{Dec}(\text{dk}_1, c)$. Let us define $\mu_i' = \text{Dec}_0(\text{dk}_i, c_0)$ for $i \in \{0, 1\}$. We also define $\mu_i = \mu_i'$ if $c_0 = \text{Enc}_0(ek_i, \mu_i', \text{G}(\mu_i'))$ and $c_1 = \text{F}(\mu_i', H'(ek_i))$, and $\bot$ otherwise.

We consider seven cases defined as follows:

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

5. Case 3 (\( \mu_0 = \perp \) and \( \mu_1 \neq \perp \)): In this case, the decapsulation algorithms output \( K = H_{\text{prf}}(s_0, c_0, c_1) \) and \( H(\mu_1, c_0, c_1) \) and we find a claw \((s_0, c_0, c_1), (\mu_1, c_0, c_1)\) of \( H_{\text{prf}} \) and \( H \). The probability that we find such a claw is negligible for any QPT adversary (Lemma 2.4).

6. Case 4 (\( \mu_0 \neq \perp \) and \( \mu_1 = \perp \)): In this case, the decapsulation algorithms output \( K = H(\mu_0, c_0, c_1) = H_{\text{prf}}(s_1, c_0, c_1) \) and we find a claw \((\mu_0, c_0, c_1), (s_1, c_0, c_1)\) of \( H_{\text{prf}} \) and \( H \). The probability that we find such a claw is negligible for any QPT adversary (Lemma 2.4).

7. Case 5 (The other cases): In this case, the decapsulation algorithms output \( K = H_{\text{prf}}(s_0, c_0, c_1) = H_{\text{prf}}(s_1, c_0, c_1) \) and we find a collision \((s_0, c_0, c_1), (s_1, c_0, c_1)\) of \( H_{\text{prf}} \) if \( s_0 \neq s_1 \), which occurs with probability at least \( 1 - 1/2^n \). The probability that we find such a collision is negligible for any QPT adversary (Lemma 2.3).

Thus, we conclude that the advantage of the adversary is negligible. \( \square \)

**Theorem R.3.** Suppose that NTRU LPrime Core is ciphertext-indistinguishable with simulator \( S \) and is IND-CPA-secure, and AES256-CTR is quantum-secure PRG. Then, NTRU LPrime is ANON-CCA-secure in the QROM.

**Proof.** Due to Theorem R.1, under the hypothesis, NTRU LPrime is SPR-CCA-secure in the QROM. Thus, applying Theorem 2.5, we have that, under those assumptions, NTRU LPrime is ANON-CCA-secure in the QROM. \( \square \)

**Theorem R.4.** Let \( \text{ColGen} \) be the event that when generating two keys \((e_k, d_k) \leftarrow \text{Gen}(1^k)\) for \( i \in \{0, 1\} \), they collide, that is, \( e_k \neq e_k \). Suppose that \( \text{Pr}[\text{ColGen}] \) is negligible. Suppose that NTRU LPrime Core is ciphertext-indistinguishable with simulator \( S \) and is IND-CPA-secure, and AES256-CTR is quantum-secure PRG. Then, NTRU LPrime leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM, combined with SPR-orCCA-secure and FROB-secure DEM.

**Proof.** Due to Theorem R.1, under the hypothesis, NTRU LPrime is SPR-CCA-secure and SSMT-CCA-secure in the QROM. Thus, combining NTRU LPrime with SPR-orCCA-secure DEM, we obtain a SPR-CCA-secure hybrid PKE in the QROM (Theorem 3.2). Moreover, NTRU LPrime is SCFR-CCA-secure in the QROM (Theorem R.2) under the assumption that \( \text{Pr}[\text{ColGen}] \) is negligible. Thus, if DEM is FROB-secure, then the hybrid PKE is SROB-CCA-secure (Theorem 2.2). \( \square \)

### S SIKE

We briefly review SIKE [JAC+20] in subsection S.1, discuss the security properties of the underlying PKE, SIKE-PKE, and its derandomized version, SIKE-DFPE, in subsection S.2, and discuss the security properties of SIKE in subsection S.3. We want to show that, under appropriate assumptions, SIKE is ANON-CCA-secure in the QROM, and SIKE leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM. In order to do so, we show that the underlying SIKE-DFPE of SIKE is strongly disjoint-simulatable under appropriate assumptions and XCFR-secure in subsection S.2. SIKE is obtained by applying \( U^k \) to SIKE-DFPE, and the former implies that SIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM under those assumptions and the latter implies that SIKE is SCFR-CCA-secure in the QROM. Those three properties lead to the anonymity of SIKE and hybrid PKE in the QROM as we wanted.

#### S.1 Review of SIKE

SIKE [JAC+20] is KEM scheme based on SIDH [JD11, ?]. For a survey of isogeny-based cryptography, we recommend reading [?].

Let \( p = 2^{65537} - 1 \). Let \( E \) be a supersingular elliptic curve over \( \mathbb{F}_{p^2} \). Let \( P_2, Q_2 \in E[2^2] \) and \( P_3, Q_3 \in E[3^3] \) linearly independent points of order \( 2^{22} \) and \( 3^{13} \) respectively. Let \( \{0, 1\}^n \) be a message space and let \( L : \mathbb{F}_{p^2} \rightarrow \{0, 1\}^n \) be a random oracle, instantiated by SHAKE256\(_n\)(·).

Roughly speaking, the underlying PKE scheme [JAC+20, Algorithm 1], which we call SIKE-PKE, is summarized as follows (for the details, see the specification):

\[ \text{isogeny}_\ell(dk_\ell) \text{ with } (m, \ell) = (2, 3) \text{ or } (3, 2) \text{ on } \text{Input } dk_\ell \in \{0, \ell^e\}, \text{ compute } S := P_\ell + [dk_\ell]Q_\ell, \text{ compute isogeny } \phi_\ell : E \rightarrow E/(S), \text{ and compute } E'_m := E/(S) = \phi_\ell(E). \text{ Compute } P'_m := \phi_\ell(P_m) \text{ and } Q'_m := \phi_\ell(Q_m). \text{ Output } ek_\ell := (E'_m, P'_m, Q'_m). \]

\[ \text{Correctly speaking, this algorithm outputs } (P'_m, Q'_m, R'_m) := P'_m - Q'_m \text{ and omits } E'_m. \text{ We can reconstruct } E'_m \text{ from } P'_m, Q'_m, \text{ and } R'_m. \]

65
- isoex\(_f\) \((ek_m, dk_f)\) with \((m, f) = (2, 3)\) or \((3, 2)\): On input \(ek_m = (E'_f, P'_f, Q'_f)\) and \(dk_f \in [0, \ell f)\), compute \(S := P'_f \cdot [dk_f]Q'_f\) and compute \(E''_f := E'_f / (S) = E'_f / (\phi_m(P'_f + [dk_f]Q'_f))\). Compute \(j_f\) as the \(j\)-invariant of \(E''_f\).

- \(Gen(pp):\) Choose \(dk_3 \leftarrow [0, 3^6)\) and \(ek_3 := \text{isogeny}_3(dk_3)\). Output \(ek_3\) and \(dk_3\).

- \(Enc(ek, \mu):\) Choose \(dk_2 \leftarrow [0, 2^e)\) and \(c_2 := \text{isogeny}_2(dk_2)\). Compute \(z := L(j) \oplus \mu\). Output \((c_2, z)\).

- \(Dec(dk_3, (c_2, z)):\) Compute \(j' := \text{isogeny}_3(c_2, dk_3)\) and output \(\mu' := z \oplus L(j')\).

SIKE applies FO\(_L\) to SIKE-PKE, where \(G = \text{SHAKE256}_{c_2}\) and \(H = \text{SHAKE256}_{k_3}\), and defined as in Figure 27.

<table>
<thead>
<tr>
<th>(Gen(1^n))</th>
<th>(Enc(ek))</th>
<th>(Dec(\overline{dk}, (c_2, z))) where (\overline{dk} = (dk, ek, s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>((ek, dk) \leftarrow Gen(1^n))</td>
<td>(s \leftarrow [0, 1]^n)</td>
<td>(\mu' := Dec(dk, (c_2, z)))</td>
</tr>
<tr>
<td>(r \leftarrow G(\mu, ek))</td>
<td>(c_2, z) := Enc(ek, \mu; r))</td>
<td>(c_2' := \text{isogeny}_2(r')) if (c_2 \neq c_2'), then return (K := H(s, c_2, z))</td>
</tr>
<tr>
<td>Return ((ek, \overline{dk}))</td>
<td>(K := H(\mu, c_2, z))</td>
<td>else return (K := H(\mu', c_2, z))</td>
</tr>
</tbody>
</table>

Fig. 27. SIKE

Remark S.1. SIKE’s \(Dec\) performs the test \(c_2 = c_2'\) but omits the test \(z = z'\). Since \(Dec\) retrieves \(\mu' := z \oplus k\) deterministically, we do not need to check the equality of \(z\) and \(z'\).

S.2 Properties of SIKE-PKE and SIKE-DPKE

Although we can invoke theorems on FO\(_L\) by Grubbs et al. [GMP21a] to show SIKE’s anonymity and collision-freeness, we take another way to show SIKE’s anonymity.

Assumptions: The security of SIKE is related to the following two variants of the Diffie-Hellman assumption:

Definition S.1 (Supersingular Computational Diffie-Hellman (SSCDH) Assumption [JD11], adapted). Let \(\phi_3: E \to E'_3\) be an isogeny whose kernel is equal to \((P_3 + [dk_3]Q_3\), where \(dk_3 \leftarrow [0, 3^6)\)). Let \(\phi_2: E \to E'_2\) be an isogeny whose kernel is equal to \((P_2 + [dk_2]Q_2\), where \(dk_2 \leftarrow [0, 2^e)\)).

For any QPT adversary, given the curves \(E'_2\) and \(E'_3\) and the points \(\phi_3(P_2), \phi_3(Q_2), \phi_2(P_3), \) and \(\phi_2(Q_3)\), finding the \(j\)-invariant of \(E / (P_3 + [dk_3]Q_3, P_2 + [dk_2]Q_2)\) is hard.

Definition S.2 (Supersingular Decisional Diffie-Hellman (SSDDH) Assumption [JD11], adapted). For any QPT adversary, given a tuple, it is hard to determine which distribution of the following two distributions generates the tuple:

- \((E'_3, \phi_3(P_2), \phi_3(Q_2), E'_3, \phi_2(P_3), \phi_2(Q_3), E_{23}\), where \(E'_3, \phi_3(P_2), \phi_3(Q_2), E'_3, \phi_2(P_3), \phi_2(Q_3)\) are as in the SSDCH assumption and \(E_{23} := E / (P_3 + [dk_3]Q_3, P_2 + [dk_2]Q_2)\).

- \((E'_2, \phi_2(P_3), \phi_2(Q_3), E'_2, \phi_3(P_2), \phi_3(Q_2), E_{12}\), where \(E'_2, \phi_2(P_3), \phi_2(Q_3), E'_2, \phi_3(P_2), \phi_3(Q_2)\) are as in the SSDCH assumption and \(E_{12} := E / (P_3 + [dk'_3]Q_3, P_2 + [dk'_2]Q_2)\),

where \(dk'_3 \leftarrow [0, 3^6)\) and \(dk'_2 \leftarrow [0, 2^e)\).

Security of SIKE-PKE: One can show the IND-CPA security of the underlying PKE of SIKE by assuming the SSDCH assumption and the entropy-smoothing property of L\(^9\) as that in [JD11].

Lemma S.1. Assume that the SSDCH assumption holds and L is entropy-smoothing. Then, SIKE-PKE is IND-CPA-secure (and OW-CPA-secure).

\(^9\) We borrow the notation from [FNP14]. We say a family of hash functions \(\overline{H} = (H: X \to Y)\) is entropy smoothing [IZ89] if for any (Q)PPT adversary, it is hard to distinguish \((H, H(x))\) with \((H, y)\), where \(H \leftarrow \overline{H}, x \leftarrow X,\) and \(y \leftarrow Y\).
For ciphertext indistinguishability, we construct a simulator $\mathcal{S}$ as follows: 1) sample $dk_2 \leftarrow \{0,2^{e_2}\}$ and compute $c_2 = (E'_3, P'_3, Q'_3) := \text{isogen}_g(dk_2)$; 2) sample $z \leftarrow \{0, 1\}^n$; 3) output $(c_2, z)$.

We can show that SIKE-PKE ciphertext indistinguishable with no assumptions:

**Lemma S.2.** SIKE-PKE is ciphertext indistinguishable with $\mathcal{S}$.

**Proof.** Notice that we can remove the assumption on L's property.

**Proof (Proof Sketch).** We consider two games Game$_0$ and Game$_1$.

- Game$_0$: In this game the challenge ciphertext is computes as
  \[
  \mu \leftarrow \{0, 1\}^{256}; \; dk_2 \leftarrow \{0, 2^{e_2}\}; \; c_2 := \text{isogen}_g(dk_2); \; j \leftarrow \text{isoe}_2(ek_3, dk_2); \; z := L(j) \oplus \mu; \; \text{return } (c_2, z).
  \]

- Game$_1$: In this game the challenge ciphertext is computes as
  \[
  dk_2 \leftarrow \{0, 2^{e_2}\}; \; c_2 := \text{isogen}_g(dk_2); \; z \leftarrow \{0, 1\}^{256}; \; \text{return } (c_2, z).
  \]

Game$_0$ and Game$_1$ are equivalent since $\mu$ in Game$_0$ and $z$ in Game$_1$ are uniformly at random.

**Security of SIKE-DPKE:** We next consider SIKE-DPKE obtained by applying T to SIKE-PKE.

**Lemma S.3.** Assume that the SSDDH assumption holds and $L$ is entropy-smoothing. Then, SIKE-DPKE is disjoint-simulatable with $\mathcal{S}$.

**Proof.** Statistical disjointness follows from the fact that $|S(1^n)| \approx 2^{e_2} \cdot 2^n$ and $|\text{Enc}'(ek, A)| \leq 2^n$. We can show ciphertext indistinguishability by invoking Theorem D.1 since BIKE-PKE is ciphertext-indistinguishable (Lemma S.2) and one-way (Lemma S.1).

We next consider SIKE-DPKE's collision-freeness.

**Lemma S.4.** Let $\epsilon_3$ be a probability that $e_{k_3}^0 \neq e_{k_3}^1$ holds for two keys $(e_{k_3}^0, dk_3^0)$ and $(e_{k_3}^1, dk_3^1)$ generated randomly and independently. Let $\epsilon_2$ be a probability that an efficient quantum adversary, given $(e_{k_3}^0, dk_3^0)$ and $(e_{k_3}^1, dk_3^1)$, finds $\mu$ such that $\text{isogen}_g(G(\mu, e_{k_3}^0)) = \text{isogen}_g(G(\mu, e_{k_3}^1))$. Suppose that $\epsilon := \epsilon_3 + \epsilon_2$ is negligible. Then, SIKE-DPKE is XCFR-secure.

**Proof.** The adversary against the XCFR security is given two encryption keys $e_{k_3}^0$ and $e_{k_3}^1$ with their decryption keys $dk_3^0$ and $dk_3^1$ and outputs $(c_2, z)$. If the adversary wins, then there is $\mu$ such that $dk_3^0 = G(\mu, e_{k_3}^0)$, $dk_3^1 = G(\mu, e_{k_3}^1)$, $\mu = \text{isogen}_g(dk_3^0) = \text{isogen}_g(dk_3^1)$, and $z = \mu \oplus L(j_0) = \mu \oplus L(j_1)$, where $j' := \text{iso}_{e^3}(e_{k_3}^0, dk_3^0)$ or $j' := \text{iso}_{e^3}(e_{k_3}^1, dk_3^1)$. We consider the following cases:

- Case 1 ($e_{k_3}^0 = e_{k_3}^1$): We assume that this rarely occurs by the correct choices of $dk_3^0$, $dk_3^1 \leftarrow \{0, 3^{e_3}\}$ and the probability is at most $\epsilon_3$.

- Case 2 ($e_{k_3}^0 \neq e_{k_3}^1$ and $dk_3^0 \neq dk_3^1$): This violates the collision resistance property of the quantum random oracle $G$ since $(\mu, e_{k_3}^0) \neq (\mu, e_{k_3}^1)$ and $G(\mu, e_{k_3}^0) = G(\mu, e_{k_3}^1)$.

- Case 3 ($e_{k_3}^0 \neq e_{k_3}^1$, $dk_3^0 = dk_3^1$, and $\text{isogen}_g(dk_3^0) = \text{isogen}_g(dk_3^1)$): We assume that it is hard to find $\mu$ such that $dk_3^0 = G(\mu, e_{k_3}^0)$ and $dk_3^1 = G(\mu, e_{k_3}^1)$ and the probability is at most $\epsilon_2$. Thus, in any cases, the winning probability of the adversary is negligible and we conclude the proof.

**S.3 Properties of SIKE**

Combining SIKE-DPKE’s strong disjoint-simulatability and XCFR security with previous theorems on U$_4$, we obtain the following theorems.

**Theorem S.1.** Suppose that the SSDDH assumption holds and $L$ is entropy-smoothing. Then, SIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM.

**Proof.** Under the SSDDH assumption and the assumption on $L$, SIKE-DPKE is strongly disjoint-simulatable (?!). Applying Theorem E.2 and Theorem E.3, we obtain the theorem.

**Theorem S.2.** Let $\epsilon_3$ be a probability that $e_{k_3}^0 \neq e_{k_3}^1$ holds for two keys $(e_{k_3}^0, dk_3^0)$ and $(e_{k_3}^1, dk_3^1)$ generated randomly and independently. Let $\epsilon_2$ be a probability that an efficient quantum adversary, given $(e_{k_3}^0, dk_3^0)$ and $(e_{k_3}^1, dk_3^1)$, finds $\mu$ such that $\text{isogen}_g(G(\mu, e_{k_3}^0)) = \text{isogen}_g(G(\mu, e_{k_3}^1))$. Suppose that $\epsilon := \epsilon_3 + \epsilon_2$ is negligible. Then, SIKE is SCFR-CCA-secure in the QROM.

**Proof.** Under the hypothesis, SIKE-DPKE is XCFR-secure (Lemma N.6). Applying Theorem E.4, we have that SIKE is SCFR-CCA-secure in the QROM.
Theorem S.3. Suppose that the SSDDH assumption holds and $L$ is entropy-smoothing. Then, SIKE is ANON-CCA-secure in the QROM.

Proof. Due to Theorem S.1, under the hypothesis, SIKE is SPR-CCA-secure in the QROM. Thus, applying Theorem 2.5, we have that, under those assumptions, SIKE is ANON-CCA-secure in the QROM. □

Theorem S.4. Let $\epsilon_3$ be a probability that $ek_3^0 \neq ek_3^1$ holds for two keys $(ek_3^0, dk_3^0)$ and $(ek_3^1, dk_3^1)$ generated randomly and independently. Let $\epsilon_2$ be a probability that an efficient quantum adversary, given $(ek_3^0, dk_3^0)$ and $(ek_3^1, dk_3^1)$, finds $\mu$ such that $\text{isogen}_2(G(\mu, ek_3^0)) = \text{isogen}_2(G(\mu, ek_3^1))$. Suppose that $\epsilon := \epsilon_3 + \epsilon_2$ is negligible. Suppose that the SSDDH assumption holds and $L$ is entropy-smoothing. Then, SIKE leads to ANON-CCA-secure and SROB-CCA-secure hybrid PKE in the QROM, combined with SPR-otCCA-secure and FROB-secure DEM.

Proof. Due to Theorem S.1, under the SSDDH assumption and the assumption on $L$, SIKE is SPR-CCA-secure and SSMT-CCA-secure in the QROM. Thus, combining SIKE with SPR-otCCA-secure DEM, we obtain a SPR-CCA-secure hybrid PKE in the QROM (Theorem 3.2). Moreover, SIKE is SCFR-CCA-secure in the QROM (Theorem S.2) under the hypothesis on $\epsilon$. Thus, if DEM is FROB-secure, then the hybrid PKE is SROB-CCA-secure (Theorem 2.2). □