NTRU+PKE: Efficient Public-Key Encryption Schemes from the NTRU Problem*

Jonghyun Kim† Jong Hwan Park‡

September 2, 2024

Abstract

We propose a new NTRU-based Public-Key Encryption (PKE) scheme called NTRU+PKE, which effectively incorporates the Fujisaki-Okamoto transformation for PKE (denoted as $\mathsf{FO}_{\mathsf{PKE}}$) to achieve chosen-ciphertext security in the Quantum Random Oracle Model (QROM). While NTRUEncrypt, a first-round candidate in the NIST PQC standardization process, was proven to be chosen-ciphertext secure in the Random Oracle Model (ROM), it lacked corresponding security proofs for QROM. Our work extends the capabilities of the recent ACWC_2 transformation, proposed by Kim and Park in 2023, by demonstrating that an ACWC_2-transformed scheme can serve as a sufficient foundation for applying $\mathsf{FO}_{\mathsf{PKE}}$. Specifically, we show that the ACWC_2-transformed scheme achieves (weak) γ-spreadness, an essential property for constructing an IND-CCA secure PKE scheme. Moreover, we provide the first proof of the security of $\mathsf{FO}_{\mathsf{PKE}}$ in the QROM. Finally, we show that $\mathsf{FO}_{\mathsf{PKE}}$ can be further optimized into a more efficient transformation, $\overline{\mathsf{FO}}_{\mathsf{PKE}}$, which eliminates the need for re-encryption during decryption. By instantiating an ACWC_2-transformed scheme with appropriate parameterizations, we construct NTRU+PKE, which supports 256-bit message encryption. Our implementation results demonstrate that at approximately a classical 180-bit security level, NTRU+PKE is about 2 times faster than KYBER + AES-256-GCM in AVX2 mode.

Keywords: NTRU, RLWE, Lattice-based cryptography, Post-quantum cryptography.

1 Introduction

Public-Key Encryption (PKE) is a fundamental cryptographic primitive that enables secure communication between parties without the need to share secrets in advance. In a typical PKE setup, one party (the receiver) generates a pair of public and private keys (pk, sk) and distributes the public key pk to potential senders over an authenticated channel. The sender can then use the public key pk and randomness $r \in \mathcal{R}$ to encrypt a message $m \in \mathcal{M}$, where $\mathcal{R}$ and $\mathcal{M}$ represent the spaces of randomness and messages, respectively. This encryption process is represented as $c = \mathsf{Enc}(pk, m; r)$, where Enc is the (randomized) encryption algorithm of the PKE scheme, and c is the resulting ciphertext. The receiver can then use the private key sk to decrypt the ciphertext c, either recovering the message m if c is valid, or outputting a decryption error $\perp$ if c is invalid. This decryption process is denoted as $\{m, \perp\} = \mathsf{Dec}(sk, c)$, where Dec is the decryption algorithm of the PKE scheme and $\perp$ indicates *explicit rejection*.

*This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr).

†Korea University, Seoul, Korea. Email: yoswuk@korea.ac.kr.

‡Sangmyung University, Seoul, Korea. Email: jhpark@smu.ac.kr.

For practical use of a PKE scheme, it is desirable for the message space $\mathcal{M}$ to consist of short (fixed-length) bit-strings. For instance, if $\mathcal{M} = \{0,1\}^{256}$, then the message m could be either an arbitrary *short message* encoded in fewer than 256 bits or a compressed elliptic-curve point on Curve25519. To encrypt such short messages, a PKE scheme alone is sufficient, without the need for a symmetric-key primitive known as a Data Encapsulation Mechanism (DEM). This is the primary advantage of PKE over another public-key primitive called a Key Encapsulation Mechanism (KEM). Indeed, whenever a KEM scheme is used to encrypt short messages, a hybrid KEM-DEM framework is required, which results in longer ciphertexts. Due to this advantage, PKE schemes are still worth considering for applications that need to encrypt short messages.

Indistinguishability under Chosen-Ciphertext Attack (IND-CCA) security is the widely accepted security notion for PKE. To date, several well-known methods [5, 14, 31] have been developed for constructing an IND-CCA secure PKE scheme without relying on the KEM-DEM paradigm. One such method is the Fujisaki-Okamoto (FO) transformation [14] for PKE, denoted as $\mathsf{FO}_{\mathsf{PKE}}$. In this method, for a sufficiently large message space $\mathcal{M}$, (1) a message m is concatenated with randomness r, forming a new message $m||r \in \mathcal{M}$, and (2) $m||r$ is then encrypted using new randomness $\mathsf{H}(m||r)$, generated by a hash function H, resulting in a ciphertext $c = \mathsf{Enc}(pk, m||r; \mathsf{H}(m||r))$. During decryption, the message $m||r$ is recovered, and then m is returned only if the re-encryption of $m||r$ results in the same ciphertext c. [14] showed that the resulting $\mathsf{FO}_{\mathsf{PKE}}$-transformed scheme is IND-CCA secure in the Random Oracle Model (ROM) if the underlying PKE scheme is secure against chosen-plaintext attacks (i.e., IND-CPA security).

In the context of lattice-based cryptography, $\mathsf{FO}_{\mathsf{PKE}}$ has not been widely considered an appropriate method for achieving IND-CCA security. The primary reason is that most lattice-based PKE schemes do not have a message space $\mathcal{M}$ that is large enough to contain the randomness required for $\mathsf{FO}_{\mathsf{PKE}}$. For example, Module-Lattice (ML) PKE schemes, such as the IND-CPA secure KYBER [30] and Saber [9], are specifically designed to encrypt a 256-bit message, which is insufficient for including the additional randomness needed to meet current security levels. Instead, such ML PKE schemes can be converted into IND-CCA secure KEMs by applying another FO transformation [15] for KEM (denoted $\mathsf{FO}_{\mathsf{KEM}}$), where the 256-bit (random) message is used to derive a symmetric key for a DEM. While it is theoretically possible to increase the message length in ML PKE schemes by enlarging the basic unit of the module lattice, doing so would result in $\mathsf{FO}_{\mathsf{PKE}}$-transformed PKE schemes that are significantly less efficient compared to the KEM-DEM framework.

NTRU-Based PKE Schemes Suitable for $\mathsf{FO}_{\mathsf{PKE}}$. In contrast to ML PKE schemes, NTRU-based PKE schemes [8, 23] provide message spaces that are sufficiently large to apply $\mathsf{FO}_{\mathsf{PKE}}$. For a positive integer q and a polynomial $f(x)$, let $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$ denote a polynomial-based ring for NTRU. For example, if the degree of $f(x)$ is 768, then the message space of the scheme in [23] is $\mathcal{M} = \{0,1\}^{768}$, allowing a short 256-bit message m to be encoded with 512-bit randomness r as $m||r \in \mathcal{M}$. When encrypting long messages, the $\mathsf{FO}_{\mathsf{PKE}}$-transformed NTRU-based PKE schemes can be used as part of a hybrid encryption scheme, i.e., PKE combined with DEM, where the 256-bit random message is used for DEM. This means that in the NTRU-based construction, PKE offers advantages over KEM because it (1) supports the encryption of both short and long messages and (2) avoids ciphertext expansion when encrypting short messages.

Let PKE be the underlying scheme for $\mathsf{FO}_{\mathsf{PKE}}$, and let PKE' be the resulting $\mathsf{FO}_{\mathsf{PKE}}$-transformed scheme. As mentioned above, proving the IND-CCA security of PKE' in the ROM requires PKE to be IND-CPA secure. However, PKE should also meet two additional *information-theoretic* properties: δ-correctness [19] and γ-spreadness [15]. Roughly speaking, for a fixed pair of keys (pk, sk), δ-correctness means that decrypting a valid ciphertext $c = \mathsf{Enc}(pk, m; r)$ returns m correctly, except with a negligible correctness

error δ. In a chosen-ciphertext attack against PKE', a ciphertext is generated as $c = \mathsf{Enc}(pk, m'; r')$ where $m' = m||r$ and $r' = \mathsf{H}(m||r)$. Assuming H is modeled as a random oracle, as usual, every decryption-oracle query c can be answered correctly without knowing sk by examining the adversary's queries $\{m||r\}$ to H. In this case, δ-correctness is essential to argue that so-called 'ROM-based decryption' is almost identical to real sk-based decryption. This is because ROM-based decryption has no correctness error for all decryption-oracle queries, whereas sk-based decryption has a correctness error δ. Thus, with a negligible δ (depending on a security parameter), it would be difficult for an adversary $\mathcal{A}$ to distinguish between the two types of decryption.

In general, δ-correctness considers a valid ciphertext $c = \mathsf{Enc}(pk, m; r)$ for any $m \in \mathcal{M}$ and $r \in \mathcal{R}$ as a worst-case correctness error [19]. This is particularly important for almost all NTRU-based PKE schemes,[1] because their δ-correctness depends on both m and r, which are adversarially chosen [20]. However, for a valid ciphertext $c = \mathsf{Enc}(pk, m'; r')$ by $\mathsf{FO}_{\mathsf{PKE}}$, r' is *honestly* created with $\mathsf{H}(m||r)$. Thus, the focus is on whether m' is also *honestly* generated from $m||r$. If that is the case, δ-correctness can be analyzed as an *average-case* scenario by assuming that m and r (in terms of PKE) are honestly chosen from their spaces. To resolve this problem, [8, 23] introduced their respective encoding methods called NAEP [8] and SOTP [23]: at a high level,[2] an encoded message $\mathsf{Encode}(m, r)$[3] is used as $c = \mathsf{Enc}(pk, \mathsf{Encode}(m, r); r)$, and the encoding method is also applied to PKE', where $c = \mathsf{Enc}(pk, m'; r')$ with $m' = \mathsf{Encode}(m||r, r')$ and $r' = \mathsf{H}(m||r)$. As a result, m' is controlled by Encode and r' is controlled by H, even though $\mathcal{A}$ can choose $m||r$ adversarially.

Next, γ-spreadness means that, roughly speaking, for a fixed message m and every possible ciphertext c, the probability that $\mathsf{Enc}(pk, m; r)$ maps to c for all honestly-sampled randomnesses $\{r\}$ is less than $2^{-\gamma}$. That is, with a probability of less than $2^{-\gamma}$, it is difficult to sample a specific randomness r that allows m to be encrypted into c. When generating a ciphertext $c = \mathsf{Enc}(pk, m'; r')$ in PKE', $\mathcal{A}$ must query $m||r$ to H in advance to sample the randomness $r' = \mathsf{H}(m||r)$. Otherwise, γ-spreadness implies that $\mathcal{A}$ faces difficulty in generating a valid ciphertext. Thus, in ROM-based decryption, if a decryption-oracle query c is not computed from the H-oracle queries $\{m||r\}$, c is invalid, except with a probability of $2^{-\gamma}$, and the error $\perp$ can be returned. For a probability of $2^{-\gamma} q_{\mathsf{D}}$, ROM-based decryption returns $\perp$ for all q_{D} numbers of possibly invalid ciphertexts queried by $\mathcal{A}$, which is almost identical to the sk-based decryption. Importantly, given that PKE' naturally produces $\perp$ as *explicit rejection* for an invalid ciphertext, γ-spreadness becomes even more essential to the design of a PKE scheme.

One advantage of an NTRU-based PKE scheme is that it is easy to analyze the γ-spreadness because of the simplicity of a ciphertext $c = \mathsf{Enc}(pk, m; r)$. Under $pk = \mathbf{h} \in R_q$, a ciphertext (without using the encoding method mentioned above) is computed as $\mathbf{c} = \mathbf{rh} + \mathbf{m}$ after generating two polynomials $\mathbf{r}$ and $\mathbf{m}$ (in R_q) that correspond to r and m, respectively. For a fixed m (and thus $\mathbf{m}$) and a possible $\mathbf{c}$, suppose there are two randomnesses $\mathbf{r}_1$ and $\mathbf{r}_2$ such that $\mathbf{c} = \mathbf{r}_1\mathbf{h} + \mathbf{m} = \mathbf{r}_2\mathbf{h} + \mathbf{m}$. This equality leads to the fact that $\mathbf{r}_1 = \mathbf{r}_2$ (assuming $\mathbf{h}$ is invertible in R_q), meaning that there exists at most one randomness that allows $\mathbf{m}$ (and thus m) to be mapped to $\mathbf{c}$. Therefore, for a fixed m, the γ-spreadness is bounded as the maximum probability that any randomness is sampled from $\mathcal{R}$. This contrasts with ML PKE schemes [30, 9], where two colliding ciphertexts are possible by solving a *computational* SIS (Shortest Integer Solution) problem, thereby making it difficult to analyze the γ-spreadness in an information-theoretic manner.

[1] In contrast, the δ-correctness of ML PKE schemes depends only on the randomness controlled by H, making it relatively easy to analyze their δ-correctness as an average-case one.

[2] We note that NAEP is NTRU-specific and SOTP is generic, but SOTP requires a randomness-recovery algorithm that is still NTRU-specific.

[3] To decode $\mathsf{Encode}(m, r)$, recovering r is required during decryption, which is easily done in NTRU-based PKE schemes [8, 23].

1.1 Our Results

We present an efficient NTRU-based PKE scheme that is IND-CCA secure in the (Q)ROM. The only prior work, NTRUEncrypt [38], appeared in the first round of the NIST PQC (Post-Quantum Cryptography) competition. NTRUEncrypt was constructed by (1) combining the NTRU-based PKE scheme [17] with the NAEP encoding method [21] to achieve IND-CPA security, and then (2) applying $\mathsf{FO}_{\mathsf{PKE}}$ to the combined scheme to achieve IND-CCA security. However, its security proof was analyzed in the ROM, and for security analysis in the QROM, NTRUEncrypt relied on the inefficient Q-OAEP method [34], which requires an additional length-preserving hash function. Although NTRUEncrypt was later modified to suggest a KEM in the third-round submission by merging it with NTRU-HRSS-KEM [29], it remains an open problem [8] to present an efficient NTRU-based PKE scheme that achieves IND-CCA security in the QROM.

Our construction is based on the recent work [23] that proposes a new NTRU-based KEM scheme, which we refer to as NTRU+KEM. To achieve this, [23] introduced the SOTP encoding method, by which the underlying SOTP-applied PKE has the following properties: (1) it provides $\mathcal{M} = \{0,1\}^n$ as a sufficiently large message space, (2) it enables the PKE to achieve δ-correctness and γ-spreadness, and (3) it ensures that the PKE is IND-CPA secure in the (Q)ROM. By applying $\mathsf{FO}_{\mathsf{KEM}}$ [19, 10], [23] showed that the underlying SOTP-applied PKE can be converted into a KEM that is IND-CCA secure in the (Q)ROM. Since the underlying PKE has all the properties necessary to apply $\mathsf{FO}_{\mathsf{PKE}}$, our research direction is to consider whether $\mathsf{FO}_{\mathsf{PKE}}$ is also applicable in the QROM. In doing so, our contributions are as follows:

1. We reprove that the underlying SOTP-applied PKE from [23] satisfies γ-spreadness. In [23], the Encode function, SOTP, is used as $\mathsf{Encode}(m, \mathsf{G}(r))$ with a hash function G. In the underlying PKE, a ciphertext is generated as $c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r)); r)$. To analyze γ-spreadness, the message m must be fixed for each randomness r (honestly chosen from $\mathcal{R}$). However, when using SOTP, the encoded message $\mathsf{Encode}(m, \mathsf{G}(r))$ also changes as r changes. Indeed, [23] did not consider this point, so their proof of γ-spreadness needs to be revised.

2. We reprove that $\mathsf{FO}_{\mathsf{PKE}}$ is secure against IND-CCA in the ROM. Following the analysis of $\mathsf{FO}_{\mathsf{KEM}}$ in [19], our proof clarifies that $\mathsf{FO}_{\mathsf{PKE}}$ can be proven using δ-correctness and γ-spreadness. The original proof [14] did not account for correctness errors, thereby establishing $\mathsf{FO}_{\mathsf{PKE}}$ only for an underlying PKE scheme with perfect correctness. Additionally, instead of relying on γ-spreadness, the original proof relied on the notion of *plaintext-awareness* [5]. Our revised proof of $\mathsf{FO}_{\mathsf{PKE}}$ is applicable to an NTRU-based PKE scheme (with a worst-case correctness error) and is easier to understand.

3. We prove that $\mathsf{FO}_{\mathsf{PKE}}$ is secure against IND-CCA in the QROM. Following the security analysis of $\mathsf{FO}_{\mathsf{KEM}}$ in the QROM [10], our proof uses an extractable random oracle simulator. This simulator overcomes the challenge of measuring inputs in the QROM without significantly disturbing the quantum states, which is essential for simulating decryption queries with explicit rejection without using the secret key. Furthermore, to bound the advantage of an IND-CCA adversary, who operates in a two-phase game, we utilize the two-phase Adaptive One-way-to-Hiding (AO2H) Lemma [35]. As in the ROM, the security proof in the QROM also relies on δ-correctness and γ-spreadness.

4. We apply $\mathsf{FO}_{\mathsf{PKE}}$ to the underlying SOTP-applied PKE to obtain an IND-CCA secure PKE scheme in the (Q)ROM. As in [23], we further show that there is an equivalent transform, $\overline{\mathsf{FO}}_{\mathsf{PKE}}$, which works identically to $\mathsf{FO}_{\mathsf{PKE}}$ but is more efficient because it eliminates re-encryption during decryption. We revise the previous proof [23] of this equivalence to be more rigorous, based on the *rigidity* properties of both PKE [6] and SOTP [23]. Consequently, by using $\overline{\mathsf{FO}}_{\mathsf{PKE}}$, the underlying SOTP-applied PKE is transformed into its final form as an IND-CCA secure PKE scheme in the (Q)ROM.

5. We instantiate the underlying SOTP-applied PKE by adapting the constructions of PKE and SOTP from [23]. Since the underlying PKE is based on a basic NTRU setting, we refer to the final form obtained through $\overline{\mathsf{FO}}_{\mathsf{PKE}}$ as 'NTRU+PKE'. Following [23], we propose four parameter sets for NTRU+PKE, with some modifications, to encrypt 256-bit messages. We implement NTRU+PKE using these four parameter sets and compare the performance against 'KYBER + AES-256-GCM', where KYBER is a KEM and AES-256-GCM is a DEM for 256-bit message encryption. Our results show that, at an approximate 180-bit classical security level, a single encryption/decryption with NTRU+PKE is about 2 times faster than with KYBER + AES-256-GCM in AVX2 mode.

1.2 Related Works

In 1998, the first NTRU encryption scheme was proposed in the literature by Hoffstein, Pipher, and Silverman [17]. In their initial construction, $\mathcal{M}$ and $\mathcal{R}$ were defined as $\{-1, 0, 1\}$ with certain distributions, which were not naturally suited to accommodate arbitrary bit-string messages from $\mathcal{M}$. Several padding schemes [18, 27] were suggested for encrypting (short) bit-string messages, based on the OAEP [5] scheme and the $\mathsf{FO}_{\mathsf{PKE}}$ [14] transformation. However, these padding schemes were shown to be insecure [20] due to the ability to manipulate messages, leading to decryption failures. In other words, these padding schemes did not account for worst-case correctness errors.

In 2003, Howgrave-Graham *et al.* [21] proposed an encoding method called NAEP, which is used to generate $c = \mathsf{Enc}(pk, \mathsf{NAEP}(m, G(r)); r)$ for a bit-string message m. The original NAEP was designed to output a polynomial with binary coefficients, but it was later revised to produce a ternary polynomial with coefficients sampled *uniformly* from $\{-1, 0, 1\}$ [16]. This uniformity within $\{-1, 0, 1\}$ is essential for hiding information about m from c. Regarding the underlying NAEP-applied PKE, (worst-case) δ-correctness was estimated in [16], but γ-spreadness has not yet been clearly analyzed. Nevertheless, by applying $\mathsf{FO}_{\mathsf{PKE}}$, the underlying NAEP-applied PKE was transformed into NTRUEncrypt, which has already been standardized according to IEEE P1363.1 [12] and ANSI X9.98 [4], and submitted to the NIST PQC competition process [38]. However, as mentioned earlier, the IND-CCA security of NTRUEncrypt was proven only in the ROM, not in the QROM.

In 2021, Duman *et al.* [11] proposed two generic transformations, ACWC_0 and ACWC, which make the average-case correctness error of an underlying scheme PKE^0 nearly equal to the worst-case correctness error of a transformed scheme PKE. ACWC_0 requires ciphertext expansion in addition to the ciphertext of PKE^0, whereas ACWC does not. Instead, ACWC relies on an encoding function, GOTP, exemplified by the One-Time Pad modulo 3 in [11], meaning that uniformity within $\{-1, 0, 1\}$ is still required, as in NAEP. In 2022, Kim *et al.* [23] proposed another generic transformation, ACWC_2, based on SOTP. Unlike NAEP and GOTP, SOTP is realized using a centered binomial distribution (CBD) in $\{-1, 0, 1\}$, which is straightforward to implement in constant time. Using $\mathsf{FO}_{\mathsf{KEM}}$ and their claims that the underlying $\{\mathsf{ACWC}_0, \mathsf{ACWC}, \mathsf{ACWC}_2\}$-transformed PKEs satisfy γ-spreadness, [11, 23] proposed their respective NTRU-based KEMs with explicit rejection and proved their security against IND-CCA in the (Q)ROM.

Recently, Fouque *et al.* [13] proposed a new NTRU-based KEM, called BAT, which reduces the size of a ciphertext. Instead of generating a typical NTRU ciphertext as $\mathbf{c} = \mathbf{rh} + \mathbf{m}$, BAT encrypts a message m as $\mathbf{c} = (\lfloor \mathbf{rh} \rceil, \mathsf{G}(\mathbf{r}) \oplus m)$ using a rounding operation $\lfloor \cdot \rceil$ and ACWC_0. For the secret key sk, BAT requires the generation of an NTRU trapdoor basis [28], which is used to find the NTRU-lattice point $\mathbf{rh}$ closest to $\lfloor \mathbf{rh} \rceil$. Consequently, the key generation algorithm of BAT is much slower than that of other NTRU-based KEMs. More recently, Zhang *et al.* [37] introduced another NTRU-based KEM, called NEV, which aims to reduce the modulus q of $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ and, consequently, the size of a ciphertext. The key idea behind NEV is to devise a novel NTRU-based key structure that integrates the vector decoding method [2] into the

NTRU variant by Stehle and Steinfeld [33]. Although both BAT and NEV provide compact ciphertext sizes, their proposed parameter sets are very limited in meeting the desired security levels because they use the polynomial $f(x) = x^n + 1$ with a power-of-2 n. Additionally, in terms of γ-spreadness, it is difficult for them to achieve an appropriate γ for similar reasons as in ML PKE schemes, meaning that $\mathsf{FO}_{\mathsf{PKE}}$ is not applicable to BAT and NEV.

2 Preliminaries

2.1 Basic Notations

The set $\mathbb{Z}_q$ is defined as $\{-(q-1)/2, \ldots, (q-1)/2\}$, where q is a positive odd integer. Mapping an integer a from $\mathbb{Z}$ to $\mathbb{Z}_q$ uses the modulo operation, setting $x = a \bmod q$ as the unique integer in $\mathbb{Z}_q$ satisfying $q \mid (x - a)$. The polynomial ring R_q is defined as $\mathbb{Z}_q[x]/\langle f(x)\rangle$ with a polynomial $f(x)$. Cyclotomic trinomials $\Phi_{3n}(x) = x^n - x^{n/2} + 1$ where $n = 2^i \cdot 3^j$ for some positive integers i and j are used as $f(x)$ in our construction. Polynomials in R_q are denoted in non-italic bold as $\mathbf{a}$, with $\mathbf{a}_i$ as the i-th coefficient.

For sampling, $u \leftarrow X$ indicates that u is sampled uniformly at random from a set X, and $u \leftarrow D$ indicates that u is drawn according to a distribution D. The notation $u \leftarrow D^\ell$ forms a vector $u = (u_1, \ldots, u_\ell)$ with each u_i drawn independently from D. Especially, $\mathbf{a} \leftarrow D$ indicates that all coefficients of a polynomial $\mathbf{a}$ is drawn according to a distribution D. Sampling from the centered binomial distribution (CBD) ψ_k involves $2k$ bits that are independent and uniformly random, summing the first k bits and the second k bits separately, then outputting their difference.

2.2 Definition of PKE and Related Properties

Definition 2.1 (Public-Key Encryption)**.** A public key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$, randomness space $\mathcal{R}$, and ciphertext space $\mathcal{C}$ consists of the following three algorithms:

- $\mathsf{Gen}(1^\lambda)$: The key generation algorithm Gen is a randomized algorithm that takes a security parameter 1^λ as input and outputs a pair of public/secret keys (pk, sk).
- $\mathsf{Enc}(pk, m; r)$: The encryption algorithm Enc is a randomized algorithm that takes a public key pk, a message $m \in \mathcal{M}$, and randomness $r \in \mathcal{R}$ as input and outputs a ciphertext $c \in \mathcal{C}$. We often write $\mathsf{Enc}(pk, m)$ to denote the encryption algorithm without explicitly mentioning the randomness.
- $\mathsf{Dec}(sk, c)$: The decryption algorithm Dec is a deterministic algorithm that takes a secret key sk and a ciphertext $c \in \mathcal{C}$ as input and outputs either a message $m \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$ to indicate that c is not a valid ciphertext.

Correctness. PKE has (worst-case) correctness error δ [19] if

$$\mathbb{E}\left[\max_{m \in \mathcal{M}} \Pr[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m]\right] \leq \delta,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and the choice of the random oracles involved (if any). PKE has average-case correctness error δ relative to distribution $\psi_{\mathcal{M}}$ over $\mathcal{M}$ if

$$\mathbb{E}\left[\Pr\left[\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) \neq m\right]\right] \leq \delta,$$

where the expectation is taken over $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, the choice of the random oracles involved (if any), and $m \leftarrow \psi_{\mathcal{M}}$.

GAME INJ

1: $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(m, r, m', r') \leftarrow \mathcal{A}(pk)$
3: $c = \mathsf{Enc}(pk, m; r)$
4: $c' = \mathsf{Enc}(pk, m'; r')$
5: **return** $[\![(m, r) \neq (m', r') \wedge c = c']\!]$

Figure 1: GAME INJ for PKE

Injectivity. Injectivity of PKE is defined via the following GAME INJ, which is shown in Figure 1, and the relevant advantage of adversary $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{INJ}}_{\mathsf{PKE}}(\mathcal{A}) = \Pr[\mathsf{IND}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1].$$

Unlike the definition of injectivity in [7, 19], it is defined in a computationally secure sense in this work.

Spreadness. PKE is γ-*spread* [19] if

$$\min_{m \in \mathcal{M}, (sk,pk)} \left(-\log \max_{c \in \mathcal{C}} \Pr_{r \leftarrow \psi_{\mathcal{R}}} [c = \mathsf{Enc}(pk, m; r)] \right) \geq \gamma,$$

where the minimum is taken over all key pairs that can be generated by Gen. This definition can be relaxed by considering an expectation over the choice of (pk, sk). PKE is *weakly* γ-*spread* [10] if

$$-\log \mathbb{E} \left[\max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr_{r \leftarrow \psi_{\mathcal{R}}} [c = \mathsf{Enc}(pk, m; r)] \right] \geq \gamma,$$

where the expectation is over $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$.

Randomness recoverability. PKE is defined as randomness recoverable (RR) if there is an algorithm RRec such that for all $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, $m \in \mathcal{M}$, and $r \in \mathcal{R}$,

$$\Pr \Big[\forall m' \in \mathsf{Pre}^m(pk, c) : \mathsf{RRec}(pk, m', c) \notin \mathcal{R} \\ \vee \mathsf{Enc}(pk, m'; \mathsf{RRec}(pk, m', c)) \neq c | c \leftarrow \mathsf{Enc}(pk, m; r) \Big] = 0,$$

where the probability is taken over $c \leftarrow \mathsf{Enc}(pk, m; r)$ and $\mathsf{Pre}^m(pk, c)$ defined as $\{m \in \mathcal{M} | \exists r \in \mathcal{R} : \mathsf{Enc}(pk, m; r) = c\}$.

Message Recoverability. PKE is defined as message recoverable (MR) if an algorithm MRec exists such that for all $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, $m \in \mathcal{M}$, and $r \in \mathcal{R}$,

$$\Pr \Big[\forall r' \in \mathsf{Pre}^r(pk, c) : \mathsf{MRec}(pk, r', c) \notin \mathcal{M} \\ \vee \mathsf{Enc}(pk, \mathsf{MRec}(pk, r', c); r') \neq c | c \leftarrow \mathsf{Enc}(pk, m; r) \Big] = 0,$$

where the probability is calculated over $c \leftarrow \mathsf{Enc}(pk, m; r)$ and $\mathsf{Pre}^r(pk, c)$ defined as $\{r \in \mathcal{R} | \exists\, m \in \mathcal{M} : \mathsf{Enc}(pk, m; r) = c\}$.

Rigidity. PKE is said to be *rigid* if, for all key pairs $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and for any ciphertext $c \in \mathcal{C}$, the following holds:

$$\text{If } m' = \mathsf{Dec}(sk, c) \in \mathcal{M} \text{ and } r' = \mathsf{RRec}(pk, m', c) \in \mathcal{R}, \text{ then } \mathsf{Enc}(pk, m'; r') = c.$$

2.3 Security of PKE

Definition 2.2 (OW-CPA Security of PKE)**.** Let PKE = (Gen, Enc, Dec) be a public-key encryption scheme with message space $\mathcal{M}$. Onewayness under chosen-plaintext attacks (OW-CPA) for message distribution $\psi_{\mathcal{M}}$ is defined via the GAME OW-CPA, which is shown in Figure 2, and the advantage function of adversary $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr\left[\mathsf{OW\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right].$$

Definition 2.3 (IND-CPA Security of PKE)**.** Let PKE = (Gen, Enc, Dec) be a public-key encryption scheme with message space $\mathcal{M}$. Indistinguishability under chosen-plaintext attacks (IND-CPA) is defined via the GAME IND-CPA, as shown in Figure 2, and the advantage function of adversary $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr\left[\mathsf{IND\text{-}CPA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] - \frac{1}{2}\right|.$$

Definition 2.4 (IND-CCA Security of PKE)**.** Let PKE = (Gen, Enc, Dec) be a public-key encryption scheme with message space $\mathcal{M}$. Indistinguishability under chosen ciphertext attacks (IND-CCA) is defined via the GAME IND-CCA, as shown in Figure 2, and the advantage function of adversary $\mathcal{A}$ is

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}}(\mathcal{A}) := \left|\Pr\left[\mathsf{IND\text{-}CCA}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] - \frac{1}{2}\right|.$$

GAME OW-CPA
1: $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $m \leftarrow \psi_{\mathcal{M}}$
3: $c^* \leftarrow \mathsf{Enc}(pk, m)$
4: $m' \leftarrow \mathcal{A}(pk, c^*)$
5: **return** $[\![m = m']\!]$

GAME IND-CPA
1: $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(m_0, m_1) \leftarrow \mathcal{A}_0(pk)$
3: $b \leftarrow \{0, 1\}$
4: $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
5: $b' \leftarrow \mathcal{A}_1(pk, c^*)$
6: **return** $[\![b = b']\!]$

GAME IND-CCA
1: $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
2: $(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{Dec}}(pk)$
3: $b \leftarrow \{0, 1\}$
4: $c^* \leftarrow \mathsf{Enc}(pk, m_b)$
5: $b' \leftarrow \mathcal{A}_1^{\mathsf{Dec}}(pk, c^*)$
6: **return** $[\![b = b']\!]$

$\mathsf{Dec}(c \neq c^*)$
1: **return** $\mathsf{Dec}(sk, c)$

Figure 2: GAMES OW-CPA, IND-CPA, and IND-CCA for PKE

2.4 Complexity Assumptions

Definition 2.5 (The NTRU problem [17])**.** Let ψ be a distribution over R_q. The NTRU problem $\mathsf{NTRU}_{n,q,\psi}$ is to distinguish $\mathbf{h} = \mathbf{g}(p\mathbf{f}' + 1)^{-1} \in R_q$ from $\mathbf{u} \in R_q$, where $\mathbf{f}', \mathbf{g} \leftarrow \psi$ and $\mathbf{u} \leftarrow R_q$. The advantage of adversary $\mathcal{A}$ in solving $\mathsf{NTRU}_{n,q,\psi}$ is defined as follows:

$$\mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\psi}(\mathcal{A}) = \Pr[\mathcal{A}(\mathbf{h}) = 1] - \Pr[\mathcal{A}(\mathbf{u}) = 1].$$

Definition 2.6 (The RLWE problem [25])**.** Let ψ be a distribution over R_q. The RLWE problem $\mathsf{RLWE}_{n,q,\psi}$ is to find $\mathbf{s}$ from $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e}) \in R_q \times R_q$, where $\mathbf{a} \leftarrow R_q$, $\mathbf{s}, \mathbf{e} \leftarrow \psi$. The advantage of an adversary $\mathcal{A}$ in solving $\mathsf{RLWE}_{n,q,\psi}$ is defined as follows:

$$\mathsf{Adv}^{\mathsf{RLWE}}_{n,q,\psi}(\mathcal{A}) = \Pr[\mathcal{A}(\mathbf{a}, \mathbf{b}) = \mathbf{s}].$$

2.5 One-way to Hiding

The O2H lemma, first introduced by Unruh [36], serves as a important proof tool for the QROM. This lemma quantifies the advantage of a quantum adversary in distinguishing between two scenarios: one that uses random oracle outputs for specific inputs and another that uses truly random values. The fundamental idea is that the probability of an adversary successfully measuring the specific input, for which the hash function output has been replaced with a truly random value, bounds the advantage between these two scenarios. In the ROM, the corresponding concept is the difference lemma proposed by Shoup [32], which similarly analyzes the differences between two games but is applicable in a classical context. This subsection outlines the variations of the O2H lemma used in the security proofs presented in this work.

Lemma 2.7 (Adaptive O2H, Lemma 14 of [35])**.** Let $\mathsf{H} : \{0,1\}^* \to \{0,1\}^n$ be a random oracle. Consider an oracle algorithm $\mathcal{A}_1$ that uses the final state of $\mathcal{A}_0$ and makes at most q_1 queries to H. Let $\mathcal{C}_1$ be an oracle algorithm that on input (j, B, x) does the following: run $\mathsf{A}_1^{\mathsf{H}}(x, B)$ until (just before) the j-th query, measure the argument of the query in the computational basis, output the measurement outcome. (When $\mathcal{A}$ makes less than j queries, $\mathcal{C}_1$ outputs $\perp \notin \{0,1\}^*$.) Let

$$\begin{aligned}
P^1_{\mathcal{A}} &:= \Pr[b' = 1 : \mathsf{H} \leftarrow (\{0,1\}^* \to \{0,1\}^n), m \leftarrow \mathcal{A}_0^{\mathsf{H}}(), x \leftarrow \{0,1\}^\ell, \\
&\qquad\qquad b' \leftarrow \mathcal{A}_1^{\mathsf{H}}(x, \mathsf{H}(x\|m))], \\
P^2_{\mathcal{A}} &:= \Pr[b' = 1 : \mathsf{H} \leftarrow (\{0,1\}^* \to \{0,1\}^n), m \leftarrow \mathcal{A}_0^{\mathsf{H}}(), x \leftarrow \{0,1\}^\ell, \\
&\qquad\qquad B \leftarrow \{0,1\}^n, b' \leftarrow \mathcal{A}_1^{\mathsf{H}}(x, B)], \\
P_{\mathcal{C}} &:= \Pr[x = x' \wedge m = m' : \mathsf{H} \leftarrow (\{0,1\}^* \to \{0,1\}^n), m \leftarrow \mathcal{A}_0^{\mathsf{H}}(), x \leftarrow \{0,1\}^\ell, \\
&\qquad\qquad B \leftarrow \{0,1\}^n, j \leftarrow \{1, ..., q_1\}, x'||m' \leftarrow \mathcal{C}_1^{\mathsf{H}}(j, B, x)].
\end{aligned}$$

Then $|P^1_{\mathcal{A}} - P^2_{\mathcal{A}}| \leq 2q_1\sqrt{P_{\mathcal{C}}} + q_0 2^{-\ell/2+2}$.

Lemma 2.8 (Classical O2H, Theorem 3 from the eprint version of [3])**.** Let $S \subset \mathcal{R}$ be random. Let G and F be random functions satisfying $\forall r \notin S : \mathsf{G}(r) = \mathsf{F}(r)$. Let z be a random classical value (S, G, F, z may have an arbitrary joint distribution). Let $\mathcal{C}$ be a quantum oracle algorithm with query depth q_{G}, expecting input z. Let $\mathcal{D}$ be the algorithm that, on input z, samples a uniform i from $\{1, ..., q_{\mathsf{G}}\}$, runs $\mathcal{C}$ right before its i-th query to F, measures all query input registers, and outputs the set T of measurement outcomes. Then

$$\left|\Pr[\mathcal{C}^{\mathsf{G}}(z) \Rightarrow 1] - \Pr[\mathcal{C}^{\mathsf{F}}(z) \Rightarrow 1]\right| \leq 2q_{\mathsf{G}}\sqrt{\Pr[S \cap T \neq \emptyset : T \leftarrow \mathcal{D}^{\mathsf{F}}(z)]}.$$

2.6 Extractable RO-Simulator $\mathcal{S}$

The extractable random oracle simulator [10] is another important proof tool for security proofs in QROM. It addresses challenges in retrieving hash inputs from superpositioned queries. This random oracle simulator is indistinguishable from a real random oracle and can extract queried inputs under specific conditions, thereby enabling security proofs in the QROM settings.

Definition 2.9. For a function $f : \mathcal{X} \times \{0,1\}^n \to \mathcal{T}$, define

$$\Gamma(f) := \max_{x,t} |\{y \mid f(x,y) = t\}| \text{ and } \Gamma'(f) := \max_{x \neq x', y'} \left|\{y \mid f(x,y) = f(x',y')\}\right|.$$

Theorem 2.10 (Theorem 4.3 of [10])**.** The extractable RO-simulator $\mathcal{S}$ constructed above, with interfaces $\mathcal{S}.RO$ and $\mathcal{S}.E$, satisfies the following properties.

1. If $\mathcal{S}.E$ is unused, $\mathcal{S}$ is perfectly indistinguishable from the random oracle RO.
2. (a) Any two subsequent independent queries to $\mathcal{S}.RO$ commute. In particular, two subsequent classical $\mathcal{S}.RO$-queries with the same input x give identical responses.
   (b) Any two subsequent independent queries to $\mathcal{S}.E$ commute. In particular, two subsequent classical $\mathcal{S}.E$-queries with the same input t give identical responses.
   (c) Any two subsequent independent queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$ $8\sqrt{2\Gamma(f)/2^n}$-almost-commute.
3. (a) Any classical query $\mathcal{S}.RO(x)$ is idempotent.
   (b) Any classical query $\mathcal{S}.E(t)$ is idempotent.
4. (a) If $\hat{x} = \mathcal{S}.E(t)$ and $\hat{h} = \mathcal{S}.RO(\hat{x})$ are two subsequent classical queries then
   $$\Pr[f(\hat{x},\hat{h}) \neq t \wedge \hat{x} \neq \emptyset] \leq Pr[f(\hat{x},\hat{h}) \neq t | \hat{x} \neq \emptyset] \leq 2 \cdot 2^{-n}\Gamma(f).$$
   (b) If $h = \mathcal{S}.RO(x)$ and $\hat{x} = \mathcal{S}.E(f(x,h))$ are two subsequent classical queries such that no prior query to $\mathcal{S}.E$ has been made, then
   $$\Pr[\hat{x} = \emptyset] \leq 2 \cdot 2^{-n}.$$

Furthermore, the total runtime of $\mathcal{S}$, when implemented using the sparse representation of the compressed oracle, is bounded as

$$T_{\mathcal{S}} = O(q_{RO} \cdot q_E \cdot \text{Time}[f] + q_{RO}^2),$$

where q_E and q_{RO} are the number of queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$, respectively.

Theorem 2.11 (Proposition 4.4. of [10])**.** Let $R' \subseteq \mathcal{X} \times \mathcal{T}$ be a relation. Consider a query algorithm $\mathcal{A}$ that makes q queries to the $\mathcal{S}.RO$ interface of $\mathcal{S}$ but no query to $\mathcal{S}.E$, outputting some $\mathbf{t} \in T^\ell$. For each i, let $\hat{x}_i$ then be obtained by making an additional query to $\mathcal{S}.E$ on input t_i. Then

$$\Pr_{\mathbf{t} \leftarrow \mathcal{A}^{\mathcal{S}.RO}, \hat{x}_i \leftarrow \mathcal{S}.E(t_i)}[\exists i : (\hat{x}_i, t_i) \in R'] \leq 128 \cdot q^2 \Gamma_R / 2^n,$$

where $R \subseteq \mathcal{X} \times \mathcal{Y}$ is the relation $(x,y) \in R \Leftrightarrow (x, f(x,y)) \in R'$ and

$$\Gamma_R := \max_{x \in \mathcal{X}} |\{y \in \{0,1\}^n | (x,y) \in R\}|.$$

3 ACWC$_2$ Transformation

3.1 Semi-Generalized One-Time Pad (SOTP)

Following [23], we define the semi-generalized one-time pad (SOTP), a variant of the generalized one-time pad (GOTP) from [11]. See Figure 16 for an example of SOTP.

Definition 3.1. A semi-generalized one-time pad $\mathsf{SOTP} = (\mathsf{Encode}, \mathsf{Inv})$ with a message space $\mathcal{X}$, a random space $\mathcal{U}$, and a code space $\mathcal{Y}$ (with respective corresponding distributions $\psi_{\mathcal{X}}$, $\psi_{\mathcal{U}}$, and $\psi_{\mathcal{Y}}$) consists of the following two algorithms:

- $\mathsf{Encode}(x, u)$: The encoding algorithm Encode is a deterministic algorithm that takes a message $x \in \mathcal{X}$ and random $u \in \mathcal{U}$ as input, and outputs a code $y \in \mathcal{Y}$.
- $\mathsf{Inv}(y, u)$: The decoding algorithm Inv is a deterministic algorithm that takes a code $y \in \mathcal{Y}$ and random $u \in \mathcal{U}$ as input, and outputs a message $x \in \mathcal{X} \cup \{\perp\}$.

It also satisfies the following three properties:

1. **Decoding:** For all $x \in \mathcal{X}$ and $u \in \mathcal{U}$, $\mathsf{Inv}(\mathsf{Encode}(x, u), u) = x$.
2. **Message hiding:** For all $x \in \mathcal{X}$, the random variable $\mathsf{Encode}(x, u)$, for $u \leftarrow \psi_{\mathcal{U}}$, has the same distribution as $\psi_{\mathcal{Y}}$.
3. **Rigidity:** For all $u \in \mathcal{U}$ and $y \in \mathcal{Y}$ with $\mathsf{Inv}(y, u) \neq \perp$, $\mathsf{Encode}(\mathsf{Inv}(y, u), u) = y$.

3.2 Description of ACWC$_2$ Transformation

We describe the ACWC$_2$ transformation [23], denoted as $\mathsf{PKE}' = \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$. First, let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an underlying PKE scheme with message space $\mathcal{M}$ and randomness space $\mathcal{R}$, where a message $M \in \mathcal{M}$ and randomness $r \in \mathcal{R}$ are drawn from distributions $\psi_{\mathcal{M}}$ and $\psi_{\mathcal{R}}$, respectively. Additionally, assume that PKE includes a randomness recovery algorithm RRec and a message recovery algorithm MRec. Define $\mathsf{SOTP} = (\mathsf{Encode}, \mathsf{Inv})$ as a semi-generalized one-time pad with distributions $(\psi_{\mathcal{U}}, \psi_{\mathcal{M}}, \psi_{\mathcal{M}'})$, such that $\mathsf{Encode} : \mathcal{M}' \times \mathcal{U} \to \mathcal{M}$ and $\mathsf{Inv} : \mathcal{M} \times \mathcal{U} \to \mathcal{M}'$. Additionally, let $\mathsf{G} : \mathcal{R} \to \mathcal{U}$ be a hash function where every output is independently $\psi_{\mathcal{U}}$-distributed. Next, let $\mathsf{PKE}' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ be

$\underline{\mathsf{Gen}'(1^\lambda)}$
1: $(pk, sk) := \mathsf{Gen}(1^\lambda)$
2: **return** (pk, sk)

$\underline{\mathsf{Enc}'(pk, m \in \mathcal{M}'; R \in \mathcal{R}')}$
1: $r \leftarrow \psi_{\mathcal{R}}$ using the randomness R
2: $M := \mathsf{Encode}(m, \mathsf{G}(r))$
3: $c := \mathsf{Enc}(pk, M; r)$
4: **return** c

$\underline{\mathsf{Dec}'(sk, c)}$
1: $M := \mathsf{Dec}(sk, c)$
2: $r := \mathsf{RRec}(pk, M, c)$
3: $m := \mathsf{Inv}(M, \mathsf{G}(r))$
4: **if** $r \notin \mathcal{R}$ or $m = \perp$
5: **return** $\perp$
6: **return** m

Figure 3: $\mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$

the transformed PKE scheme with message space $\mathcal{M}'$ and randomness space $\mathcal{R}'$, where a message $m \in \mathcal{M}'$ and randomness $R \in \mathcal{R}'$ are drawn from distributions $\psi_{\mathcal{M}'}$ and $\psi_{\mathcal{R}'}$, respectively. The ACWC_2-transformed scheme $\mathsf{PKE}' = \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ is described in Figure 3.

Theorem 3.2 (Average-Case to Worst-Case Correctness Error [23])**.** Let PKE be RR and have a randomness space $\mathcal{R}$ relative to the distribution $\psi_{\mathcal{R}}$. Let $\mathsf{SOTP} = (\mathsf{Encode}, \mathsf{Inv})$ be a semi-generalized one-time pad described above, and let G be a hash function (mentioned above) modeled as a random oracle. If PKE is δ-average-case-correct, then $\mathsf{PKE}' := \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ is δ'-worst-case-correct for

$$\delta' = \delta + \|\psi_{\mathcal{R}}\| \cdot \left(1 + \sqrt{(\ln|\mathcal{M}'| - \ln\|\psi_{\mathcal{R}}\|)/2}\right),$$

where $\|\psi_{\mathcal{R}}\| := \sqrt{\sum_r \psi_{\mathcal{R}}(r)^2}$.

Theorem 3.3 (OW-CPA of $\mathsf{PKE} \overset{\text{ROM}}{\Longrightarrow}$ IND-CPA of $\mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ [23])**.** Let PKE be a public-key encryption scheme with RR and MR properties. For any adversary $\mathcal{A}$ against the IND-CPA security of $\mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$, making at most q_{G} random oracle queries, there exists an adversary $\mathcal{B}$ against the OW-CPA security of PKE and an adversary $\mathcal{C}$ against the injectivity of PKE such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_2[\mathsf{PKE},\mathsf{SOTP},\mathsf{G}]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{INJ}}_{\mathsf{PKE}}(\mathcal{C}),$$

where the running time of $\mathcal{B}$ is about $\mathrm{Time}(\mathcal{A}) + O(q_{\mathsf{G}})$.

Theorem 3.4 (OW-CPA of $\mathsf{PKE} \overset{\text{QROM}}{\Longrightarrow}$ IND-CPA of $\mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ [23])**.** Let PKE be a public-key encryption scheme with RR and MR properties. For any quantum adversary $\mathcal{A}$ against the IND-CPA security of $\mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ with a query depth of at most q_{G}, there exists a quantum adversary $\mathcal{B}$ against the OW-CPA security of PKE and an adversary $\mathcal{C}$ against the injectivity of PKE such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{ACWC}_2[\mathsf{PKE},\mathsf{SOTP},\mathsf{G}]}(\mathcal{A}) \leq 2q_{\mathsf{G}}\sqrt{\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{INJ}}_{\mathsf{PKE}}(\mathcal{C})},$$

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

3.3 New Proof on the γ-Spreadness of PKE'

Our approach leverages the fact that, assuming G is modeled as a random oracle, the encoded message $M = \mathsf{Encode}(m, G(r))$ is $\psi_{\mathcal{M}}$-distributed due to the message hiding property of SOTP. In general, proofs in the ROM can exploit the random choice of G [22], and thus, for a fixed message m, we can treat M as a random variable chosen according to $\psi_{\mathcal{M}}$. In this context, the γ'-spreadness of PKE' is analyzed as the maximum probability $\Pr[c = \mathsf{Enc}(pk, M; r)]$ for $M \leftarrow \psi_{\mathcal{M}}$ and $r \leftarrow \psi_{\mathcal{R}}$ of PKE. For each sampled message $M \in \mathcal{M}$ with probability $\psi_{\mathcal{M}}(M)$, we consider the γ-spreadness of PKE. To account for all possible messages $\{M\}$ from $\mathcal{M}$, we upper-bound the maximum probability by multiplying $2^{-\gamma}$ by the value $|\mathcal{M}| \cdot \max_{M \in \mathcal{M}} \psi_{\mathcal{M}}(M)$.

Theorem 3.5. If PKE is (weakly) γ-spread, SOTP has the message hiding property, and G is modeled as a random oracle, then $\mathsf{PKE}' = \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ is (weakly) γ'-spread with

$$\gamma' = \gamma - \log_2\left(|\mathcal{M}| \cdot \max_{M \in \mathcal{M}} \psi_{\mathcal{M}}(M)\right),$$

where $\mathcal{M}$ is the message space of PKE and $\psi_{\mathcal{M}}(M)$ is the probability that M is sampled from $\mathcal{M}$ according to the distribution $\psi_{\mathcal{M}}$.

Proof. For a fixed key pair (pk, sk) and a fixed m in PKE', we consider the probability $\Pr_{R\leftarrow\mathcal{R}',\mathsf{G}}[c = \mathsf{Enc}'(pk, m; R)]$ for every possible ciphertext c. Because G is modeled as a random oralce, the probability is taken over random choice of G. Given that r is sampled as $r \leftarrow \psi_{\mathcal{R}}$ using the randomness $R \leftarrow \mathcal{R}'$, the probability can be rewritten as

$$\Pr_{R\leftarrow\mathcal{R}',\mathsf{G}}[c = \mathsf{Enc}'(pk, m; R)] = \Pr_{r\leftarrow\psi_{\mathcal{R}},\mathsf{G}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r)); r)].$$

Using the law of total probability based on all possible values of $r_i \in \mathcal{R}$ into conditions:

$$\begin{aligned}
&\Pr_{r\leftarrow\psi_{\mathcal{R}},\mathsf{G}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r)); r)] \\
&= \sum_{r_i\in\mathcal{R}} \Pr_{\mathsf{G}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r)); r) \mid r = r_i] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \\
&= \sum_{r_i\in\mathcal{R}} \Pr_{\mathsf{G}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r_i)); r_i)] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i].
\end{aligned}$$

By the fact that $\mathsf{G}(r_i)$ is $\psi_{\mathcal{U}}$-distributed, the message hiding property of SOTP holds using $\mathsf{G}(r_i)$, so the output $M = \mathsf{Encode}(m, \mathsf{G}(r_i))$ is $\psi_{\mathcal{M}}$-distributed over random choice of G:

$$\begin{aligned}
&\sum_{r_i\in\mathcal{R}} \Pr_{\mathsf{G}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, \mathsf{G}(r_i)); r_i)] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \\
&= \sum_{r_i\in\mathcal{R}} \Pr_{u\leftarrow\psi_{\mathcal{U}}}[c = \mathsf{Enc}(pk, \mathsf{Encode}(m, u); r_i)] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \\
&= \sum_{r_i\in\mathcal{R}} \Pr_{M\leftarrow\psi_{\mathcal{M}}}[c = \mathsf{Enc}(pk, M; r_i)] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i].
\end{aligned}$$

For the ease of analysis, we define an indicator function $\mathbf{I}(pk, M, r, c) = [\![c == \mathsf{Enc}(pk, M; r)]\!]$. Then,

$$\begin{aligned}
&\sum_{r_i\in\mathcal{R}} \Pr_{M\leftarrow\psi_{\mathcal{M}}}[c = \mathsf{Enc}(pk, M; r_i)] \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \\
&= \sum_{r_i\in\mathcal{R}} \Big(\sum_{M_j\in\mathcal{M}} \mathbf{I}(pk, M_j, r_i, c) \cdot \Pr_{M\leftarrow\psi_{\mathcal{M}}}[M = M_j] \Big) \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \\
&= \sum_{M_j\in\mathcal{M}} \Big(\sum_{r_i\in\mathcal{R}} \mathbf{I}(pk, M_j, r_i, c) \cdot \Pr_{r\leftarrow\psi_{\mathcal{R}}}[r = r_i] \Big) \cdot \Pr_{M\leftarrow\psi_{\mathcal{M}}}[M = M_j] \\
&= \sum_{M_j\in\mathcal{M}} \Pr_{r\leftarrow\psi_{\mathcal{R}}}[c = \mathsf{Enc}(pk, M_j; r)] \cdot \Pr_{M\leftarrow\psi_{\mathcal{M}}}[M = M_j].
\end{aligned}$$

Considering $\Pr_{r\leftarrow\psi_{\mathcal{R}}}[c = \mathsf{Enc}(pk, M_j; r)]$ as the γ-spreadness of PKE on any message M_j, the γ'-spreadness of PKE' is upper-bounded as follows:

$$\begin{aligned}
\Pr_{R\leftarrow\mathcal{R}',\mathsf{G}}[c = \mathsf{Enc}'(pk, m; R)] &= \sum_{M_j\in\mathcal{M}} \Pr_{r\leftarrow\psi_{\mathcal{R}}}[c = \mathsf{Enc}(pk, M_j; r)] \cdot \Pr_{M\leftarrow\psi_{\mathcal{M}}}[M = M_j] \\
&\leq |\mathcal{M}| \cdot 2^{-\gamma} \cdot \max_{M\in\mathcal{M}} \psi_{\mathcal{M}}(M).
\end{aligned}$$

By averaging over (pk, sk), the weak γ'-spreadness of PKE' is also obtained. □

4 $\mathsf{FO}_{\mathsf{PKE}}$ Transformation and Its Security Proofs in the (Q)ROM

Figure 4 presents the FO transformation $\mathsf{FO}_{\mathsf{PKE}}$ [14] for PKE, described as $\mathsf{PKE}'' := \mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$. For some positive integers ℓ_m and ℓ_r, let $\mathcal{M}' = \{0,1\}^{\ell_m+\ell_r}$ and $\mathcal{M}'' = \{0,1\}^{\ell}_{m}$ be the message spaces of PKE' and PKE'', respectively. A message $m \in \{0,1\}^{\ell_m}$ of PKE'' is converted into a new message $\tilde{m} := m||r \in \{0,1\}^{\ell_m+\ell_r}$ of PKE' by concatenating a sufficiently large random bit-string $r \in \{0,1\}^{\ell_r}$ with m. During decryption of PKE'', m is recovered by taking $[\tilde{m}']_{\ell_m}$, which denotes the most significant ℓ_m bits of $\tilde{m}'$. PKE'' preserves the worst-case correctness error of PKE' since Dec'' works correctly as long as Dec' performs correctly.

Assuming that PKE' is IND-CPA secure and that the hash function H is modeled as a (quantum) random oracle, we prove that PKE'' is IND-CCA secure in the (Q)ROM. In the ROM, our proof is based on the previous work [14], but we more clearly make use of the worst-case δ-correctness and weak γ-spreadness of the underlying PKE'. In the QROM, in addition to these two information-theoretic properties, we use the adaptive O2H lemma [35] and the extractable random oracle (RO) simulator [10] to prove the IND-CCA security of PKE''.

$\underline{\mathsf{Gen}''(1^\lambda)}$
1: $(pk, sk) := \mathsf{Gen}'(1^\lambda)$
2: **return** (pk, sk)

$\underline{\mathsf{Enc}''(pk, m \in \{0,1\}^{\ell_m})}$
1: $r \leftarrow \{0,1\}^{\ell_r}$
2: $\tilde{m} = m||r \in \{0,1\}^{\ell_m+\ell_r}$
3: $R := \mathsf{H}(\tilde{m})$
4: $c := \mathsf{Enc}'(pk, \tilde{m}; R)$
5: **return** c

$\underline{\mathsf{Dec}''(sk, c)}$
1: $\tilde{m}' = \mathsf{Dec}'(sk, c)$
2: $R' := \mathsf{H}(\tilde{m}')$
3: **if** $\tilde{m}' = \perp$ **or** $c \neq \mathsf{Enc}'(pk, \tilde{m}'; R')$
4: **return** $\perp$
5: **else**
6: **return** $[\tilde{m}']_{\ell_m}$

Figure 4: $\mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}] = (\mathsf{Gen}'', \mathsf{Enc}'', \mathsf{Dec}'')$

4.1 Security Proof in the ROM

Theorem 4.1 (IND-CPA of $\mathsf{PKE}' \overset{\text{ROM}}{\Longrightarrow}$ IND-CCA of PKE'')**.** Let PKE' be a public-key encryption scheme with worst-case correctness error δ and weakly γ-spreadness. For any classical adversary $\mathcal{A}$ against the IND-CCA security of PKE'', making at most q_{D} queries to the decryption oracle Dec'' and at most q_{H} queries to $\mathsf{H} : \mathcal{M} \to \mathcal{R}$, there exists a classical adversary $\mathcal{B}$ against the IND-CPA security of PKE' such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}''}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot (2^{-\gamma} + \delta) + q_{\mathsf{H}} \cdot 2^{-\ell_r}.$$

Proof. For the security proof, we analyze hybrid games G_0 to G_5, defined in Figures 5 and 6, with a fixed key pair (pk, sk). To do this, we define $\delta_{sk} := \max_{m \in \mathcal{M}} \Pr_{r \leftarrow \psi_R}[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m; r)) \neq m]$ as the maximum probability of a decryption error and $\gamma_{sk} := -\log \max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr_{r \leftarrow \psi_R}[c = \mathsf{Enc}'(pk, m; r)]$ as the negative logarithm of the maximum probability of any ciphertext for the fixed key pair (pk, sk), ensuring $\mathbb{E}[\delta_{sk}] \leq \delta$ and $\mathbb{E}[2^{-\gamma_{sk}}] \leq 2^{-\gamma}$, with expectations taken over $(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$. A detailed explanation of the security proof is provided below.

GAMES G_0-G_2

$(pk, sk) \leftarrow \mathsf{Gen}''(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$b \leftarrow \{0,1\}$
$r \leftarrow \{0,1\}^{\ell_r}$
$\tilde{m} = m_b || r \in \{0,1\}^{n=\ell_m+\ell_r}$
$\tilde{r} = \mathsf{H}(\tilde{m})$
$c^* = \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$
$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return $[\![b = b']\!]$

GAME G_3

$(pk, sk) \leftarrow \mathsf{Gen}''(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$
$b \leftarrow \{0,1\}$
$\tilde{m}_b = m_b || r_b \in \{0,1\}^{n=\ell_m+\ell_r}$
$\tilde{r} = \mathsf{H}(\tilde{m}_b)$
$c^* := \mathsf{Enc}'(pk, \tilde{m}_b; \tilde{r})$
$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c)$
return $[\![b = b']\!]$

$\mathsf{H}(\tilde{m})$

if $\exists \tilde{r}$ such that $(\tilde{m}, \tilde{r}) \in \mathcal{L}_\mathsf{H}$
  return $\tilde{r}$
$\tilde{r} \leftarrow \mathcal{R}$
$\mathcal{L}_\mathsf{H} := \mathcal{L}_\mathsf{H} \cup \{(\tilde{m}, \tilde{r})\}$
return $\tilde{r}$

$\mathsf{Dec}''(c \neq c^*)$ //G_0

$\tilde{m}' = \mathsf{Dec}'(sk, c)$
if $\tilde{m}' = \perp$ or
  $c \neq \mathsf{Enc}'(pk, \tilde{m}'; \mathsf{H}(\tilde{m}'))$
  return $\perp$
else, return $[\tilde{m}']_{\ell_m}$

$\mathsf{Dec}''(c \neq c^*)$ //G_1-G_3

$\tilde{m}' = \mathsf{Dec}'(sk, c)$
if $\exists (\tilde{m}, \tilde{r}) \in \mathcal{L}_\mathsf{H}$ such that
  $c = \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$ //G_1-G_3
  and $\tilde{m} = \tilde{m}'$ //G_1
  return $[\tilde{m}]_{\ell_m}$
else, return $\perp$

Figure 5: GAMES G_0-G_3 for the proof of Theorem 4.1

GAME G_0. G_0 is the IND-CCA game against PKE'' with a fixed key pair (pk, sk) (see Figure 5). Here, we define the advantage of an adversary $\mathcal{A}$ in the IND-CCA game against PKE'' for a fixed key pair (pk, sk) as:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}'',sk}(\mathcal{A}) = \left| \Pr[G_0^\mathcal{A} \Rightarrow 1] - \frac{1}{2} \right|.$$

GAME G_1. G_1 is defined by modifying the Dec'' oracle, as shown in Figure 5. In G_1, the Dec'' oracle is altered to first compute $\tilde{m}' = \mathsf{Dec}'(sk, c)$ and return $[\tilde{m}']_{\ell_m}$ if there exists $(\tilde{m}, \tilde{r}) \in \mathcal{L}_\mathsf{H}$ such that $\mathsf{Enc}'(pk, \tilde{m}; \tilde{r}) = c$ and $\tilde{m} = \tilde{m}'$. The Dec'' oracle in G_0 differs from that in G_1 if $\mathsf{H}(\tilde{m})$ has not been queried, which occurs with probability $\cdot 2^{-\gamma_{sk}}$. By the union bound:

$$|\Pr[G_0^\mathcal{A} \Rightarrow 1] - \Pr[G_1^\mathcal{A} \Rightarrow 1]| \leq (q_\mathsf{H} + q_\mathsf{D}) \cdot 2^{-\gamma_{sk}}.$$

GAME G_2. G_2 is defined by modifying the Dec'' oracle, as shown in Figure 5. In G_2, Dec'' no longer checks whether $\tilde{m} = \tilde{m}'$, where $\tilde{m}' = \mathsf{Dec}'(sk, c)$. Instead, it returns $\tilde{m}$ directly if there exists $(\tilde{m}, \tilde{r}) \in \mathcal{L}_\mathsf{H}$ such that $\mathsf{Enc}'(pk, \tilde{m}; \tilde{r}) = c$. Since the Dec'' oracle in G_1 is identical to that of G_2 if there are no hash queries to H that lead to a correctness error, by the union bound, the following holds:

$$|\Pr[G_1^\mathcal{A} \Rightarrow 1] - \Pr[G_2^\mathcal{A} \Rightarrow 1]| \leq (q_\mathsf{H} + q_\mathsf{D}) \cdot \delta_{sk}.$$

Note that the Dec'' oracle in G_2 no longer requires the secret key.

GAMES G_4-G_5

$(pk, sk) \leftarrow \mathsf{Gen}''(1^\lambda)$
$(\tilde{m}_0, \tilde{m}_1) \leftarrow \mathcal{C}_0^{\mathsf{H}}(pk)$
$b \leftarrow \{0,1\}$
$\tilde{r}^* = \mathsf{H}(\tilde{m}_b)$ //G_4
$\tilde{r}^* \leftarrow \mathcal{R}$ //G_5
$c^* := \mathsf{Enc}'(pk, \tilde{m}_b; \tilde{r}^*)$
$b' \leftarrow \mathcal{C}_1^{\mathsf{H}}(pk, c^*)$
return $[\![b = b']\!]$

$\mathsf{H}(\tilde{m})$

if $\exists \tilde{r}$ such that $(\tilde{m}, \tilde{r}) \in \mathcal{L}_{\mathsf{H}}$
  return $\tilde{r}$
else, $\tilde{r} \leftarrow \mathcal{R}$
$\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(\tilde{m}, \tilde{r})\}$
return $\tilde{r}$

$\mathsf{Dec}''(c \neq c^*)$

if $\exists(\tilde{m}, \tilde{r}) \in \mathcal{L}_{\mathsf{H}}$ such that $c = \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$
  return $[\tilde{m}]_{\ell_m}$
else, return $\perp$

$\mathcal{C}_0^{\mathsf{H}}(pk)$

$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$
return $(\tilde{m}_0, \tilde{m}_1) = (m_0||r_0, m_1||r_1)$

$\mathcal{C}_1^{\mathsf{H}}(pk)$

$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return b'

Figure 6: GAMES G_4-G_5 of Theorem 4.1

GAME G_3. G_3 is defined by replacing $\tilde{m}$ by $\tilde{m}_b$, as shown in Figure 5. Since this change is only conceptual, the following holds:

$$\Pr[G_2^{\mathcal{A}} \Rightarrow 1] = \Pr[G_3^{\mathcal{A}} \Rightarrow 1].$$

GAME G_4. G_4 is defined by moving part of the game into an adversary $\mathcal{C}^{\mathsf{H}} = (\mathcal{C}_0^{\mathsf{H}}, \mathcal{C}_1^{\mathsf{H}})$, defined in Figure 6. Since the change is only conceptual, the following holds:

$$\Pr[G_3^{\mathcal{A}} \Rightarrow 1] = \Pr[G_4^{\mathcal{A}} \Rightarrow 1].$$

GAME G_5. G_5 is defined by changing how $\tilde{r}^*$ is chosen. In G_5, instead of generating $\tilde{r}^*$ using H, $\tilde{r}^*$ is chosen randomly from $\mathcal{R}$, which will not be noticed by $\mathcal{A}$ as long as $\mathcal{A}$ does not query $\tilde{r}$ to H. Let QUERY be an event that $\mathcal{A}$ queries H on $\tilde{m}_b$. Due to the difference lemma [32], the following holds:

$$|\Pr[G_4^{\mathcal{A}} \Rightarrow 1] - \Pr[G_5^{\mathcal{A}} \Rightarrow 1]| \leq \Pr[\mathsf{QUERY}].$$

Also, since the adversary $\mathcal{C}$ in G_5 is playing the original $\mathsf{IND\text{-}CPA}$ game against PKE', the following holds

$$|\Pr[G_5^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}',sk}(\mathcal{C}).$$

Now, construct an adversary $\mathcal{D}^{\mathcal{H}} = (\mathcal{D}_0^{\mathcal{H}}, \mathcal{D}_1^{\mathcal{H}})$ in Figure 7 that solves the $\mathsf{IND\text{-}CPA}$ game with PKE' when the event QUERY occurs. Since r_{1-b} is completely hidden from the adversary $\mathcal{A}$, the probability that $\mathcal{A}$ ever queries $\tilde{m}_{1-b} = (m_{1-b}||r_{1-b})$ to H can be bounded to $q_{\mathsf{H}} \cdot 2^{-\ell_r}$. Therefore, the following holds:

$$\Pr[\mathsf{QUERY}] \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}',sk}(\mathcal{D}) + q_{\mathsf{H}} \cdot 2^{-l_r}.$$

Combining the intermediate results and folding $\mathcal{C}$ and $\mathcal{D}$ into one single adversary $\mathcal{B}$ against $\mathsf{IND\text{-}CPA}$ with PKE', and then taking the expectation over $(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$ yields the required bound of the theorem. □

$\underline{\mathcal{D}_0^{\mathsf{H}}(pk)}$

1: $\mathcal{L}_{\mathsf{H}}, \mathcal{L}_{\tilde{m}} := \emptyset$
2: $(\tilde{m}_0, \tilde{m}_1) \leftarrow \mathcal{C}_0^{\mathsf{H}}(pk)$
3: **return** $(\tilde{m}_0, \tilde{m}_1)$

$\underline{\mathcal{D}_1^{\mathsf{H}}(pk, c^*)}$

1: $\mathcal{C}_1^{\mathsf{H}}(pk, c^*)$
2: **if** $\tilde{m}_0 \in \mathcal{L}_{\tilde{m}}$**, return** $b' = 0$
3: **else, return** $b' = 1$

$\underline{\mathsf{H}(\tilde{m})}$

1: **if** $\exists \tilde{r}$ such that $(\tilde{m}, \tilde{r}) \in \mathcal{L}_{\mathsf{H}}$
2: **return** $\tilde{r}$
3: $\tilde{r} \leftarrow \mathcal{R}$
4: $\mathcal{L}_{\mathsf{H}} := \mathcal{L}_{\mathsf{H}} \cup \{(\tilde{m}, \tilde{r})\}$
5: $\mathcal{L}_{\tilde{m}} := \mathcal{L}_{\tilde{m}} \cup \{\tilde{m}\}$
6: **return** $\tilde{r}$

Figure 7: The adversary $\mathcal{D}$ in Theorem 4.1

4.2 Security Proof in the QROM

Theorem 4.2 (IND-CPA of PKE' $\overset{\text{QROM}}{\Longrightarrow}$ IND-CCA of PKE'')**.** Let PKE' be a public-key encryption scheme with a worst-case correctness error δ that satisfies weak γ-spreadness. For any quantum adversary $\mathcal{A}$ against the IND-CCA security of PKE'', making at most q_{D} queries to the decryption oracle Dec'' and at most q_{H} queries to $\mathsf{H} : \mathcal{M} \to \mathcal{R}$, there exist a quantum adversary $\mathcal{B}$ against the IND-CPA security of PKE' such that

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}''}(\mathcal{A}) \leq (2q_{\mathsf{H}} + 2q_{\mathsf{D}} + 1)\sqrt{2\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}'}(\mathcal{B}) + \varepsilon} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2}$$

where $\varepsilon = 128(q_{\mathsf{H}} + q_{\mathsf{D}})^2\delta + q_{\mathsf{D}} \cdot (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{(-\gamma+9)/2} + q_{\mathsf{D}} \cdot 2^{-\ell_r+1}$.

The proof strategy for Theorem 4.2 closely follows Theorem 6.1 in [10], with a key distinction in the application of the O2H lemma. While [10] used Lemma 2.8 (Theorem 3 of [3]) to prove the IND-CCA security of the KEM, an adaptive version of the O2H lemma, as outlined in Lemma 2.7, is used to prove the IND-CCA security of PKE''.

Proof. The security proof begins by analyzing hybrid games with a fixed key pair (pk, sk). To do this, we define $\delta_{sk} := \max_{m \in \mathcal{M}} \Pr_{r \leftarrow \psi_R}[\mathsf{Dec}'(sk, \mathsf{Enc}'(pk, m; r)) \neq m]$ as the maximum probability of a decryption error and $\gamma_{sk} := -\log \max_{m \in \mathcal{M}, c \in \mathcal{C}} \Pr_{r \leftarrow \psi_R}[c = \mathsf{Enc}'(pk, m; r)]$ as the negative logarithm of the maximum probability of any ciphertext for the fixed key pair (pk, sk), ensuring $\mathbb{E}[\delta_{sk}] \leq \delta$ and $\mathbb{E}[2^{-\gamma_{sk}}] \leq 2^{-\gamma}$, with expectations taken over $(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$. A detailed explanation of the security proof is provided below.

GAME G_0. G_0 is the original IND-CCA game against PKE'' with the fixed key pair (pk, sk). Here, define the advantage of adversary $\mathcal{A}$ in the IND-CCA game against PKE'' for a fixed key pair (pk, sk) as:

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}'',sk}(\mathcal{A}) = \left|\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right|.$$

GAME G_1. G_1 is defined by moving parts of the game into a set of algorithms $\mathcal{C}^{\mathsf{H}} = (\mathcal{C}_0^{\mathsf{H}}, \mathcal{C}_1^{\mathsf{H}})$, as shown in Figure 8. Since this change is only conceptual, it holds that:

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[G_1^{\mathcal{A}} \Rightarrow 1].$$

GAME G_0

$\mathsf{H} \leftarrow (\mathcal{M} \to \mathcal{R})$
$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$b \leftarrow \{0,1\}$
$r \leftarrow \{0,1\}^{\ell_r}$
$\tilde{m} = m_b || r \in \{0,1\}^{n=\ell_m+\ell_r}$
$\tilde{r} = \mathsf{H}(\tilde{m})$
$c^* = \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$
$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return $[\![b = b']\!]$

GAMES G_1-G_3

$\mathsf{H} \leftarrow (\mathcal{M} \to \mathcal{R})$
$m_b \leftarrow \mathcal{C}_0^{\mathsf{H}}()$
$r \leftarrow \{0,1\}^{\ell_r}$
$\tilde{m} = m_b || r$
$\tilde{r} := \mathsf{H}(\tilde{m})$ //G_1
$\tilde{r} \leftarrow \mathcal{R}$ //G_2-G_3
$b' \leftarrow \mathcal{C}_1^{\mathsf{H}}(r, \tilde{r})$ //G_1-G_2
$\tilde{m}' \leftarrow \mathcal{D}^{\mathsf{H}}(r, \tilde{r})$ //G_3
return $[\![b = b']\!]$ //G_1-G_2
return $[\![\tilde{m}_b = \tilde{m}']\!]$ //G_3

$\mathsf{Dec}''(c \neq c^*)$

$\tilde{m}' = \mathsf{Dec}'(sk, c)$
$\tilde{r}' = \mathsf{H}(\tilde{m}')$
if $c \neq \mathsf{Enc}'(pk, \tilde{m}'; \tilde{r}')$
  return $\perp$
else, return $[\![\tilde{m}']\!]_{\ell_m}$

$\mathcal{C}_0^{\mathsf{H}}()$

$(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$b \leftarrow \{0,1\}$
return m_b

$\mathcal{C}_1^{\mathsf{H}}(r, \tilde{r})$

$c^* \leftarrow \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$
$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return b'

$\mathcal{D}^{\mathsf{H}}(r, \tilde{r})$

$i \leftarrow \{1, \cdots, q_{\mathsf{H}}\}$
Run $\mathcal{C}_1^{\mathsf{H}}(r, \tilde{r})$ till i-th H-query
$\tilde{m}' \leftarrow$ measure i-th H-query
return $\tilde{m}'$

Figure 8: GAMES G_0-G_3 for the proof of Theorem 4.2

GAMES G_2 AND G_3. G_2 and G_3 are defined by applying Lemma 2.7 to G_1 and $\mathcal{C}^{\mathsf{H}}$ (see Figure 8). Note that G_2 and G_3 generate $\tilde{r} \leftarrow \mathcal{R}$ instead of $\tilde{r} = \mathsf{H}(\tilde{m})$. As a result, it holds that:

$$|\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]| \leq 2 \cdot (q_{\mathsf{H}} + q_{\mathsf{D}})\sqrt{\Pr[G_3 \Rightarrow 1]} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2}.$$

Combining the analyses of G_0 to G_3, the following inequality holds:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}_{\mathsf{PKE}',sk}(\mathcal{A}) &= |\Pr[G_0^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| = |\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| \\
&\leq |\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_2^{\mathcal{A}} \Rightarrow 1]| + |\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| \\
&\leq 2 \cdot (q_{\mathsf{H}} + q_{\mathsf{D}})\sqrt{\Pr[G_3 \Rightarrow 1]} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2} + |\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}|. \quad (1)
\end{aligned}$$

GAME $G_{2.1}$. $G_{2.1}$ is defined by modifying G_2, moving parts of the set of algorithms $\mathcal{C}^{\mathsf{H}} = (\mathcal{C}_0^{\mathsf{H}}, \mathcal{C}_1^{\mathsf{H}})$ into the game, as shown in Figure 9. Since this change is only conceptual, it holds that:

$$\Pr[G_2^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{2.1}^{\mathcal{A}} \Rightarrow 1].$$

GAMES $G_{2.1}$-$G_{2.2}$

$\mathsf{H} \leftarrow (\mathcal{M} \to \mathcal{R})$
$(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$ //$G_{2.2}$
$b \leftarrow \{0,1\}$
$r \leftarrow \{0,1\}^{\ell_r}$ //$G_{2.1}$
$\tilde{m} = m_b || r$ //$G_{2.1}$
$\tilde{m} = m_b || r_b$ //$G_{2.2}$
$\tilde{r} \leftarrow \mathcal{R}$
$c^* \leftarrow \mathsf{Enc}'(pk, \tilde{m}; \tilde{r})$
$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return $[\![b = b']\!]$

GAMES $G_{2.3}$-$G_{2.7}$

$\mathsf{H} \leftarrow (\mathcal{M} \to \mathcal{R})$ //$G_{2.3}$
$\mathsf{H} = \mathcal{S}.RO$ //$G_{2.4}$-$G_{2.7}$
$(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$
$(\tilde{m}_0, \tilde{m}_1) \leftarrow \mathcal{E}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$b \leftarrow \{0,1\}$
$\tilde{r} \leftarrow \mathcal{R}$
$c^* \leftarrow \mathsf{Enc}'(pk, \tilde{m}_b; \tilde{r})$
$b' \leftarrow \mathcal{E}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return $[\![b = b']\!]$
while $i \in I$ **do** //$G_{2.4}$
  $\hat{m}_i \leftarrow \mathcal{S}.E(c_i)$ //$G_{2.4}$

$\mathsf{Dec}''(c \neq c^*)$

$\tilde{m}' = \mathsf{Dec}'(sk, c)$ //$G_{2.1}$-$G_{2.6}$
$\tilde{r}' = \mathsf{H}(\tilde{m}')$ //$G_{2.1}$-$G_{2.6}$
if $c \neq \mathsf{Enc}'(pk, \tilde{m}'; \tilde{r}')$ //$G_{2.1}$-$G_{2.5}$
  return $\perp$ //$G_{2.1}$-$G_{2.5}$
else, return $[\![\tilde{m}']\!]_{\ell_m}$ //$G_{2.1}$-$G_{2.5}$
$\hat{m}' \leftarrow \mathcal{S}.E(c)$ //$G_{2.5}$-$G_{2.7}$
if $\hat{m}' = \perp$, **return** $\perp$ //$G_{2.6}$-$G_{2.7}$
else, return $[\![\hat{m}']\!]_{\ell_m}$ //$G_{2.6}$-$G_{2.7}$

$\mathcal{E}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$

$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$
return $(\tilde{m}_0, \tilde{m}_1) = (m_0 || r_0, m_1 || r_1)$

$\mathcal{E}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$

$b' \leftarrow \mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$
return b'

Figure 9: GAMES $G_{2.1}$-$G_{2.7}$ for the proof of Theorem 4.2

GAME $G_{2.2}$. $G_{2.2}$ is defined by modifying the generation of $\tilde{m}$, as shown in Figure 9. Since this change is only conceptual, the following holds:

$$\Pr[G_{2.1}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{2.2}^{\mathcal{A}} \Rightarrow 1].$$

GAME $G_{2.3}$. $G_{2.3}$ is defined by moving parts of the game into a set of algorithms $\mathcal{E}^{\mathsf{H},\mathsf{Dec}''} = (\mathcal{E}_0^{\mathsf{H},\mathsf{Dec}''}, \mathcal{E}_1^{\mathsf{H},\mathsf{Dec}''})$, as shown in Figure 9. Since this change is conceptual, it holds that:

$$\Pr[G_{2.2}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{2.3}^{\mathcal{A}} \Rightarrow 1].$$

GAME $G_{2.4}$. $G_{2.4}$ is defined by replacing the random oracle H with the extractable RO-simulator $\mathcal{S}$ for the relation $R_t := \{(x, y) \mid f(x, y) = t\}$, where $f(x, y) = \mathsf{Enc}'(pk, x; y)$ from Theorem 2.10, as shown in Figure 9. Furthermore, at the end of the game, the extractor interface $\mathcal{S}.E$ is invoked to compute $\hat{m}_i := \mathcal{S}.E(c_i)$ for each c_i that $\mathcal{A}$ queried to Dec'' during its run. According to the first statement of Theorem 2.10,

$$\Pr[G_{2.3}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{2.4}^{\mathcal{A}} \Rightarrow 1].$$

Furthermore, applying Theorem 2.11 for $R' := \{(m, c) : \mathsf{Dec}'(sk, c) \neq m\}$, the event

$$P^{\dagger} := [\forall i : \hat{m}_i = \tilde{m}'_i := \mathsf{Dec}'(sk, c_i) \vee \hat{m}_i = \emptyset]$$

holds except with probability $\varepsilon_{1,sk} := 128(q_{\mathsf{H}} + q_{\mathsf{D}})^2 \Gamma_R / |\mathcal{R}| = 128(q_{\mathsf{H}} + q_{\mathsf{D}})^2 \delta_{sk}$. Thus,

$$|\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P^{\dagger}]| \leq \varepsilon^1_{sk}.$$

GAME $G_{2.5}$. $G_{2.5}$ is defined by moving each query $\mathcal{S}.E(c_i)$ to the end of the $\mathsf{Dec}''(c_i)$ oracle. Since $\mathcal{S}.RO(m)$ and $\mathcal{S}.E(c_i)$ now form consecutive classical queries, it follows from the contraposition of 4.(b) of Theorem 2.10 that, except with probability $2 \cdot 2^{-\ell_r}$, $\hat{m}_i = \emptyset$ implies $\mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i$. Applying the union bound, $P^{\dagger}$ implies

$$P := [\forall i : \hat{m}_i = m_i \vee (\hat{m}_i = \emptyset \wedge \mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i)]$$

except with probability $q_{\mathsf{D}} \cdot 2 \cdot 2^{-\ell_r}$. Furthermore, by 2.(c) of Theorem 2.10, each swap of a $\mathcal{S}.RO$ with a $\mathcal{S}.E$ query affects the final probability by at most $8\sqrt{2\Gamma(f)/|\mathcal{R}|} = 8\sqrt{2 \cdot 2^{-\gamma_{sk}}}$. Thus,

$$|\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P^{\dagger}] - \Pr[G^{\mathcal{A}}_{2.5} \Rightarrow 1 \wedge P]| \leq \varepsilon_{2,sk}$$

with $\varepsilon_{2,sk} = 2q_{\mathsf{D}} \cdot ((q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 4\sqrt{2 \cdot 2^{-\gamma_{sk}}} + 2^{-\ell_r})$.
GAME $G_{2.6}$. In $G_{2.6}$, the decryption oracle Dec'' uses $\hat{m}'_i$ instead of $\tilde{m}'_i$ to response to the queries. However, Dec'' still queries $\mathcal{S}.RO(\tilde{m}'_i)$, maintaining the interaction pattern between Dec'' and $\mathcal{S}.RO$ as in $G_{2.5}$.

Note that if the event

$$P_i := [\hat{m}'_i = m_i \vee (\hat{m}_i = \emptyset \wedge \mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i)]$$

holds for a given i, then the above change will not affect the response of Dec'' and thus will not affect the probability for P_{i+1} to hold as well. Therefore, by mathematical induction, the following holds:

$$\Pr[G^{\mathcal{A}}_{2.5} \Rightarrow 1 \wedge P] = \Pr[G^{\mathcal{A}}_{2.6} \Rightarrow 1 \wedge P].$$

GAME $G_{2.7}$. In $G_{2.7}$, all $\tilde{r}' = \mathsf{H}(\tilde{m}')$ queries in Dec'' are dropped or, equivalently, moved to the very end of the game execution. Invoking 2.(c) of Theorem 2.10 once again, the following holds:

$$|\Pr[G^{\mathcal{A}}_{2.6} \Rightarrow 1 \wedge P] - \Pr[G^{\mathcal{A}}_{2.7} \Rightarrow 1 \wedge P]| \leq \varepsilon_{3,sk}.$$

with $\varepsilon_{3,sk} = q_{\mathsf{D}} \cdot (q_{\mathsf{D}} + q_{\mathsf{H}}) \cdot 8\sqrt{2 \cdot 2^{-\gamma_{sk}}}$. Also, note that $G_{2.7}$ works without knowledge of the secret key sk and thus constitutes a IND-CPA attacker $\mathcal{E}$ against PKE for a fixed key pair (pk, sk). Therefore,

$$|\Pr[G^{\mathcal{A}}_{2.7} \Rightarrow 1 \wedge P] - \frac{1}{2}| \leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{E}),$$

where $\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{E})$ is the advantage of the adversary $\mathcal{E}$ in the IND-CPA game against PKE for a fixed key pair (pk, sk). Combining the analyses from G_2 to $G_{2.7}$ so far, the following holds:

$$\begin{aligned}
|\Pr[G^{\mathcal{A}}_2 \Rightarrow 1] - \frac{1}{2}| &= |\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1] - \frac{1}{2}| \\
&\leq |\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P^{\dagger}]| + |\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P] - \frac{1}{2}|
\end{aligned}$$

$$
\begin{aligned}
&\leq |\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P^{\dagger}] - \frac{1}{2}| + \varepsilon_{1,sk} \\
&\leq |\Pr[G^{\mathcal{A}}_{2.4} \Rightarrow 1 \wedge P^{\dagger}] - \Pr[G^{\mathcal{A}}_{2.5} \Rightarrow 1 \wedge P]| + |\Pr[G^{\mathcal{A}}_{2.5} \Rightarrow 1 \wedge P] - \frac{1}{2}| + \varepsilon_{1,sk} \\
&\leq |\Pr[G^{\mathcal{A}}_{2.5} \Rightarrow 1 \wedge P] - \frac{1}{2}| + \varepsilon_{1,sk} + \varepsilon_{2,sk} \\
&= |\Pr[G^{\mathcal{A}}_{2.6} \Rightarrow 1 \wedge P] - \frac{1}{2}| + \varepsilon_{1,sk} + \varepsilon_{2,sk} \\
&\leq |\Pr[G^{\mathcal{A}}_{2.6} \Rightarrow 1 \wedge P] - \Pr[G^{\mathcal{A}}_{2.7} \Rightarrow 1 \wedge P]| + |\Pr[G^{\mathcal{A}}_{2.7} \Rightarrow 1 \wedge P] - \frac{1}{2}| + \varepsilon_{1,sk} + \varepsilon_{2,sk} \\
&\leq |\Pr[G^{\mathcal{A}}_{2.7} \Rightarrow 1 \wedge P] - \frac{1}{2}| + \varepsilon_{1,sk} + \varepsilon_{2,sk} + \varepsilon_{3,sk} \\
&\leq \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{E}) + \varepsilon_{sk}, \qquad (2)
\end{aligned}
$$

where $\varepsilon_{sk} = \varepsilon_{1,sk} + \varepsilon_{2,sk} + \varepsilon_{3,sk}$.

GAME $G_{3.1}$. $G_{3.1}$ is defined by modifying G_3, moving parts of the set of algorithms $\mathcal{C}^{\mathsf{H}} = (\mathcal{C}^{\mathsf{H}}_0, \mathcal{C}^{\mathsf{H}}_1)$ to the game and the algorithm $\mathcal{F}^{\mathsf{H},\mathsf{Dec}''}_1$, as shown in Figure 10. Since this change is only conceptual, the following holds:

$$\Pr[G^{\mathcal{A}}_3 \Rightarrow 1] = \Pr[G^{\mathcal{A}}_{3.1} \Rightarrow 1].$$

GAME $G_{3.2}$. $G_{3.2}$ is defined by modifying the generation of $\tilde{m}_b$, as shown in Figure 10. Since this change is only conceptual, the following holds:

$$\Pr[G^{\mathcal{A}}_{3.1} \Rightarrow 1] = \Pr[G^{\mathcal{A}}_{3.2} \Rightarrow 1].$$

GAME $G_{3.3}$. $G_{3.3}$ is defined by moving parts of the game into the algorithm $\mathcal{F}^{\mathsf{H},\mathsf{Dec}''}_0$, as defined in Figure 10. Since this change is only conceptual, the following holds:

$$\Pr[G^{\mathcal{A}}_{3.2} \Rightarrow 1] = \Pr[G^{\mathcal{A}}_{3.3} \Rightarrow 1].$$

GAME $G_{3.4}$. $G_{3.4}$ is defined by replacing the random oracle H with the extractable RO-simulator $\mathcal{S}$ for the relation $R_t := \{(x, y) \mid f(x, y) = t\}$, where $f(x, y) = \mathsf{Enc}'(pk, x; y)$ from Theorem 2.10, as shown in Figure 10. Furthermore, at the end of the game, the extractor interface $\mathcal{S}.E$ is invoked to compute $\hat{m}_i := \mathcal{S}.E(c_i)$ for each c_i that $\mathcal{A}$ queried to Dec'' during its run. According to the first statement of Theorem 2.10,

$$\Pr[G^{\mathcal{A}}_{3.3} \Rightarrow 1] = \Pr[G^{\mathcal{A}}_{3.4} \Rightarrow 1].$$

Furthermore, applying Theorem 2.11 for $R' := \{(m, c) : \mathsf{Dec}'(sk, c) \neq m\}$, the event

$$P^{\dagger} := [\forall i : \hat{m}_i = \tilde{m}'_i := \mathsf{Dec}'(sk, c_i) \vee \hat{m}_i = \emptyset]$$

holds except with probability $\varepsilon_{1,sk} := 128(q_{\mathsf{H}} + q_{\mathsf{D}})^2 \delta_{sk}$. Thus,

$$|\Pr[G^{\mathcal{A}}_{3.4} \Rightarrow 1] - \Pr[G^{\mathcal{A}}_{3.4} \Rightarrow 1 \wedge P^{\dagger}]| \leq \varepsilon_{1,sk}.$$

GAME $G_{3.5}$. $G_{3.5}$ is defined by moving each query $\mathcal{S}.E(c_i)$ to the end of the $\mathsf{Dec}''(c_i)$ oracle. Since $\mathcal{S}.RO(m)$ and $\mathcal{S}.E(c_i)$ now form consecutive classical queries, it follows from the contraposition of 4.(b)

GAMES $G_{3.1}$-$G_{3.8}$

$\mathsf{H} \leftarrow (\mathcal{M} \to \mathcal{R})$ //$G_{3.1}$-$G_{3.3}$
$\mathsf{H} = \mathcal{S}.RO$ //$G_{3.4}$-$G_{3.8}$
$(pk, sk) \leftarrow \mathsf{Gen}'(1^\lambda)$
$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$ //$G_{3.1}$-$G_{3.2}$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$ //$G_{3.2}$
$(\tilde{m}_0, \tilde{m}_1) \leftarrow \mathcal{F}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$ //$G_{3.3}$-$G_{3.8}$
$b \leftarrow \{0,1\}$
$r_b \leftarrow \{0,1\}^{\ell_r}$ //$G_{3.1}$
$\tilde{m}_b = m_b || r_b$ //$G_{3.1}$-$G_{3.2}$
$\tilde{r} \leftarrow \mathcal{R}$
$c^* \leftarrow \mathsf{Enc}'(pk, \tilde{m}_b; \tilde{r})$
$\tilde{m}' \leftarrow \mathcal{F}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$ //$G_{3.1}$-$G_{3.7}$
$b' \leftarrow \mathcal{G}_1^{\mathsf{H}}(pk, c^*)$ //$G_{3.8}$
return $[\![\tilde{m}_b = \tilde{m}']\!]$ //$G_{3.1}$-$G_{3.7}$
return $[\![b = b']\!]$ //$G_{3.8}$
while $i \in I$ **do** //$G_{3.4}$
  $\hat{m}_i \leftarrow \mathcal{S}.E(c_i)$ //$G_{3.4}$

$\mathsf{Dec}''(c \neq c^*)$

$\tilde{m}' = \mathsf{Dec}'(sk, c)$ //$G_{3.1}$-$G_{3.6}$
$\tilde{r}' = \mathsf{H}(\tilde{m}')$ //$G_{3.1}$-$G_{3.6}$
if $c \neq \mathsf{Enc}'(pk, \tilde{m}'; \tilde{r}')$ //$G_{3.1}$-$G_{3.5}$
  return $\perp$ //$G_{3.1}$-$G_{3.5}$
else, return $[\![\tilde{m}']\!]_{\ell_m}$ //$G_{3.1}$-$G_{3.5}$
$\hat{m}' \leftarrow \mathcal{S}.E(c)$ //$G_{3.5}$-$G_{3.8}$
if $\hat{m}' = \perp$, **return** $\perp$ //$G_{3.6}$-$G_{3.8}$
else, return $[\![\hat{m}']\!]_{\ell_m}$ //$G_{3.6}$-$G_{3.8}$

$\mathcal{F}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$

$(m_0, m_1) \leftarrow \mathcal{A}_0^{\mathsf{H},\mathsf{Dec}''}(pk)$
$(r_0, r_1) \leftarrow \{0,1\}^{\ell_r} \times \{0,1\}^{\ell_r}$
return $(m_0 || r_0, m_1 || r_1)$

$\mathcal{F}_1^{\mathsf{H},\mathsf{Dec}''}(pk, c^*)$

$i \leftarrow \{1, \cdots, q_{\mathsf{H}}\}$
Run $\mathcal{A}_1^{\mathsf{H},\mathsf{Dec}''}(r, \tilde{r})$ till i-th H-query
$\tilde{m}' \leftarrow$ measure i-th H-query
return $\tilde{m}'$

$\mathcal{G}_1^{\mathsf{H}}(pk, c^*)$

$\tilde{m}' \leftarrow \mathcal{F}_1^{\mathsf{H}}(pk, c^*)$
if $\tilde{m}_0 = \tilde{m}'$, **return** 0
else if $\tilde{m}_1 = \tilde{m}'$, **return** 1
else, return $b' \leftarrow \{0,1\}$

Figure 10: GAMES $G_{3.1}$-$G_{3.8}$ for the proof of Theorem 4.2

of Theorem 2.10 that, except with probability $2 \cdot 2^{-\ell_r}$, $\hat{m}_i = \emptyset$ implies $\mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i$. Applying the union bound, $P^\dagger$ implies

$$P := [\forall i : \hat{m}_i = m_i \vee (\hat{m}_i = \emptyset \wedge \mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i)]$$

except with probability $q_{\mathsf{D}} \cdot 2 \cdot 2^{-\ell_r}$. Furthermore, by 2.(c) of Theorem 2.10, each swap of $\mathcal{S}.RO$ with $\mathcal{S}.E$ affects the final probability by at most $8\sqrt{2\Gamma(f)/|\mathcal{R}|} = 8\sqrt{2 \cdot 2^{-\gamma_{sk}}}$. Thus,

$$|\Pr[G_{3.4}^{\mathcal{A}} \Rightarrow 1 \wedge P^\dagger] - \Pr[G_{3.5}^{\mathcal{A}} \Rightarrow 1 \wedge P]| \leq \varepsilon_{2,sk}$$

with $\varepsilon_{2,sk} = 2q_{\mathsf{D}} \cdot ((q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 4\sqrt{2 \cdot 2^{-\gamma_{sk}}} + 2^{-\ell_r})$.

GAME $G_{3.6}$. In $G_{3.6}$, the Dec'' oracle uses $\hat{m}_i'$ instead of $\tilde{m}_i'$ to respond to the queries, but still queries $\mathcal{S}.RO(\tilde{m}_i')$, maintaining the interaction pattern from $G_{3.5}$.

Note that if the event

$$P_i := [\hat{m}_i' = m_i \vee (\hat{m}_i = \emptyset \wedge \mathsf{Enc}'(pk, m_i; \mathcal{S}.RO(m_i)) \neq c_i)]$$

holds for a given i, then the above change will not affect the response of Dec'' and thus will not affect the probability for P_{i+1} to hold as well. Thus, by mathematical induction,

$$\Pr[G_{3.5}^{\mathcal{A}} \Rightarrow 1 \wedge P] = \Pr[G_{3.6}^{\mathcal{A}} \Rightarrow 1 \wedge P].$$

GAME $G_{3.7}$. In $G_{3.7}$, all $\tilde{r}' = \mathsf{H}(\tilde{m}')$ queries in Dec'' are dropped or, equivalently, moved to the very end of the game execution. Invoking 2.(c) of Theorem 2.10, it holds that:

$$|\Pr[G_{3.6}^{\mathcal{A}} \Rightarrow 1 \wedge P] - \Pr[G_{3.7}^{\mathcal{A}} \Rightarrow 1 \wedge P]| \leq \varepsilon_{3,sk},$$

where $\varepsilon_{3,sk} = q_{\mathsf{D}} \cdot (q_{\mathsf{D}} + q_{\mathsf{H}}) \cdot 8\sqrt{2 \cdot 2^{-\gamma_{sk}}}$. Note that $G_{3.7}$ works without the secret key sk.
GAME $G_{3.8}$. $G_{3.8}$ is defined by constructing the adversary $\mathcal{G} = (\mathcal{F}_0, \mathcal{G}_1)$ from the adversary $\mathcal{F} = (\mathcal{F}_0, \mathcal{F}_1)$, as shown in Figure 10. The adversary $\mathcal{G}$ is now playing an $\mathsf{IND\text{-}CPA}$ game with PKE for a fixed key pair (pk, sk). Similar to the analysis in $G_{2.7}$, it holds that:

$$|\Pr[G_{3.8}^{\mathcal{A}} \Rightarrow 1 \wedge P] - \frac{1}{2}| = \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{G}).$$

Also, since $G_{3.8} \Rightarrow 1$ holds if $G_{3.7} \Rightarrow 1$ hold, the following holds:

$$\begin{aligned}\Pr[G_{3.8} \Rightarrow 1 \wedge P] &= \Pr[G_{3.7} \Rightarrow 1 \wedge P] + \frac{1}{2}(1 - \Pr[G_{3.7} \Rightarrow 1 \wedge P]) \\ &= \frac{1}{2}\Pr[G_{3.7} \Rightarrow 1 \wedge P] + \frac{1}{2}.\end{aligned}$$

The above equality can be simplified as follows:

$$\Pr[G_{3.7} \Rightarrow 1 \wedge P] = 2\Pr[G_{3.8} \Rightarrow 1 \wedge P] - 1 \leq 2\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{G}).$$

Combining the analyses from G_3 to $G_{3.8}$ so far, the following inequality holds:

$$\begin{aligned}\Pr[G_3^{\mathcal{A}} \Rightarrow 1] &= \Pr[G_{3.1}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{3.2}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{3.3}^{\mathcal{A}} \Rightarrow 1] = \Pr[G_{3.4}^{\mathcal{A}} \Rightarrow 1] \\ &\leq \Pr[G_{3.4}^{\mathcal{A}} \Rightarrow 1 \wedge P^{\dagger}] + \varepsilon_{1,sk} \\ &\leq \Pr[G_{3.5}^{\mathcal{A}} \Rightarrow 1 \wedge P] + \varepsilon_{2,sk} + \varepsilon_{1,sk} = \Pr[G_{3.6}^{\mathcal{A}} \Rightarrow 1 \wedge P] + \varepsilon_{2,sk} + \varepsilon_{1,sk} \\ &\leq \Pr[G_{3.7}^{\mathcal{A}} \Rightarrow 1 \wedge P] + \varepsilon_{3,sk} + \varepsilon_{2,sk} + \varepsilon_{1,sk} \\ &= 2\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{G}) + \varepsilon_{sk}. \end{aligned} \tag{3}$$

The claimed bound is obtained by combining inequalities (1), (2), and (3) as follows and then taking the expectation over $(pk, sk) \leftarrow \mathsf{Gen}'(1^{\lambda})$:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE}',sk}(\mathcal{A}) &\leq 2 \cdot (q_{\mathsf{H}} + q_{\mathsf{D}})\sqrt{\Pr[G_3 \Rightarrow 1]} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2} + |\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \frac{1}{2}| \\ &\leq 2 \cdot (q_{\mathsf{H}} + q_{\mathsf{D}})\sqrt{2\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{G}) + \varepsilon_{sk}} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2} + \mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{E}) + \varepsilon_{sk} \\ &\leq (2q_{\mathsf{H}} + 2q_{\mathsf{D}} + 1)\sqrt{2\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\mathsf{PKE},sk}(\mathcal{G}) + \varepsilon_{sk}} + (q_{\mathsf{H}} + q_{\mathsf{D}}) \cdot 2^{-\ell_r/2+2}.\end{aligned}$$

□

5 $\mathsf{FO}_{\mathsf{PKE}}$-Equivalent Transformation Without Re-encryption

In the previous section, we presented the construction of $\mathsf{PKE}'' := \mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$, which can be proven to be IND-CCA secure in the (Q)ROM. When using $\mathsf{PKE}' := \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$ as the underlying scheme, the resulting $\mathsf{PKE}'' = (\mathsf{Gen}'', \mathsf{Enc}'', \mathsf{Dec}'')$ can be obtained as shown in Figure 11. However, re-encryption during the decryption of PKE'' is necessary to verify whether a ciphertext c is valid. As shown in [23], $\mathsf{FO}_{\mathsf{PKE}}$ based on ACWC_2 can be identically converted into a more efficient transformation, denoted as $\overline{\mathsf{FO}}_{\mathsf{PKE}}$ (shown in Figure 13), where the ciphertext comparison $c \stackrel{?}{=} \mathsf{Enc}'(pk, \tilde{m}'; R')$ in Dec'' is replaced with a simpler comparison $r' \stackrel{?}{=} r''$. This is achieved by first changing Dec'' from Figure 11 to the version in Figure 12, which are conceptually identical. The equivalence of Dec'' in Figure 12 and that in Figure 13 is proven in Lemma 5.1, primarily based on the injectivity and rigidity of PKE. Additionally, it is required that the decryption output of PKE is always within its message space $\mathcal{M}$, meaning it does not produce $\perp$, which is a weak requirement that can be easily satisfied. A typical example is $\mathsf{GenNTRU}[\psi_1^n]$, described in Figure 15, where the use of modulo 3 during decryption ensures that a recovered message always belongs to its message space. Since the proof of Lemma 5.1 is independent of the (Q)ROM, the resulting schemes $\mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$ and $\overline{\mathsf{FO}}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$ operate identically.

$\underline{\mathsf{Gen}''(1^\lambda)}$

1: $(pk, sk) := \mathsf{Gen}'(1^\lambda)$
2: **return** (pk, sk)

$\underline{\mathsf{Enc}''(pk, m \in \{0,1\}^{\ell_m})}$

1: $r \leftarrow \{0,1\}^{\ell_r}$
2: $\tilde{m} = m || r \in \{0,1\}^{\ell_m + \ell_r}$
3: $R := \mathsf{H}(\tilde{m})$
4: $c := \mathsf{Enc}'(pk, \tilde{m}; R)$
   - $r \leftarrow \psi_{\mathcal{R}}$ using the randomness R
   - $M := \mathsf{Encode}(\tilde{m}, \mathsf{G}(r))$
   - $c := \mathsf{Enc}(pk, m; r)$
5: **return** c

$\underline{\mathsf{Dec}''(sk, c)}$

1: $\tilde{m}' = \mathsf{Dec}'(sk, c)$
   - $M' = \mathsf{Dec}(sk, c)$
   - $r' = \mathsf{RRec}(pk, M', c)$
   - $\tilde{m}' = \mathsf{Inv}(M', \mathsf{G}(r'))$
   - **if** $r' \notin \mathcal{R}$ or $\tilde{m}' = \perp$, **return** $\perp$
   - **return** $\tilde{m}'$
2: $R' := \mathsf{H}(\tilde{m}')$
3: **if** $\tilde{m}' = \perp$ **or** $c \neq \mathsf{Enc}'(pk, \tilde{m}'; R')$
4: **return** $\perp$
5: **else**
6: **return** $[\tilde{m}']_{\ell_m}$

Figure 11: $\mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$ with $\mathsf{PKE}' := \mathsf{ACWC}_2[\mathsf{PKE}, \mathsf{SOTP}, \mathsf{G}]$

Lemma 5.1. Assume that the output of Dec in PKE always belongs to $\mathcal{M}$, PKE is injective in the injectivity game of Figure 1, and PKE and SOTP are rigid. Then, $r' \in \mathcal{R}$ and $c = \mathsf{Enc}'(pk, \tilde{m}'; R')$ in $\mathsf{FO}_{\mathsf{PKE}}$ holds if and only if $r' = r''$ in $\overline{\mathsf{FO}}_{\mathsf{PKE}}$ holds.

Proof. Assume that $\tilde{m}' \neq \perp$, $r' \in \mathcal{R}$, and $c = \mathsf{Enc}'(pk, \tilde{m}'; R')$ holds in the Dec'' of $\mathsf{FO}_{\mathsf{PKE}}$. By the definition of Enc', we have $c = \mathsf{Enc}(pk, \mathsf{Encode}(\tilde{m}', \mathsf{G}(r''))); r'')$, where $r'' \leftarrow \psi_{\mathcal{R}}$ is sampled using the randomness R'. Furthermore, since $M' = \mathsf{Dec}(sk, c) \in \mathcal{M}$ and $r' = \mathsf{RRec}(pk, M', c) \in \mathcal{R}$, the rigidity of the PKE leads to the equality $c = \mathsf{Enc}(pk, M'; r')$. Because PKE is injective, these two equations with respect to c imply that $r' = r''$.

Conversely, assume that $\tilde{m}' \neq \perp$ and $r' = r''$ holds for a ciphertext c in the Dec'' of $\overline{\mathsf{FO}}_{\mathsf{PKE}}$. By the rigidity of the SOTP, $\tilde{m}' = \mathsf{Inv}(M', \mathsf{G}(r')) \neq \perp$ implies $M' = \mathsf{Encode}(\tilde{m}', \mathsf{G}(r'))$, thus $M' =$

$\underline{\mathsf{Dec}''(sk, c)}$
1: $M' = \mathsf{Dec}(sk, c)$
2: $r' = \mathsf{RRec}(pk, M', c)$
3: $\tilde{m}' = \mathsf{Inv}(M', \mathsf{G}(r'))$
4: $R' := \mathsf{H}(\tilde{m}')$
5: **if** $\tilde{m}' = \perp$ or $\boxed{r' \notin \mathcal{R} \text{ or } c \neq \mathsf{Enc}'(pk, \tilde{m}'; R')}$
6: **return** $\perp$
7: **else**
8: **return** $[\tilde{m}']_{\ell_m}$

Figure 12: Dec'' of $\mathsf{PKE}'' = \mathsf{FO}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$

$\underline{\mathsf{Dec}''(sk, c)}$
1: $M' = \mathsf{Dec}(sk, c)$
2: $r' = \mathsf{RRec}(pk, M', c)$
3: $\tilde{m}' = \mathsf{Inv}(M', \mathsf{G}(r'))$
4: $R' := \mathsf{H}(\tilde{m}')$
5: $\boxed{r'' \leftarrow \psi_{\mathcal{R}} \text{ with the randomness } R'}$
6: **if** $\tilde{m}' = \perp$ or $\boxed{r' \neq r''}$
7: **return** $\perp$
8: **else**
9: **return** $[\tilde{m}']_{\ell_m}$

Figure 13: Dec'' of $\mathsf{PKE}'' = \overline{\mathsf{FO}}_{\mathsf{PKE}}[\mathsf{PKE}', \mathsf{H}]$

$\mathsf{Encode}(\tilde{m}', \mathsf{G}(r''))$. Also, since $r'' \leftarrow \psi_{\mathcal{R}}$ is sampled using the randomness R' and $r' = r''$, it follows that $r' \in \mathcal{R}$. Since $M' = \mathsf{Dec}(sk, c) \in \mathcal{M}$ and $r' = \mathsf{RRec}(pk, M', c) \in \mathcal{R}$, by the rigidity of the PKE, $c = \mathsf{Enc}(pk, \mathsf{Dec}(sk, c); r') = \mathsf{Enc}(pk, \mathsf{Encode}(\tilde{m}', \mathsf{G}(r'')); r'') = \mathsf{Enc}'(pk, \tilde{m}'; R')$ holds. □

6 Instantiation of NTRU+PKE

In this section, we present our NTRU+PKE scheme, which can be proven to be IND-CCA secure in the (Q)ROM. We begin by adapting two central components, $\mathsf{GenNTRU}[\psi_1^n]$ and SOTP, from [23]. Next, by sequentially applying the ACWC_2, $\mathsf{FO}_{\mathsf{PKE}}$, and $\overline{\mathsf{FO}}_{\mathsf{PKE}}$ transformations, we obtain the final NTRU+PKE as the resulting scheme. Figure 14 provides an overview of the security reductions used to achieve NTRU+PKE.

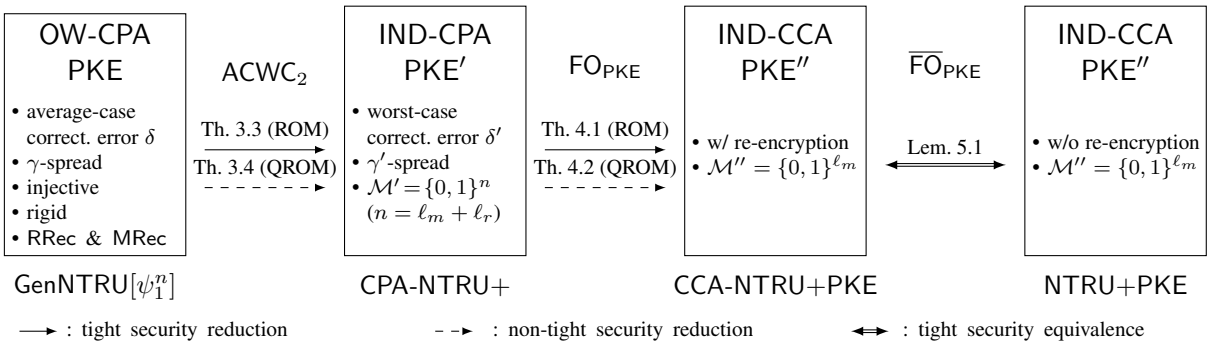

Figure 14: Overview of Security Reductions for NTRU+PKE

6.1 GenNTRU$[\psi_1^n]$ and SOTP Constructions

Let $R_q = \mathbb{Z}_q[x]/\langle f(x) \rangle$ be a polynomial-based ring for a modulus q and a degree n of a polynomial $f(x)$, and let ψ_1^n be n consecutive centered binomial distributions, each obtained by subtracting two uniformly random bits from one another. Figure 15 presents $\mathsf{GenNTRU}[\psi_1^n]$ relative to the distribution ψ_1^n over R_q, including two additional algorithms: RRec and MRec. To apply the ACWC_2, $\mathsf{FO}_{\mathsf{PKE}}$, and

$\underline{\mathsf{Gen}(1^\lambda)}$
1: $\mathbf{f}', \mathbf{g} \leftarrow \psi_1^n$
2: $\mathbf{f} = 3\mathbf{f}' + 1$
3: **if** $\mathbf{f}$, $\mathbf{g}$ is not invertible in R_q
4: restart
5: $\mathbf{h} = 3\mathbf{g}\mathbf{f}^{-1}$
6: **return** $(pk, sk) = (\mathbf{h}, \mathbf{f})$

$\underline{\mathsf{Enc}(\mathbf{h}, \mathbf{m} \leftarrow \psi_1^n; \mathbf{r} \leftarrow \psi_1^n)}$
1: **return** $\mathbf{c} = \mathbf{h}\mathbf{r} + \mathbf{m}$

$\underline{\mathsf{Dec}(\mathbf{f}, \mathbf{c})}$
1: **return** $\mathbf{m} = (\mathbf{c}\mathbf{f} \bmod q) \bmod 3$

$\underline{\mathsf{RRec}(\mathbf{h}, \mathbf{m}, \mathbf{c})}$
1: **return** $\mathbf{r} = (\mathbf{c} - \mathbf{m})\mathbf{h}^{-1}$

$\underline{\mathsf{MRec}(\mathbf{h}, \mathbf{r}, \mathbf{c})}$
1: **return** $\mathbf{m} = \mathbf{c} - \mathbf{h}\mathbf{r}$

Figure 15: $\mathsf{GenNTRU}[\psi_1^n]$ $(= \mathsf{PKE})$

$\overline{\mathsf{FO}}_{\mathsf{PKE}}$ transformations, $\mathsf{GenNTRU}[\psi_1^n]$ (as the underlying PKE) must have the following properties: (1) OW-CPA security, (2) average-case correctness error δ, (3) γ-spreadness, (4) injectivity, and (5) rigidity. [23] already showed that $\mathsf{GenNTRU}[\psi_1^n]$ satisfies all those properties. In particular, the OW-CPA security of $\mathsf{GenNTRU}[\psi_1^n]$ is based on the *decisional* NTRU and the *computational* Ring Learning-with-Errors (RLWE) problems with respect to the parameters (n, q, ψ_1^n). We restate the following theorems from [23].

Theorem 6.1 (OW-CPA security of $\mathsf{GenNTRU}[\psi_1^n]$)**.** For any adversary $\mathcal{A}$, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{GenNTRU}[\psi_1^n]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{NTRU}}_{n,q,\psi_1^n}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{RLWE}}_{n,q,\psi_1^n}(\mathcal{C}).$$

Next, Figure 16 presents the SOTP construction, which is designed to fit the distribution ψ_1^n well. Using ACWC_2 based on SOTP, $\mathsf{GenNTRU}[\psi_1^n]$ can be transformed into an IND-CPA secure PKE', called CPA-NTRU+, which achieves worst-case δ'-correctness and (weak) γ'-spreadness. Theorem 3.2 shows that the worst-case correctness error δ' of CPA-NTRU+ is nearly identical to the average-case error δ of $\mathsf{GenNTRU}[\psi_1^n]$. Additionally, Theorem 3.5 bounds the γ'-spreadness of CPA-NTRU+ by $\gamma' = \gamma - \log_2(|\mathcal{M}| \cdot \max_{M \in \mathcal{M}} \psi_{\mathcal{M}}(M))$, where γ-spreadness and $\mathcal{M}$ refer to those of $\mathsf{GenNTRU}[\psi_1^n]$. For the message space $\mathcal{M} = \{-1, 0, 1\}^n$ according to ψ_1^n, we have $|\mathcal{M}| = 3^n$ and $\max_{M \in \mathcal{M}} \psi_{\mathcal{M}}(M) = 2^{-n}$. Since $\mathsf{GenNTRU}[\psi_1^n]$ is n-spread [23], the value of γ' becomes $n - \log_2(3^n \cdot 2^{-n}) \approx 0.415n$. For instance, with $n = 768$ in $\mathsf{GenNTRU}[\psi_1^n]$, CPA-NTRU+ is approximately 318-spread. Moreover, due to SOTP with respect to ψ_1^n, the message space $\mathcal{M}'$ of CPA-NTRU+ becomes a very natural set of bit-strings with arbitrary distributions. Indeed, $\mathcal{M}' = \{0, 1\}^n$ is sufficient to apply $\mathsf{FO}_{\mathsf{PKE}}$ for parameters with a large degree n.

$\underline{\mathsf{Encode}(x \in \mathcal{M}' = \{0,1\}^n, u \leftarrow U^{2n})}$
1: $u = (u_1, u_2) \in \{0,1\}^n \times \{0,1\}^n$
2: $y = (x \oplus u_1) - u_2 \in \{-1, 0, 1\}^n$
3: **return** y

$\underline{\mathsf{Inv}(y \in \mathcal{M} = \{-1,0,1\}^n, u \in \{0,1\}^{2n})}$
1: $u = (u_1, u_2) \in \{0,1\}^n \times \{0,1\}^n$
2: **if** $y + u_2 \notin \{0,1\}^n$, **return** $\perp$
3: $x = (y + u_2) \oplus u_1 \in \{0,1\}^n$
4: **return** x

Figure 16: SOTP instantiation

6.2 NTRU+PKE Construction

We now achieve an IND-CCA secure PKE'' (denoted as CCA-NTRU+PKE in Figure 14) by applying $\mathsf{FO}_{\mathsf{PKE}}$ to CPA-NTRU+. Next, we obtain our final scheme NTRU+PKE by applying $\overline{\mathsf{FO}}_{\mathsf{PKE}}$ to CCA-NTRU+PKE, described as $\mathsf{NTRU+PKE} := \overline{\mathsf{FO}}^{\perp}_{\mathsf{PKE}}[\mathsf{CCA\text{-}NTRU+PKE}, \mathsf{H}]$. Figure 17 shows the resultant NTRU+PKE, which is the basis of our implementation in the next subsection. By combining Theorems 4.1, 4.2, and Lemma 5.1, we achieve the IND-CCA security of NTRU+PKE in the (Q)ROM, based on the two assumptions $\mathsf{NTRU}_{n,q,\psi_1^n}$ and $\mathsf{RLWE}_{n,q,\psi_1^n}$. As in NTRU+KEM [23], NTRU+PKE preserves the worst-case correctness error of the underlying CPA-NTRU+.

$\underline{\mathsf{Gen}''(1^\lambda)}$

1: $\mathbf{f}', \mathbf{g} \leftarrow \psi_1^n$
2: $\mathbf{f} = 3\mathbf{f}' + 1$
3: **if** $\mathbf{f}$, $\mathbf{g}$ are not invertible in R_q, restart
4: **return** $(pk, sk) = (\mathbf{h} = 3\mathbf{g}\mathbf{f}^{-1}, \mathbf{f})$

$\underline{\mathsf{Enc}''(pk, m \in \{0,1\}^{\ell_m})}$

1: $r \leftarrow \{0,1\}^{\ell_r}$
2: $\tilde{m} = m||r \in \{0,1\}^{n=\ell_m+\ell_r}$
3: $R = \mathsf{H}(\tilde{m})$
4: $\mathbf{r} \leftarrow \psi_1^n$ using the randomness R
5: $\mathbf{m} = \mathsf{Encode}(\tilde{m}, \mathsf{G}(\mathbf{r}))$
6: $\mathbf{c} = \mathbf{h}\mathbf{r} + \mathbf{m}$
7: **return** $\mathbf{c}$

$\underline{\mathsf{Dec}''(sk, \mathbf{c})}$

1: $\mathbf{m} = (\mathbf{c}\mathbf{f} \bmod q) \bmod 3$
2: $\mathbf{r} = (\mathbf{c} - \mathbf{m})\mathbf{h}^{-1}$
3: $\tilde{m} = \mathsf{Inv}(\mathbf{m}, \mathsf{G}(\mathbf{r}))$
4: $R' = \mathsf{H}(\tilde{m})$
5: $\mathbf{r}' \leftarrow \psi_1^n$ using the randomness R'
6: **if** $\tilde{m} = \perp$ or $\mathbf{r} \neq \mathbf{r}'$
7: **return** $\perp$
8: **else**
9: **return** $[\tilde{m}]_{\ell_m}$

Figure 17: NTRU+PKE

6.3 Parameters and Implementations

Table 1 presents four parameter sets for NTRU+PKE. We refer to them as NTRU+PKE{576, 768, 864, 1152}, respectively, based on the degree n of $f(x)$ over $R_q = \mathbb{Z}_q[x]/\langle f(x)\rangle$. Following [23], we set $f(x) = x^n - x^{n/2} + 1$ and $q = 3457$ to perform the Number-Theoretic Transform (NTT) in all parameter sets. The worst-case correctness error δ' of CPA-NTRU+ (and thus NTRU+PKE) was calculated [23] by adding the average-case correctness error δ of $\mathsf{GenNTRU}[\psi_1^n]$ and the value $\Delta = \|\psi_{\mathcal{R}}\|(1+\sqrt{(\ln|\mathcal{M}'| - \ln\|\psi_{\mathcal{R}}\|)/2})$ using the equation from Theorem 3.2, where $\psi_{\mathcal{R}} = \psi_1^n$ and $\|\psi_{\mathcal{R}}\| := \sqrt{\sum_r \psi_{\mathcal{R}}(r)^2}$ and $\mathcal{M}' = \{0,1\}^n$. Additionally, the γ'-spreadness of CPA-NTRU+ is computed as $\gamma' \approx 0.415n$, as mentioned above. For each parameter set of NTRU+PKE, we set $\ell_m = 33$ bytes to encrypt messages of up to 32 bytes (including a 1-byte indicator that specifies the start of a message), with $\ell_r = n/8 - \ell_m$ bytes allocated for randomness. Regarding the concrete security level of NTRU+PKE, we analyze the classical and quantum hardness of the two problems $\mathsf{RLWE}_{n,q,\psi_1^n}$ and $\mathsf{NTRU}_{n,q,\psi_1^n}$ for each parameter set, using the Lattice estimator [1] and the NTRU estimator [8]. Currently, the concrete security level of the NTRU problem is similar to that of the RLWE problem when using the Lattice and NTRU estimators. Recently, Lee *et al.* [24] proposed a combinatorial attack that improves upon May's Meet-LWE attack [26] and analyzed the concrete security level of NTRU+KEM, which uses the same parameters as NTRU+PKE. Their analysis demonstrated that

Table 1: Parameter Sets for NTRU+PKE

Scheme	Security		n	q	pk	ct	sk	(ℓ_m, ℓ_r)	$\log_2 \delta'$	γ'
	c	q								
NTRU+PKE576	114	101	576	3457	864	864	1760	(33,39)	-487	239
NTRU+PKE768	164	144	768	3457	1152	1152	2336	(33,63)	-379	318
NTRU+PKE864	189	166	864	3457	1296	1296	2624	(33,75)	-340	358
NTRU+PKE1152	263	233	1152	3457	1728	1728	3488	(33,111)	-260	478

c: classical. q: quantum. n: polynomial degree. q: modulus.
$(pk, ct, sk, \ell_m, \ell_r)$: bytes. δ': (worst-case) correctness error. γ': (weak) spreadness.

the security of NTRU+KEM against their combinatorial attack does not degrade below the level predicted by the above Lattice and NTRU estimators.

Table 2 compares KYBER [30], the NTRU finalist [8], and NTRU+PKE. For a fair comparison, KYBER and the NTRU finalist are implemented with AES-256-GCM to encrypt a 256-bit message using the KEM-DEM framework. At an approximate 180-bit classical security level, we compare NTRU+PKE864 with KYBER768 and ntruhps4096821 in terms of encryption/decryption cycles in AVX2 mode. As shown in Table 2, NTRU+PKE864 is approximately $98/49 \approx 2$ times faster than KYBER768 + AES-256-GCM and approximately $489/49 \approx 9.9$ times faster than ntruhps4096821 + AES-256-GCM. Regarding ciphertext size, KYBER and the NTRU finalist include an additional 48 bytes (32 bytes for encryption and 16 bytes for authentication), compared to their KEMs.

Table 2: Comparison between NTRU+PKE, finalist NTRU, and KYBER

Scheme	Security		pk	ct	$\log_2 \delta'$	reference			AVX2		
	c	q				Gen	Enc	Dec	Gen	Enc	Dec
NTRU+PKE576	114	101	864	864	-487	195	80	98	24	22	13
NTRU+PKE768	164	144	1152	1152	-379	259	107	137	27	27	17
NTRU+PKE864	189	166	1296	1296	-340	241	124	162	27	30	19
NTRU+PKE1152	263	233	1728	1728	-260	495	160	227	52	39	26
KYBER512*	118	104	800	816	-139	115	139	160	36	42	27
KYBER768*	182	160	1184	1232	-164	182	205	233	51	58	40
KYBER1024*	255	224	1568	1616	-174	269	321	360	64	76	56
ntruhps2048509*	104	93	699	747	$-\infty$	7979	746	1383	373	262	37
ntruhrss701*	133	119	1138	1186	$-\infty$	14585	1026	2617	362	168	55
ntruhps2048677*	144	127	930	978	$-\infty$	13789	1197	2435	541	349	52
ntruhps4096821*	178	158	1230	1278	$-\infty$	20253	1638	3508	704	423	66

c: classical. q: quantum. (pk, ct): bytes. δ': (worst-case or perfect) correctness error. (Gen, Enc, Dec): K cycles of reference/AVX2 implementations. $*$: means that 32-byte messages are encrypted using AES-256-GCM.

References

[1] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *J. Math. Cryptol.*, 9(3):169–203, 2015.

[2] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016: 25th USENIX Security Symposium*, pages 327–343, Austin, TX, USA, August 10–12, 2016. USENIX Association.

[3] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part II*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295, Santa Barbara, CA, USA, August 18–22, 2019. Springer, Cham, Switzerland.

[4] X9 ANSI. 98: Lattice-based polynomial public key establishment algorithm for the financial services industry. Technical report, Technical report, ANSI, 2010.

[5] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111, Perugia, Italy, May 9–12, 1995. Springer, Berlin, Heidelberg, Germany.

[6] Daniel J. Bernstein and Edoardo Persichetti. Towards KEM unification. Cryptology ePrint Archive, Report 2018/526, 2018. https://eprint.iacr.org/2018/526.

[7] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 61–90, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland.

[8] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[9] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[10] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part III*, volume 13277 of *Lecture Notes in Computer Science*, pages 677–706, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland.

[11] Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, Gregor Seiler, and Dominique Unruh. A thorough treatment of highly-efficient NTRU instantiations. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023: 26th International Conference on Theory and Practice of*

Public Key Cryptography, Part I, volume 13940 of *Lecture Notes in Computer Science*, pages 65–94, Atlanta, GA, USA, May 7–10, 2023. Springer, Cham, Switzerland.

[12] Consortium for Efficient Embedded Security. EESS #1: Implementation aspects of NTRU-Encrypt and pqNTRUSign v.3.3. Technical report, Consortium for Efficient Embedded Security, June 2017.

[13] Pierre-Alain Fouque, Paul Kirchner, Thomas Pornin, and Yang Yu. BAT: Small and fast KEM over NTRU lattices. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(2):240–265, 2022.

[14] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *PKC'99: 2nd International Workshop on Theory and Practice in Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68, Kamakura, Japan, March 1–3, 1999. Springer, Berlin, Heidelberg, Germany.

[15] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Berlin, Heidelberg, Germany.

[16] Jeffrey Hoffstein, Jill Pipher, John M. Schanck, Joseph H. Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for NTRUEncrypt. In Helena Handschuh, editor, *Topics in Cryptology – CT-RSA 2017*, volume 10159 of *Lecture Notes in Computer Science*, pages 3–18, San Francisco, CA, USA, February 14–17, 2017. Springer, Cham, Switzerland.

[17] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *Third Algorithmic Number Theory Symposium (ANTS)*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, June 1998.

[18] Jeffrey Hoffstein and Joseph H. Silverman. Protecting NTRU Against Chosen Ciphertext and Reaction Attacks. *NTRU Cryptosystems Technical Report Report# 016, Version 1*, 2000.

[19] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part I*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371, Baltimore, MD, USA, November 12–15, 2017. Springer, Cham, Switzerland.

[20] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The impact of decryption failures on the security of NTRU encryption. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 226–246, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Heidelberg, Germany.

[21] Nick Howgrave-Graham, Joseph H. Silverman, Ari Singer, and William Whyte. NAEP: Provable security in the presence of decryption failures. Cryptology ePrint Archive, Report 2003/172, 2003. https://eprint.iacr.org/2003/172.

[22] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Third Edition*. CRC Press, 2020.

[23] Jonghyun Kim and Jong Hwan Park. NTRU+: Compact Construction of NTRU Using Simple Encoding Method. *IEEE Transactions on Information Forensics and Security*, 18:4760–4774, 2023.

[24] Eunmin Lee, Joohee Lee, and Yuntao Wang. Improved Meet-LWE Attack via Ternary Trees. Cryptology ePrint Archive, Report 2024/824, 2024.

[25] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Heidelberg, Germany.

[26] Alexander May. How to meet ternary LWE keys. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part II*, volume 12826 of *Lecture Notes in Computer Science*, pages 701–731, Virtual Event, August 16–20, 2021. Springer, Cham, Switzerland.

[27] Phong Q. Nguyen and David Pointcheval. Analysis and improvements of NTRU encryption paddings. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 210–225, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Heidelberg, Germany.

[28] Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. FALCON. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[29] John M. Schanck, Andreas Hulsing, Joost Rijneveld, and Peter Schwabe. NTRU-HRSS-KEM. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.

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

[31] Victor Shoup. OAEP reconsidered. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Heidelberg, Germany.

[32] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. https://eprint.iacr.org/2004/332.

[33] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 27–47, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Heidelberg, Germany.

[34] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II*, volume 9986 of *Lecture Notes in Computer Science*, pages 192–216, Beijing, China, October 31 – November 3, 2016. Springer, Berlin, Heidelberg, Germany.

[35] Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 1–18, Santa Barbara, CA, USA, August 17–21, 2014. Springer, Berlin, Heidelberg, Germany.

[36] Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 129–146, Copenhagen, Denmark, May 11–15, 2014. Springer, Berlin, Heidelberg, Germany.

[37] Jiang Zhang, Dengguo Feng, and Di Yan. NEV: Faster and smaller NTRU encryption using vector decoding. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023, Part VII*, volume 14444 of *Lecture Notes in Computer Science*, pages 157–189, Guangzhou, China, December 4–8, 2023. Springer, Singapore, Singapore.

[38] Zhenfei Zhang, Cong Chen, Jeffrey Hoffstein, and William Whyte. NTRUEncrypt. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions.