

OAEP[†] Transform in the Post-Quantum World

You Lyu^{1,2}, Shengli Liu^{1,2}, Shuai Han^{1,2}, and Bohang Chen^{1,2}

¹ School of Computer Science,
Shanghai Jiao Tong University, Shanghai 200240, China

² State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
{`vergil,slliu,dalen17,bohang0918`}@sjtu.edu.cn

Abstract. We provide a new variant of OAEP called OAEP[†], which converts an almost trapdoor injective function (ATIF) to a public-key encryption (PKE) scheme. The resulting PKE not only has CCA security but also enjoys pseudo-randomness, anonymity, and robustness under chosen-ciphertext attacks in the quantum random oracle (QRO) model.

Compared with the plain OAEP and its variants whose structure does not serve the quantum world very well, our OAEP[†] is designed with a new structure, admitting more flexible choices for ATIF and enjoying better security and efficiency. For OAEP[†], we present two instantiations of ATIF from NTRU, which yield two practical post-quantum PKE schemes from lattices. The resulting PKE schemes are comparable to Kyber and NTRU-HPS derived from FO transform. The performance evaluations show that one of our PKE schemes is faster than Kyber512, and the other shares the same basic underlying structure with NTRU-hps2048677 but can additionally encrypt 132 bytes message.

Our OAEP[†] does not use the (third) additional hash function (unlike Q-OAEP) and has a tighter security reduction (than Q-OAEP), and thus answers the open problems proposed by the authors of NTRU, who proposed the NTRU KEM candidates in NIST's third round of post-quantum cryptography standardization. And replacing the FO transform with OAEP[†] in NTRU-HPS (one of NTRU KEM in the third round) yields a CCA-secure PKE scheme that is as efficient as the original NTRU-HPS (from FO) but better than the PKE hybrid from NTRU-HPS and a symmetric encryption.

Besides NTRU, ATIF also has instantiations from isogenies/lattices. So OAEP[†] yields post-quantum CCA-secure PKE/KEM schemes from lattices/isogenies in the QRO model as well, suggesting the wide applicability of OAEP[†] in the quantum world.

Keywords: OAEP, Post Quantum Security, QROM

1 Introduction

Over the decades, there have been abundant treasured cryptographic techniques and frameworks developed for the classic world. For example, there are two

famous transforms that convert weak cryptographic primitives to CCA-secure public-key encryption (PKE) schemes: one is the FO transform, due to Fujisaki and Okamoto [23], which can turn a CPA-secure PKE to a CCA-secure PKE or key encapsulation mechanism (KEM). The other is the OAEP transform, due to Bellare and Rogaway [13], which can convert a trapdoor permutation (TDP) into a CCA-secure PKE scheme. Both of the FO and OAEP transforms rely on hash functions which are modeled as random oracles (RO).

How to take advantage of these nice results and adapt them to the quantum world is definitely an attractive topic. As for the FO and OAEP transforms in the scenario of quantum computing, an adversary may issue hash queries in superposition, which is completely different from the hash query in the classic world. For example, the adversary may submit one hash query in superposition of all inputs and see all corresponding outputs from this single query. Therefore, the classic proving techniques of RO, like keeping hash list or re-programming hash functions, do not work in the quantum world. To adapt FO and OAEP to the quantum world, we have to consider quantum random oracles (QRO), to which the adversary has quantum access.

Up to now, great progress has been made on the FO transform in the QRO model, and a variety of FO variants have been proposed and extensively studied in QRO model [28,37,30,16,29,26,42,25]. It is noteworthy that Kyber, the KEM standardized by NIST, just utilizes the FO transform to achieve CCA security.

In comparison to FO, research on the OAEP transform in QRO model lags behind. Though OAEP has served as a cornerstone for constructing CCA-secure PKE schemes and RSA-OAEP widely deployed in practice [3,2,1], there are only few works examining OAEP and exploring its variants in QRO model, and the results are far from abundant compared to FO, as shown next.

OAEP Transform in QROM. In [21], the CCA security of the plain OAEP transform was established in the QRO model, based on the partial-domain one-wayness (PDOW) of the underlying trapdoor permutation (TDP). Later, Ebrahimi [22] proved that the TDP can be relaxed to normal one-wayness (OW) to support the CCA security of the OAEP+ transform, a variant of OAEP due to Shoup [38]. These are elegant theoretical results in the QRO model. However, finding post-quantum TDP possessing one-wayness or partial-domain one-wayness seems very challenging. As far as we know, there are hardly any TDP instantiations from lattices or isogenies.

In [39], Targhi and Unruh proposed Q-OAEP, a new variant of OAEP requiring only partial-domain one-way Trapdoor Injective Function (TIF) for CCA-security. However, its dependency on an extra length-preserving hash function increases the ciphertext size. For instance, Chen et al. [18] suggested applying Q-OAEP to NTRU to achieve CCA security, but the price is an overhead of at least 126 bytes added to the ciphertext and the loss of optimized parameters.

All the above OAEP/OAEP+/Q-OAEP transforms in [21,22,39] share the same feature: the (partial-domain) one-wayness of TDP and TIF requires that the inputs have uniform distribution over the input space. This is not friendly to instantiations from lattices, since the error (even the secret) always follows

discrete Gaussian, centered binomial, or sparse ternary distributions. As pointed in [18], this discrepancy is particularly acute for the deterministic PKE under NTRU-HPS, which is NIST’s third-round KEM candidate. The underlying deterministic PKE from NTRU employs sparse ternary secrets to optimize ciphertext size and correctness, and it can be regarded as a trapdoor injective function F_{NTRU} . When applying Q-OAEP to F_{NTRU} , the resulting PKE will completely lose the optimized advantage of F_{NTRU} , since it has to cater to the uniform distribution of Q-OAEP and add an extra length-preserving hash. Moreover, the security proof of Q-OAEP in [39] uses O2H lemma three times recursively, and this directly leads to a security decay $((\epsilon^{1/2})^{1/2})^{1/2} = \epsilon^{1/8}$ from the partial-domain one-wayness advantage ϵ of TIF. These features impede the applications of OAEPs in the quantum world. This is why NTRU in the third round of NIST abandoned Q-OAEP and resorted to the FO transform to obtain CCA security for NTRU-HPS. As pointed out in [18], for Q-OAEP,

“Removing the length-preserving hash, and tightening the reduction, are interesting open problems.”

These are significant questions left to be solved for NTRU, especially now that NTT has submitted IETF draft [4] and is striving to standardize NTRU.

Ciphertext Pseudo-Randomness, Anonymity, and Robustness. Besides CCA security, there are other critical security notions like ciphertext pseudo-randomness/anonymity [12] and robustness [7]. Ciphertext pseudo-randomness asks pseudo-randomness of ciphertext under CCA attacks and gives full protection on users’ privacy. Ciphertext anonymity is implied by ciphertext pseudo-randomness and requires ciphertexts do not reveal any information about the underlying public key. Ciphertext robustness requires that the decryption of a ciphertext intended for a user will be rejected by other users. These features are highly desirable in anonymous systems, cryptocurrencies, auction protocols, etc.

A thorough inspection [26, 42] on ciphertext pseudo-randomness/anonymity and robustness has been done to FO and its variants when applying to the NIST PQC candidates. For example, it was shown that Kyber has anonymity [35] but no robustness [42] under CCA attacks. Up to now, there is hardly any such research on OAEP and its variants. This raises another natural question:

How to design OAEP achieving both ciphertext pseudo-randomness and robustness in the QRO model?

Our Contributions. We answer the above open questions with a new OAEP variant named $\text{OAEP}^\dagger$, which converts an Almost Trapdoor Injective Function (ATIF) to CCA-secure PKE/KEM. Our $\text{OAEP}^\dagger$ enjoys the following features.

- (1) *The scheme obtained from $\text{OAEP}^\dagger$ can be a PKE or KEM scheme.* Like OAEP and its variants, $\text{OAEP}^\dagger$ can result in a PKE scheme for some message space $\mathcal{M}$ or a KEM scheme for some encapsulation key space $\mathcal{K}$. Notably, the resulting KEM scheme from our $\text{OAEP}^\dagger$ can not only encapsulate a key but also encrypt a message in $\mathcal{M}$.

- (2) $\text{OAEP}^\dagger$ achieves CCA security with security decay only $\epsilon^{1/2}$ from ϵ -PDOW of ATIF in the QRO model. The building block of our $\text{OAEP}^\dagger$ is ATIF which only requires overwhelming injectivity rather than perfect injectivity. The security decay $\epsilon^{1/2}$ is much better than $\epsilon^{1/8}$ in Q-OAEP.
- (3) $\text{OAEP}^\dagger$ also achieves pseudo-randomness, anonymity and robustness, besides CCA security. Our $\text{OAEP}^\dagger$ is the first OAEP variant achieving anonymity and robustness in the QRO model.
- (4) $\text{OAEP}^\dagger$ avoids an additional hash function and supports flexible pre-image distribution of ATIF. Like OAEP, our $\text{OAEP}^\dagger$ makes use of only two hash functions for leveraging CCA security (no additional hash function involved). Notably, our $\text{OAEP}^\dagger$ is structured with two hash functions in a way differently from OAEP and admits more flexible pre-image distributions for ATIF.
- (5) $\text{OAEP}^\dagger$ admits efficient instantiations from NTRU. Our $\text{OAEP}^\dagger$ yields a variety of post-quantum CCA-secure PKE/KEM schemes in the QRO model, since ATIF admits instantiations from isogenies, lattices and NTRU. Notably, $\text{OAEP}^\dagger$ admits efficient instantiations from NTRU, thanks to our new structure of $\text{OAEP}^\dagger$ and its support to more flexible pre-image distributions for ATIF. In particular, our $\text{OAEP}^\dagger$ transform friendly accommodates the trapdoor injective function F_{NTRU} contained in NTRU-HPS-2048677 (a third-round KEM candidate submitted to NIST) [18] and keeps its optimized advantage while helping it to achieve CCA security, pseudo-randomness, anonymity and robustness.

Comparison between our $\text{OAEP}^\dagger$ and OAEP, OAEP+, Q-OAEP. In Table 1, we compare our $\text{OAEP}^\dagger$ with the plain OAEP and its variants. It is easy to see that our $\text{OAEP}^\dagger$ enjoys better security and efficiency.

There is a recent work [33] that also considers the security of OAEP in the QROM. Different from our work of exploiting new structure of $\text{OAEP}^\dagger$ for efficiency, CCA security and other properties like robustness, the work in [33] still focuses on the plain OAEP, but with a tighter QROM security proof for IND-qCCA, where the adversary has quantum access to the decryption oracle.

Comparison between PKE/KEM from $\text{OAEP}^\dagger$ and those from FO. In Table 2, we compare our $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ (see Sect. 4) instantiated from NTRU via $\text{OAEP}^\dagger$ with other KEM/PKE schemes via the FO transform, including the NIST final standardized post-quantum KEM scheme Kyber512 [10], the Korean final KEM scheme NTRU+KEM576 and PKE scheme NTRU+PKE576 in Korean PQC competition, the KEM candidate NTRU-hps2048677 in NIST’s third round and the NIST standardized HQC-128 [24].

The performance evaluations in Table 2 are obtained by experiments on Intel(R) Core(TM) i7-1068NG7 CPU @ 2.30GHz under macOS 13.3.1. All the FO schemes in Table 2 are evaluated with their official reference versions [10,31,18].

- Unlike FO transform, $\text{OAEP}^\dagger$ does not need re-encryption check in the decryption algorithm, which improves the decryption efficiency and avoids potential side-channel attacks due to re-encryption [40].

Table 1: Comparison to OAEP and its variants. “type of F , F ’s one-wayness, F ’s pre-image distribution” denote the requirements for the function F underlying the OAEP variant. “TDP” means trapdoor permutation, “TIF” means trapdoor injective function, and “ATIF” means almost trapdoor injective function. “OW” denotes one-wayness and “PDOW” denotes partial domain one-wayness. “ciphertext overhead” represents the increased ciphertext size compared to the output size of the underlying function F . “CCA”, “ANO”, and “ROB” represent the indistinguishability, anonymity, and robustness under CCA attacks of the PKE schemes from OAEP (and its variants). “inst” denotes whether there exists any instantiation from post-quantum assumptions. “?” denotes unknown.

Transform	type of F	F ’s one-wayness	F ’s pre-image distribution	ciphertext overhead	CCA	ANO	ROB	inst
OAEP[21,33]	TDP	PDOW	uniform	0	✓	?	?	?
OAEP+[22]	TDP	OW	uniform	0	✓	?	?	?
Q-OAEP[39]	TIF	PDOW	uniform	$ \mathcal{X} \geq 126$ bytes	✓	?	?	✓
Our OAEP [†]	ATIF	PDOW	flexible	32 bytes	✓	✓	✓	✓

- Compared to the KEM schemes like Kyber512, NTRU+KEM576, NTRU-hps2048677 and HQC-128, our $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ can additionally encrypt some suitable-length messages. For example, $\text{KEM}_{\text{NTRU}}^1$ additionally encrypts an 81-byte message while $\text{KEM}_{\text{NTRU}}^2$ a 132-byte message, besides encapsulating keys.
- Compared to the NIST standardized KEM scheme Kyber512 [10], our $\text{KEM}_{\text{NTRU}}^1$ is faster. $\text{KEM}_{\text{NTRU}}^2$ shares a same basic underlying NTRU with NTRU-hps2048677, keeps optimized parameters of the basic NTRU and is comparable to the FO-variant NTRU-hps2048677.
- Compared to NTRU+KEM576 and NTRU+PKE576, the final scheme of Korean PQC competition [6], our $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ are a little bit slower. However, the security proofs of NTRU+KEM576 and NTRU+PKE576 rely on computational injectivity of F , and has a security gap as pointed out in [14] (See Sect. 3.4.2 in [14]). This gap was admitted by its submitter [5].

Technique Overview. We first recall the common structure of the existing OAEP variants and explain why they are not friendly to post-quantum instantiations. Then we present our new design of OAEP[†] and show in high-level why our new structure admits more flexible pre-image distribution of the core function F . Finally we show a rough overview for proving CCA security for our OAEP[†].

(1) OAEP, OAEP+, Q-OAEP: common structure and its dilemma.

The OAEP and its variants have a core trapdoor permutation or injective function (F, F^{-1}) and two hash functions G, H' (and additional hash function H'' for Q-OAEP) as building blocks, and they all adopt two-round Feistel network

Table 2: Comparison to the KEM/PKE schemes via the FO transform. All schemes are under NIST security level I. $|pk|, |ct|, |sk|$ represent the byte size of public key, ciphertext and secret key. $T(\text{Alg})$ represents the kilo cycles of the corresponding algorithm of KEM/PKE schemes. The PKE schemes are obtained via KEM+DEM(AES) paradigm and tested with message length of 128 bytes. Bit security for all schemes are determined by the lattice estimator [8] except that the 128-bit security for HQC-128, and “Bit security” represents estimated core-problem security as in [34].

KEM (via transform)	$ pk $	$ ct $	$ sk $	$T(\text{Gen})$	$T(\text{Encap})$	$T(\text{Decap})$	Bit security
Kyber512 (FO)[10]	800	768	1632	71	83	91	140
NTRU+KEM576 (FO)[31]	864	864	1760	99	54	60	138
NTRU+PKE576(FO)[31]	864	864	1760	113	53	56	138
NTRU-hps2048677 (FO)[18]	930	930	1234	9716	204	412	170
HQC-128 (FO)[24]	2249	4433	2305	313	790	2366	128
Our $\text{KEM}_{\text{NTRU}}^1$ (OAEP [†])	1053	1085	2170	128	59	60	144
Our $\text{KEM}_{\text{NTRU}}^2$ (OAEP [†])	930	962	1234	9403	216	404	170
PKE = KEM+DEM	$ pk $	$ ct $	$ sk $	$T(\text{Gen})$	$T(\text{Enc})$	$T(\text{Dec})$	Bit security
Kyber512 [10]	800	924	1632	71	92	95	140
NTRU+KEM576 [31]	864	1020	1760	99	61	63	138
NTRU+PKE576 [31]	864	1020	1760	111	63	61	138
NTRU-hps2048677 [18]	930	1086	1234	9716	212	422	170
HQC-128 [24]	2249	4589	2305	313	799	2372	128
Our $\text{KEM}_{\text{NTRU}}^1$ (OAEP [†])	1053	1160	2170	128	62	62	144
Our $\text{KEM}_{\text{NTRU}}^2$ (OAEP [†])	930	962	1234	9403	216	404	170

to couple (F, F^{-1}) with G, H' . See Fig. 1 for the common structure of OAEP, OAEP+ and Q-OAEP for constructing PKE³.

For example, let m be the plaintext and r be the randomness. Then the ciphertext ct is given by

$$ct := c = F(x_1, x_2) \text{ where } x_1 := G(r) + m \text{ and } x_2 := H'(x_1) + r,$$

and its security heavily relies on the one-wayness of F .

To support decryption, $G(r)$ and $H'(x_1)$ must take some groups $(\mathbb{G}_1, +), (\mathbb{G}_2, +)$ as their co-domains. When G, H' are modeled as random oracles, $G(r)$ and $H'(x_1)$ are uniform over $\mathbb{G}_1, \mathbb{G}_2$. Consequently, OAEP (and its variants) needs one-wayness of $F : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{Y}$ supported by the uniform distribution over $\mathcal{X}_1 \times \mathcal{X}_2 = \mathbb{G}_1 \times \mathbb{G}_2$.

The two hash functions impose a strict restriction on the distribution of F 's pre-image. This is not very friendly to the current post-quantum trapdoor injective functions. For example, the injective one-way function $\mathbf{y} = F(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e}$ over lattices always requires that $\mathbf{e}$ (even $\mathbf{s}$) satisfy discrete Gaussian, centered

³ Fig. 1 is just a rough demonstration, neglecting the padding of m in OAEP and OAEP+, and the additional hash in Q-OAEP.

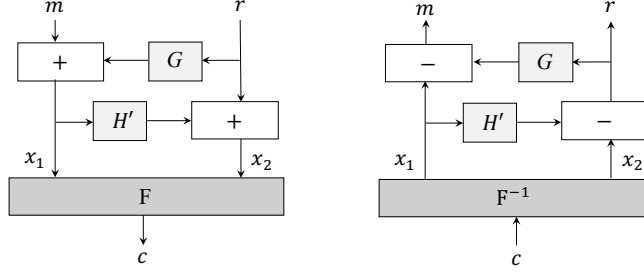


Fig. 1: Overview of the common structure of OAEP and its variants: encryption (left) & decryption (right). The ciphertext is $ct := c$.

binomial, or sparse ternary distributions to accommodate to practical applications. Such distributions are far from uniform distribution and do not work well with OAEP. This partially explains why the existing post-quantum PKE/KEM candidate/final constructions in NIST and Korean PQC competitions all resort to the FO transform, rather than OAEP, to achieve both efficiency and CCA security.

(2) Our OAEP[†]: new structure for flexible pre-image distribution.

To avoid the limitations from the two hash functions, we design new structure for our OAEP[†] to construct PKE. See Fig. 2 for details.

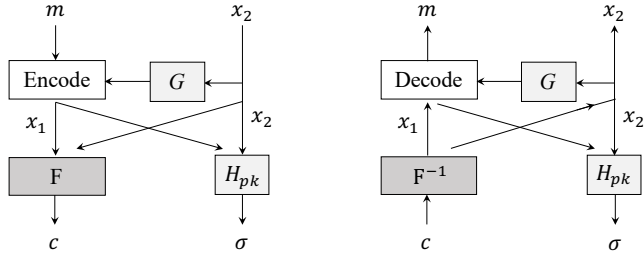


Fig. 2: Overview of our OAEP[†]: encryption (left) & decryption (right). The ciphertext is $ct = (c, \sigma)$, where $c := F(x_1, x_2)$ for $x_1 := \text{Encode}(m, G(x_2))$ and $\sigma := H_{pk}(x_1, x_2)$.

Firstly, we remove the hash function $H'(\cdot)$ and let $r = x_2$ directly serve as the second input of F . Therefore, x_2 gets rid of the limitation from H' , and it can freely take any distribution ψ_2 over $\mathcal{X}_2$.

Secondly, we replace the group operation “+” (in encryption) and “−” (in decryption) with the Encode and Decode algorithms of a so-called semi-generalized one-time pad (SOTP) scheme [31]. Consequently, $x_1 := \text{Encode}(m, G(x_2))$ serves as the first input of F , and $m := \text{Decode}(x_1, G(x_2))$ can be correctly recovered.

We require that $\text{Encode}(m, r)$ has a same distribution ψ_1 for any m when r is uniform. This is captured by the ψ_1 -message hiding property of SOTP. If G is a random oracle and $G(x_2)$ is uniform, then $x_1 := \text{Encode}(m, G(x_2))$ follows the distribution ψ_1 . We stress that ψ_1 is not necessarily a uniform distribution.

Now the pre-image distribution is $\psi = (\psi_1, \psi_2)$ for the core function F , so we only require one-wayness of F for $\psi = (\psi_1, \psi_2)$, admitting more flexible choices for the pre-image distribution.

Finally, to support CCA security, there must be a mechanism to reject invalid ciphertexts in decryption. Different from OAEP and OAEP+, our OAEP[†] does not have padding in m , so we have to explore a new way of rejecting invalid ciphertexts. To this end, we introduce a hash function $H(pk, x_1, x_2) = H_{pk}(x_1, x_2)$ to compute an authenticator, and add it in the ciphertext. Now the ciphertext is $ct = (c, \sigma)$. During decryption, x_1, x_2 can be recovered from $F^{-1}(c)$ and the validity of ct can be checked with $\sigma \stackrel{?}{=} H(pk, x_1, x_2)$.

(3) The CCA security of OAEP[†] in the QRO model.

The CCA security in QRO model asks that it is hard for any quantum polynomial-time adversary $\mathcal{A}$ to tell whether the challenge ciphertext $ct^* = (c^*, \sigma^*)$ encrypts m_0 or m_1 , even if $\mathcal{A}$ can decrypt any $ct \neq ct^*$ through a decryption oracle and has quantum access to random oracles G, H .

Suppose that $ct^* = (c^*, \sigma^*)$ is generated with $x_2^* \leftarrow \psi_2, x_1^* := \text{Encode}(m_b, G(x_2^*))$, $c^* = F(x_1^*, x_2^*)$, and $\sigma^* = H(pk = F, x_1^*, x_2^*)$ for $b \leftarrow_{\$} \{0, 1\}$. We prove the CCA security of OAEP[†] based on the partial domain one-wayness (PDOW) of F over pre-image distribution $\psi = (\psi_1, \psi_2)$. Here ϵ -PDOW means that the adversary's probability of recovering x_2^* from $c^* = F(x_1^*, x_2^*)$ is bounded by ϵ for $(x_1^*, x_2^*) \leftarrow (\psi_1, \psi_2)$.

The key point in the security proof is to change the hash value of $G(x_2^*)$ to a random element. If $G(x_2^*)$ is uniform, then the ψ_1 -message hiding property of SOTP makes sure that x_1^* follows distribution ψ_1 . As a result, (x_1^*, x_2^*) follows distribution (ψ_1, ψ_2) , so the ciphertext ct^* is independent of the encrypted message m_b and thus the CCA security is proved.

The indistinguishable change of $G(x_2^*)$ to a uniform element is justified by the O2H lemma and the ϵ -PDOW of the underlying F . However, the security reduction to ϵ -PDOW of F faces two problems.

- One problem is that the reduction algorithm does not know F^{-1} so it cannot answer $\mathcal{A}$ decryption queries.
- Another problem is that the reduction algorithm does not know (x_1^*, x_2^*) so it cannot compute $\sigma^* = H(pk = F, x_1^*, x_2^*)$ for ct^* .

Below we show how to address the problems to accomplish the security proof.

(i) *Simulate the decryption oracle without knowing F^{-1} by resorting to extractable random oracle simulator.* To simulate the decryption oracle without secret key $sk = F^{-1}$, we apply the recent online-extractable technique[19], which can simulate oracle H and can also extract the pre-image (pk, x_1, x_2) from the authenticator $\sigma = H(pk, x_1, x_2)$ with the help of an extractable random oracle simulator.

Therefore, upon a decryption query $ct = (c, \sigma)$, one can extract (pk', x'_1, x'_2) from σ for those σ satisfying $\sigma = H(pk', x'_1, x'_2)$, and then check whether $pk = pk'$ and $F(x'_1, x'_2) = c$. Due to the injectivity (and also the rigidity) of underlying F , the decryption simulation (without F^{-1}) is indistinguishable to the original decryption oracle.

A subtlety lies in the combined use of the O2H lemma for H and the online-extractable technique for H . The online-extractable technique simulates the underlying hash function H with an extractable random oracle simulator, which defines the hash function on the fly, just like the lazy-sampling technique in the classic random oracle model. In contrast, the O2H lemma requires that the underlying hash function H be fixed at the beginning of the game. Therefore, the O2H lemma must be used before applying the online-extractable technique to H when getting rid of F^{-1} in the decryption oracle.

(ii) *Simulate $\sigma^* = H(pk, x_1^*, x_2^*)$ with a randomly chosen element.* According to the adaptive O2H lemma, one can change $\sigma^* = H(pk, x_1^*, x_2^*)$ to a random value, and this change indistinguishability is justified by a probability bounded by $2q\sqrt{\Pr[(x'_1, x'_2) = (x_1^*, x_2^*)]}$, where q is the number of queries and (x'_1, x'_2) is obtained by measuring one of $\mathcal{A}$'s some hash queries to H . All we have to do is to bound $\Pr[(x'_1, x'_2) = (x_1^*, x_2^*)]$. Intuitively, $(x'_1, x'_2) = (x_1^*, x_2^*)$ means the recovery of pre-image of c^* . So we expect a security reduction to ϵ -PDOW of F . To make the security reduction possible, we apply the online-extractable technique to H to simulate the decryption oracle without usage of secret key F^{-1} , just like (i), after the adaptive O2H lemma is applied to H . Next, we use the *semi-classical* O2H lemma to change $G(x_2^*)$ to a random element, so that (x_1^*, x_2^*) follows the distribution (ψ_1, ψ_2) . Meanwhile, the semi-classical O2H lemma guarantees

$$\left| \sqrt{\Pr[(x'_1, x'_2) = (x_1^*, x_2^*)]} - \sqrt{\Pr[(x'_1, x'_2) = (x_1^*, x_2^*) \wedge \neg \text{Find}]} \right| \leq 2\sqrt{(q+1)\Pr[\text{Find}]} \quad (1)$$

where event Find means that the measurement x'_2 of some hash query to G equals x_2^* . Recall that (x'_1, x'_2) is obtained by measuring some query to H . So one can additionally query $G(x'_2)$ after it obtains (x'_1, x'_2) . Then the event Find must happen if $(x'_1, x'_2) = (x_1^*, x_2^*)$, and hence $\Pr[(x'_1, x'_2) = (x_1^*, x_2^*) \wedge \neg \text{Find}] = 0$. In this way, $\Pr[(x'_1, x'_2) = (x_1^*, x_2^*)] \leq 4(q+1)\Pr[\text{Find}]$ and this gets rid of the square root. Event Find happening implies successful recovery of x_2^* from c^* , so we can bound $\Pr[\text{Find}]$ with ϵ -PDOW of F with a security reduction. This reduction is smooth since the decryption oracle does not need F^{-1} for decryption, σ^* is random, and (x_1^*, x_2^*) involved in ct^* follows distribution (ψ_1, ψ_2) . The security decay in this step is only about $\epsilon^{1/2}$, thanks to the semi-classical O2H lemma removing the square root in the right-hand of (1).

(iii) *Simulate $G(x_2^*)$ with a randomly chosen element.* Now σ^* is a randomly chosen one and independent of (x_1^*, x_2^*) , and the simulation of decryption oracle makes no use of F^{-1} . We finally are able to use the semi-classical O2H lemma to G and give a security reduction to ϵ -PDOW of F to finish the proof: apply semi-classical O2H lemma to $G(x_2^*)$, so $G(x_2^*)$ can be replaced with a random element which makes (x_1^*, x_2^*) follow distribution (ψ_1, ψ_2) . According to the O2H

lemma, the indistinguishability of changing $G(x_2^*)$ is justified by a probability bounded by $2\sqrt{(q+1) \cdot \Pr[x'_2 = x_2^*]}$, where x'_2 is the measurement of some hash query to G . Then $\Pr[x'_2 = x_2^*] \leq \epsilon$ (up to some polynomial factor) can be proved by a security reduction to ϵ -PDOW of F . This reduction again goes well since the decryption oracle does not need F^{-1} for decryption, σ^* is random, and (x_1^*, x_2^*) involved in ct^* follows distribution (ψ_1, ψ_2) . Lastly, (x_1^*, x_2^*) leaks nothing on the message m_b due to the distribution (ψ_1, ψ_2) independent of m_b , so the adversary has no advantage for guessing b and IND-CCA security follows.

Note that (ii) and (iii) are done in serial, so the whole security proof only incurs $\epsilon^{1/2}$ (up to some polynomial factor) security decay from ϵ -PDOW of F .

(iv) *The anonymity and robustness of OAEP[†]*. We show a stronger security notion called pseudo-randomness, which implies anonymity. Here pseudo-randomness (PR-CCA security) means that the ciphertext is pseudo-random under CCA attack, anonymity means the ciphertext does not leak any information about the public key pk under CCA attack. Recall that in the IND-CCA security proof, the challenge ciphertext $ct^* = (c^*, \sigma^*)$ can be indistinguishably changed to $c^* := F(x_1^*, x_2^*)$ for $x_1^* \leftarrow \psi_1$, $x_2^* \leftarrow \psi_2$, and $\sigma^* \leftarrow_{\$} \{0, 1\}^\lambda$, and the decryption oracle can be simulated with the online-extractable technique. Suppose that $F(x_1^*, x_2^*)$ has pseudo-random image for $x_1^* \leftarrow \psi_1$ and $x_2^* \leftarrow \psi_2$, which is captured by the pseudo-randomness of F , then $c^* := F(x_1^*, x_2^*)$ can be changed to a random element. Hence $ct^* = (c^*, \sigma^*)$ is pseudo-random, and this finishes the PR-CCA proof for OAEP[†]. We can also prove the robustness of OAEP[†], where robustness (ROB-CCA security) means that a ciphertext can not be valid under two different public keys. It is easy to prove ROB-CCA with collision resistance of H : If the adversary submits a ciphertext $ct = (c, \sigma)$ s.t. $\text{Decap}(sk_0, ct) \neq \perp$ and $\text{Decap}(sk_1, ct) \neq \perp$, then it must hold that $H(pk_0, x_1, x_2) = H(pk_1, x'_1, x'_2)$ and thus the adversary breaks the collision resistance of the underlying hash H .

(4) OAEP[†] as a KEM. Like OAEP and its variants, our OAEP[†] is a PKE scheme. If $H(pk, x_1, x_2) = (\sigma, k)$ additionally outputs an encapsulation key k , then our OAEP[†] also becomes KEM. Such a KEM not only encapsulates k but also encrypts a message m . Like the pseudo-randomness of (c, σ) , we can also prove k is pseudo-random, so PR-CCA security of the KEM-version of OAEP[†] follows. ROB-CCA security can be similarly proved.

2 Preliminary

Let $\emptyset$ denote the empty set. If x is defined by y or the value of y is assigned to x , we write $x := y$. For $\mu \in \mathbb{N}$, define $[\mu] := \{1, 2, \dots, \mu\}$. Denote by $x \leftarrow_{\$} \mathcal{X}$ the procedure of sampling x from set $\mathcal{X}$ uniformly at random. Let $|\mathcal{X}|$ denote the number of elements in set $\mathcal{X}$. All our algorithms are probabilistic unless stated otherwise. PPT abbreviates probabilistic polynomial time. We use $y \leftarrow \mathcal{A}(x)$ to define the random variable y obtained by executing algorithm $\mathcal{A}$ on input x . We use $y \in \mathcal{A}(x)$ to indicate that y lies in the support of $\mathcal{A}(x)$. We also use $y \leftarrow \mathcal{A}(x; r)$ to make explicit the random coins r used in the probabilistic computation. Let λ denote the security parameter.

Definition 1 (Collision Entropy). For a distribution ψ , we define its collision entropy as $-\log_2 \Pr[x = x' : x, x' \leftarrow \psi]$.

2.1 (General) KEM and Its Security Notions

A key encapsulated mechanism (KEM) consists of three algorithms $\text{KEM} = (\text{KeyGen}, \text{Encap}, \text{Decap})$. Here we present a *general* formalization for KEM, which encompasses KEM with *partial message recovery* [17] so that **Encap** is also able to encrypt a message m in the encapsulation ct besides encapsulating a key k . Accordingly, **Decap** can not only recover the key k but also the message m . Below is the syntax for the general KEM.

- **KeyGen**: It outputs a key pair (pk, sk) .
- **Encap** $(pk, m \in \{0, 1\}^\tau)$: It takes as input a public key pk and a (partial) message m , and outputs a ciphertext ct and an encapsulated key k .
- **Decap** (sk, ct) : It takes as input a secret key sk and a ciphertext ct , and outputs a (partial) message m and an encapsulated key k .

The correctness of KEM requires that $\text{Decap}(sk, ct) = (m, k)$, where $(pk, sk) \leftarrow \text{KeyGen}$ and $(ct, k) \leftarrow \text{Encap}(pk, m)$.

If m is null in **Encap**, then this general KEM reduces to the classic KEM. If k is null in **Encap**, then this general KEM reduces to the classic PKE.

To encrypt a longer message, say $m = m_1 | m_2 \in \{0, 1\}^\lambda$ with a τ -bit m_1 and $(\lambda - \tau)$ -bit m_2 , we can resort to a new way of $\text{PKE} = \text{KEM} + \text{DEM}$, where KEM encrypts m_1 and encapsulates k and DEM uses k to encrypt m_2 . Clearly, this approach saves at least τ -bit ciphertext compared to the classic KEM+DEM.

Below we define for KEM its CCA security, pseudo-randomness, anonymity and robustness under chosen-ciphertext attacks. These security notions are denoted by IND-CCA, PR-CCA, ANO-CCA and ROB-CCA respectively.

Definition 2 (Security Notions for KEM). For any adversary $\mathcal{A}$ and attack $\text{ATK} \in \{\text{IND}, \text{PR}, \text{ANO}\}$ and ROB-CCA, the advantage function against ATK-CCA security for KEM is defined as

$$\text{Adv}_{\text{KEM}}^{\text{ATK-CCA}}(\mathcal{A}) := \left| \Pr \left[\text{Exp}_{\text{KEM}}^{\text{ATK-CCA}} \Rightarrow 1 \right] - \frac{1}{2} \right|, \quad \text{Adv}_{\text{KEM}}^{\text{ROB-CCA}}(\mathcal{A}) := \Pr \left[\text{Exp}_{\text{KEM}}^{\text{ROB-CCA}} \Rightarrow 1 \right],$$

where $\text{Exp}_{\text{KEM}}^{\text{IND-CCA}}$, $\text{Exp}_{\text{KEM}}^{\text{PR-CCA}}$, $\text{Exp}_{\text{KEM}}^{\text{ANO-CCA}}$, and $\text{Exp}_{\text{KEM}}^{\text{ROB-CCA}}$ are given in Fig. 3.

2.2 Lemmas for Quantum Random Oracles

In the quantum world, a quantum algorithm $\mathcal{A}$ can perform superposition queries to the random oracle H , and then oracle H behaves as a unitary operation $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$. In this case, H becomes a quantum random oracle.

We recall the online-extractable technique [19]. First, we define $\Gamma(f)$ and $\Gamma'(f)$ which are associated with a function f .

$\text{Exp}_{\text{KEM}}^{\text{IND-CCA}}:$ $b \leftarrow_{\$} \{0, 1\}$ $(pk, sk) \leftarrow \text{KeyGen}$ $(m_0, m_1, st) \leftarrow \mathcal{A}^{\text{DECAP}_{\perp}(\cdot)}(pk)$ $(ct^*, k^*) \leftarrow \text{Encap}(pk, m_b)$ $k_0^* := k^*, k_1^* \leftarrow_{\$} \mathcal{K}$ $b' \leftarrow \mathcal{A}^{\text{DECAP}_{ct^*}(sk, \cdot)}(st, ct^*, k_b^*)$ Return $b = b'$	$\text{Exp}_{\text{KEM}}^{\text{PR-CCA}}:$ $b \leftarrow_{\$} \{0, 1\}$ $(pk, sk) \leftarrow \text{KeyGen}$ $(m, st) \leftarrow \mathcal{A}^{\text{DECAP}_{\perp}(\cdot)}(pk)$ $(ct_0^*, k_0^*) \leftarrow \text{Encap}(pk, m)$ $ct_1^* \leftarrow_{\$} \mathcal{CT}, k_1^* \leftarrow_{\$} \mathcal{K}$ $b' \leftarrow \mathcal{A}^{\text{DECAP}_{ct_b^*}(sk, \cdot)}(st, ct_b^*, k_b^*)$ Return $b = b'$	$\text{DECAP}_{ct^*}(ct):$ If $ct = ct^*$: Return $\perp$ $(m, k) \leftarrow \text{Decap}(sk, ct)$ Return (m, k)
$\text{Exp}_{\text{KEM}}^{\text{ANO-CCA}}:$ $b \leftarrow_{\$} \{0, 1\}$ $(pk_0, sk_0) \leftarrow \text{KeyGen}$ $(pk_1, sk_1) \leftarrow \text{KeyGen}$ $(m, st) \leftarrow \mathcal{A}^{\text{DECAP}'_{\perp}(\cdot, \cdot)}(pk_0, pk_1)$ $(ct^*, k^*) \leftarrow \text{Encap}(pk_b, m)$ $b' \leftarrow \mathcal{A}^{\text{DECAP}'_{ct^*}(\cdot, \cdot)}(st, ct^*, k^*)$ Return $b = b'$	$\text{Exp}_{\text{KEM}}^{\text{ROB-CCA}}:$ $(pk_0, sk_0) \leftarrow \text{KeyGen}$ $(pk_1, sk_1) \leftarrow \text{KeyGen}$ $ct \leftarrow \mathcal{A}^{\text{DECAP}'_{\perp}(\cdot, \cdot)}(pk_0, pk_1)$ $(m_0, k_0) \leftarrow \text{Decap}(sk_0, ct)$ $(m_1, k_1) \leftarrow \text{Decap}(sk_1, ct)$ Return $(m_0, k_0) \neq \perp \wedge$ $(m_1, k_1) \neq \perp$	$\text{DECAP}'_{ct^*}(\beta, ct):$ If $ct = ct^*$: Return $\perp$ If $\beta \notin \{0, 1\}$: Return $\perp$ $(m, k) \leftarrow \text{Decap}(sk_{\beta}, ct)$ Return (m, k)

Fig. 3: Experiments $\text{Exp}_{\text{KEM}}^{\text{IND-CCA}}$, $\text{Exp}_{\text{KEM}}^{\text{PR-CCA}}$, $\text{Exp}_{\text{KEM}}^{\text{ANO-CCA}}$, and $\text{Exp}_{\text{KEM}}^{\text{ROB-CCA}}$.

Definition 3 ([19]). Let $f : \mathcal{X} \times \{0, 1\}^n \rightarrow \mathcal{C}$ be an arbitrary fixed function. Define $\Gamma(f) := \max_{x, c} |\{y \mid f(x, y) = c\}|$, $\Gamma'(f) := \max_{x \neq x', y'} |\{y \mid f(x, y) = f(x', y')\}|$.

Roughly speaking, the online-extractable technique in [19] suggests that a random oracle $H(\cdot)$ can be successfully simulated and the pre-image x can be successfully extracted from the image $c = f(x, H(x))$ by an efficient extractable random oracle simulator $\text{eRO} = (\text{eRO.RO}, \text{eRO.E})$. The following theorem makes the statement formal.

Theorem 1 (Theorem 4.3 in [19]). Let H be a random oracle, and $\Gamma(\cdot)$ and $\Gamma'(\cdot)$ be defined as per Def. 3. There exists an efficient extractable random oracle simulator $\text{eRO} = (\text{eRO.RO}, \text{eRO.E})$ satisfying the following properties.

- (1) If eRO.E is unused, eRO.RO is perfectly indistinguishable from the random oracle $H : \mathcal{X} \rightarrow \{0, 1\}^n$.
- (2.a) Any two subsequent independent queries to eRO.RO commute. We refer two subsequent queries as being independent if the input to one query does not depend on the output of the other.
- (2.b) Any two subsequent independent queries to eRO.E commute.
- (2.c) Any two subsequent independent queries to eRO.E and eRO.RO are $8\sqrt{2\Gamma(f)/2^n}$ -almost-commute.
- (3) (Proposition 4.5 in [19].) Consider a query algorithm $\mathcal{A}$ that makes q queries to eRO.RO but no query to eRO.E , outputting some $(c_1, \dots, c_\ell) \in \mathcal{C}^\ell$ and $(x_1, \dots, x_\ell) \in \mathcal{X}^\ell$. Suppose that $(y_1, \dots, y_\ell)$ is obtained by making ℓ additional queries to eRO.RO on input $(x_1, \dots, x_\ell)$, and $(\hat{x}_1, \dots, \hat{x}_\ell)$ obtained by

making ℓ additional queries to $\text{eRO.E}(f, c_i)$ for each $i \in [\ell]$. Then

$$\Pr_{\substack{(c_1, x_1, \dots, c_\ell, x_\ell) \leftarrow \mathcal{A}^{\text{eRO.RO}} \\ y_i \leftarrow \text{eRO.RO}(x_i), \hat{x}_i \leftarrow \text{eRO.E}(f, c_i)}} [\exists i : \hat{x}_i \neq x_i \wedge f(x_i, y_i) = c_i] \leq \frac{300(q + \ell + 1)^3 \Gamma'(f) + 2}{2^n}. \quad (2)$$

Furthermore, according to (2.c), the above ℓ queries to eRO.RO and the ℓ independent queries to eRO.E can be arbitrarily interleaved, at the cost of an extra additive term $8\ell^2 \sqrt{\frac{2\Gamma(f)}{2^n}}$ added to the right-hand side of (2).

Next, let us recall the semi-classical oracle and the punctured oracle, and then recall the semi-classical O2H lemma proposed in [9].

Definition 4 (Semi-Classical Oracle and Punctured Oracle). Let $f_S(x)$ be an indicator function for a set $\mathcal{S}$, which outputs 1 if $x \in \mathcal{S}$ and 0 otherwise. A semi-classical oracle $\mathcal{O}_S^{SC}$ for a set $\mathcal{S}$ is defined as follows: on input $|x\rangle$, it computes and measures $f_S(x)$ in an auxiliary qubit $|f_S(x)\rangle$ indicating whether $x \in \mathcal{S}$ or not. For an oracle H , the punctured oracle $H \setminus \mathcal{S}$ is oracle H punctured on $\mathcal{S}$, which upon an input x first queries $\mathcal{O}_S^{SC}(x)$ and then queries $H(x)$.

Lemma 1 (Semi-Classical O2H Lemma, Part of Theorem 1 in [9]). Let $\mathcal{S}$ be a random subset of $\mathcal{X}$. Let $G, H : \mathcal{X} \rightarrow \mathcal{Y}$ be random oracles satisfying $G(x) = H(x)$ for all $x \notin \mathcal{S}$. Let z be a random bit-string such that $\mathcal{S}, G, H, z$ may have arbitrary joint distribution. Let $\mathcal{A}^{\mathcal{O}}$ be a quantum oracle algorithm which makes at most q queries to oracle $\mathcal{O}$. In the execution of $\mathcal{A}^{G \setminus \mathcal{S}}$, let Find be the event that any of measurements of $|f_S(x)\rangle$ returns 1. Define

$$\begin{aligned} P_{\text{left}} &:= \Pr [b = 1 : b \leftarrow \mathcal{A}^H(z)], \\ P_{\text{right}} &:= \Pr [b = 1 \wedge \neg \text{Find} : b \leftarrow \mathcal{A}^{G \setminus \mathcal{S}}(z)], \quad P'_{\text{right}} := \Pr [b = 1 : b \leftarrow \mathcal{A}^G(z)], \\ P_{\text{find}} &:= \Pr [\text{Find} : \mathcal{A}^{G \setminus \mathcal{S}}(z)]. \end{aligned}$$

$$\begin{aligned} \text{Then } |P_{\text{left}} - P_{\text{right}}| &\leq 2\sqrt{(q+1) \cdot P_{\text{find}}}, \quad |P_{\text{left}} - P'_{\text{right}}| \leq 2\sqrt{(q+1) \cdot P_{\text{find}}} \\ \text{and } |\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}| &\leq 2\sqrt{(q+1) \cdot P_{\text{find}}}, \quad |\sqrt{P_{\text{left}}} - \sqrt{P'_{\text{right}}}| \leq 2\sqrt{(q+1) \cdot P_{\text{find}}}. \end{aligned}$$

Lemma 2 (Search in Semi-Classical Oracle, Theorem 2 in [9]). Let $\mathcal{O}_S^{SC}$ be a semi-classical oracle with domain $\mathcal{X}$. Then $\mathcal{O}_\emptyset^{SC}$ is just a quantum oracle. Let $\mathcal{S} \subseteq \mathcal{X}$ and $z \in \{0, 1\}^*$ such that $\mathcal{S}, z$ may have arbitrary joint distribution. Let $\mathcal{A}^{\mathcal{O}_S^{SC}}(z)$ be any quantum oracle algorithm making at most q queries to $\mathcal{O}_S^{SC}$.

Let $\mathcal{B}$ be an algorithm that, on input z , chooses $i \leftarrow_{\$} [q]$, runs $\mathcal{A}^{\mathcal{O}_\emptyset^{SC}}(z)$ until (just before) the i -th query, and then measures all query input registers in the computational basis and outputs the set $\mathcal{T}$ of measurement outcomes. Then

$$\Pr [\text{Find} : \mathcal{A}^{\mathcal{O}_S^{SC}}(z)] \leq 4q \cdot \Pr [\mathcal{S} \cap \mathcal{T} \neq \emptyset : \mathcal{T} \leftarrow \mathcal{B}(z)].$$

Next we recall the adaptive version of O2H lemma. Compared to the semi-classical O2H lemma where the punctured set S is fixed beforehand, the quantum adversary may have interactions with a classical algorithm $\mathcal{C}$, so it can adaptively choose a punctured point $\bar{x}$ based on what it has learned from the interactions. This adaptive O2H lemma was originally proposed for the design of non-interactive zero-knowledge proof and signature schemes in QROM [41].

Lemma 3 (Adaptive O2H Lemma, Lemma 9 in [41]). *Let $H : \{0, 1\}^* \rightarrow \{0, 1\}^n$ be a random oracle. Consider the following algorithms:*

- The oracle algorithm $\mathcal{C}_0^H$, that makes at most q_0 (quantum) queries to H .
- The classical algorithm $\bar{\mathcal{C}}$, that may access the classical part of the final state of $\mathcal{C}_0^H$, and has output of collision entropy at least k for every initial state.
- The oracle algorithm $\mathcal{C}_1^H$, that may access the final states of $\mathcal{C}_0^H$ and $\bar{\mathcal{C}}$ and makes at most q_1 (quantum) queries to H .
- Let $\mathcal{C}'^H$ be an oracle algorithm, that on input $(st, j, B, \bar{x})$ does the following: run $\mathcal{C}_1^H(st, \bar{x}, B)$ until (just before) the j -th query, measure the argument of the query in the computational basis, output the measurement outcome. (When $\mathcal{C}_1^H(st, \bar{x}, B)$ makes less than j queries, $\mathcal{C}'^H$ outputs $\perp$.)

$$\begin{aligned}
P_A^1 &:= \Pr [\beta = 1 : (\bar{m}, st) \leftarrow \mathcal{C}_0^H, \bar{x} \leftarrow \bar{\mathcal{C}}(\bar{m}), \beta \leftarrow \mathcal{C}_1^H(st, \bar{x}, H(\bar{x}))], \\
P_A^2 &:= \Pr [\beta = 1 : (\bar{m}, st) \leftarrow \mathcal{C}_0^H, \bar{x} \leftarrow \bar{\mathcal{C}}(\bar{m}), B \leftarrow_{\$} \{0, 1\}^n, \beta \leftarrow \mathcal{C}_1^H(st, \bar{x}, B)], \\
P_C &:= \Pr [\bar{x} = \bar{x}' : (\bar{m}, st) \leftarrow \mathcal{C}_0^H, \bar{x} \leftarrow_{\$} \bar{\mathcal{C}}(\bar{m}), B \leftarrow_{\$} \{0, 1\}^n, j \leftarrow_{\$} [q_1], \bar{x}' \leftarrow \mathcal{C}'^H(st, j, B, \bar{x})].
\end{aligned}$$

Then $|P_A^1 - P_A^2| \leq 2q_1\sqrt{P_C} + (4 + \sqrt{2})\sqrt{q_0}2^{-k/4}$.

Next, we recall the bound on quantum collision finding.

Lemma 4 (Theorem 8 in [43]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite sets. Let $H : \mathcal{X} \rightarrow \mathcal{Y}$ be a random oracle. If an unbounded-time quantum adversary $\mathcal{A}$ makes a (quantum) query to H at most q times, then we have*

$$\Pr [H(x_0) = H(x_1) \wedge x_0 \neq x_1 : (x_0, x_1) \leftarrow \mathcal{A}^H] \leq \frac{648(q+1)^3}{|\mathcal{Y}|}.$$

3 The OAEP[†] Transform

In Subsect. 3.1, we introduce the concepts of almost trapdoor injective function (ATIF) and semi-generalized one-time pad (SOTP) and define useful properties for them. In Subsect. 3.2, we present our OAEP[†] constructed from ATIF, SOTP and two hash functions. We also prove a variety of security notions for OAEP[†].

3.1 ATIF and SOTP

We define almost trapdoor injective function (ATIF) family and its injectivity error and its properties like rigidity, partial domain one-wayness (PDOW), pseudo-randomness and collision resistance. We also introduce a primitive called semi-generalized one-time pad (SOTP) and its property of message hiding.

Definition 5 (ATIF). *An almost trapdoor injective function (ATIF) family $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ consists of the following three algorithms:*

- Gen : *It outputs (descriptions of) two algorithms F and F^{-1} .*
- $F(x)$: *It is a deterministic algorithm, which takes as input x and outputs the evaluated value $y := F(x)$.*
- $F^{-1}(y)$: *It is a deterministic algorithm which takes as input y and outputs a pre-image x or a special symbol $\perp$.*

In addition, we say an ATIF family $\mathcal{F}$ has η -injectivity error, denoted by $\eta\text{-ATIF}$, if $\Pr [\exists x \in \mathcal{X}, F^{-1}(F(x)) \neq x : (F, F^{-1}) \leftarrow \text{Gen}] \leq \eta$.

Definition 6 (Rigidity of ATIF [15]). *An ATIF family $\mathcal{F}$ has rigidity if for all $(F, F^{-1}) \leftarrow \text{Gen}$ and all $y \in \mathcal{Y}$ with $x := F^{-1}(y) \neq \perp$, it holds that $F(x) = y$.*

Definition 7 (Partial Domain One-Wayness of ATIF w.r.t. Distribution ψ). *Let $\mathcal{F} : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{Y}$ be an ATIF family, and $\psi = (\psi_1, \psi_2)$ be a distribution over $\mathcal{X}_1 \times \mathcal{X}_2$ such that ψ_1 and ψ_2 are independent. The advantage function of an adversary $\mathcal{A}$ against partial domain one-wayness (PDOW) of $\mathcal{F}$ w.r.t. distribution ψ is defined by*

$$\text{Adv}_{\mathcal{F}, \psi}^{\text{PD-OW}}(\mathcal{A}) := \Pr \left[x'_2 = x_2^* : \begin{array}{l} (F, F^{-1}) \leftarrow \text{Gen}, (x_1^*, x_2^*) \leftarrow \psi \\ c^* := F(x_1^*, x_2^*), x'_2 \leftarrow \mathcal{A}(F, c^*) \end{array} \right].$$

Similarly, we define $\text{Adv}_{F, F^{-1}, \psi}^{\text{PD-OW}}(\mathcal{A})$ as the advantage of $\mathcal{A}$ against partial domain one-wayness for a fixed pair (F, F^{-1}) . Then $\text{Adv}_{\mathcal{F}, \psi}^{\text{PD-OW}}(\mathcal{A}) = \mathbb{E}_{F, F^{-1}} [\text{Adv}_{F, F^{-1}, \psi}^{\text{PD-OW}}(\mathcal{A})]$.

Definition 8 (Pseudo-Randomness of ATIF w.r.t. Distribution ψ). *Let $\mathcal{F} : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{Y}$ be an almost injective function (ATIF) family, and ψ be a distribution over $\mathcal{X}_1 \times \mathcal{X}_2$. The advantage function of an adversary $\mathcal{A}$ against pseudo-randomness of $\mathcal{F}$ w.r.t. distribution ψ is defined by*

$$\text{Adv}_{\mathcal{F}, \psi}^{\text{PR}}(\mathcal{A}) := \left| \Pr [b = b' : b' \leftarrow \mathcal{A}(F, c_b)] - \frac{1}{2} \right|,$$

where $(F, F^{-1}) \leftarrow \text{Gen}$, $b \leftarrow_{\$} \{0, 1\}$, $(x_1^*, x_2^*) \leftarrow \psi$, $c_0 := F(x_1^*, x_2^*)$, and $c_1 \leftarrow_{\$} \mathcal{Y}$.

Definition 9 (Collision Resistance of ATIF). *Let $\mathcal{F}$ be an almost injective function (ATIF) family. $\mathcal{F}$ has collision resistance if*

$$\Pr [F_0 = F_1 : (F_0, F_0^{-1}) \leftarrow \text{Gen}, (F_1, F_1^{-1}) \leftarrow \text{Gen}] = \text{negl}(\lambda).$$

Next, we present the primitive called Semi-Generalized One-Time Pad (SOTP), which was introduced in [31]. SOTP can be regarded as a generalized one-time pad. It uses Encode and Decode algorithms, rather than the simple “ $\oplus$ ” over binary strings in one-time pad. More importantly, SOTP requires that different codewords encoded from different plaintexts share the same distribution ψ' , rather than the uniform distribution as asked by the one-time pad.

Definition 10 (Semi-Generalized One-Time Pad [31]). *A semi-generalized one-time pad $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{X}, \mathcal{U}, \mathcal{Y}, \psi')$ with message space $\mathcal{X}$, randomness space $\mathcal{U}$, and a code space $\mathcal{Y}$ with corresponding distribution ψ' consists of the following two algorithms and satisfies correctness and ψ' -message hiding.*

- **Encode**(x, r): *It is a deterministic algorithm which takes as input $(x, r) \in \mathcal{X} \times \mathcal{U}$ and outputs a codeword $y \in \mathcal{Y}$.*
- **Decode**(y, r): *It is a deterministic algorithm which takes as input $(y, r) \in \mathcal{Y} \times \mathcal{U}$ and outputs $x \in \mathcal{X}$.*

A SOTP scheme should satisfy the following properties:

Correctness of SOTP. $\forall (x, r) \in \mathcal{X} \times \mathcal{U}$, we have $\text{Decode}(\text{Encode}(x, r), r) = x$.

ψ' -Message Hiding of SOTP. For a fixed $x \in \mathcal{X}$, suppose that the random variable $y \leftarrow \text{Encode}(x, r)$ for $r \leftarrow \mathcal{U}$ follows distribution ψ_x over $\mathcal{Y}$. If $\psi_x = \psi'$ for all $x \in \mathcal{X}$, then SOTP has the property of ψ' -message hiding.

3.2 OAEP[†]: KEM from ATIF and SOTP

Now we present our OAEP[†] transform which converts an ATIF family $\mathcal{F}$ to a CCA-secure KEM with the help of SOTP and two hash functions G, H . We show that the resulting KEM enjoys the property of partial message recovery and has IND-CCA, PR-CCA, ANO-CCA and ROB-CCA security in QROM.

Below are the building blocks needed in our OAEP[†] transform.

- An ATIF family $\mathcal{F} : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathcal{Y}$.
- A SOTP scheme $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{M}, \mathcal{U}, \mathcal{X}_1, \psi_1)$
- Two hash functions $G : \mathcal{X}_2 \rightarrow \mathcal{U}$ and $H : \{0, 1\}^* \rightarrow \{0, 1\}^{2\lambda}$.
- Let ψ_2 be a distribution on $\mathcal{X}_2$ with collision entropy κ .

The generic construction of KEM = (KeyGen, Encap, Decap) via our OAEP[†] transform is shown in Fig. 4. Clearly, KEM supports partial message recovery.

Theorem 2. *Suppose that $\mathcal{F}$ is an η -ATIF family with rigidity and partial domain one-wayness w.r.t. $\psi = (\psi_1, \psi_2)$, $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{M}, \mathcal{U}, \mathcal{X}_1, \psi_1)$ is a semi-generalized one-time pad scheme, and ψ_2 is a distribution with collision entropy κ . Then scheme KEM from OAEP[†] in Fig. 4 has IND-CCA security. More precisely, for any adversary $\mathcal{A}$ issuing q_G and q_H (quantum) random oracle queries to G and H , and q_d decryption queries, if $q = q_G + q_H + q_d + 2$, then*

$$\text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(\mathcal{A}) \leq 16q^2 \sqrt{\text{Adv}_{\mathcal{F}, \psi}^{\text{PD-OW}}(\mathcal{B})} + 4q \sqrt{\frac{300q^3}{2^\lambda} + \frac{16\sqrt{2}q^2}{2^{\lambda/2}}} + \frac{6\sqrt{q}}{2^{\kappa/4}} + \eta.$$

KeyGen :	Encap($pk = F, m$) :	Decap($sk = F^{-1}, ct = (c, \sigma)$)
$(F, F^{-1}) \leftarrow \text{Gen}$	$x_2 \leftarrow \psi_2$	$(x_1, x_2) := F^{-1}(c)$
$pk := F$	$x_1 := \text{Encode}(m, G(x_2))$	If $(x_1, x_2) = \perp$: Return $\perp$
$sk := F^{-1}$	$c := F(x_1, x_2)$	$(\sigma', k) := H(pk, x_1, x_2)$
Return (pk, sk)	$(\sigma, k) := H(pk, x_1, x_2)$	If $\sigma' \neq \sigma$: Return $\perp$
	Return $(ct = (c, \sigma), k)$	$m := \text{Decode}(x_1, G(x_2))$
		Return (m, k)

Fig. 4: Scheme KEM constructed from ATIF family $\mathcal{F}$ and SOTP via OAEP[†].

Proof. We now consider a sequence of games for a *fixed* key pair (F, F^{-1}) and analyze $\mathcal{A}$'s advantages in these games. Define $G_i \Rightarrow 1$ as the event $b = b'$ in G_i .

Let $\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A})$ be $\mathcal{A}$'s advantage for the key pair (F, F^{-1}) . We know that $\text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(\mathcal{A}) = \mathbb{E}_{F, F^{-1}} [\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A})]$. Note that $\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A}) \leq 1$ for all $(F, F^{-1}) \leftarrow \text{Gen}$. Due to the η -injective error of $\mathcal{F}$, we have

$$\text{Adv}_{\text{KEM}}^{\text{IND-CCA}}(\mathcal{A}) \leq \mathbb{E}_{\text{injective } F, F^{-1}} [\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A})] + \eta.$$

In the following games we focus on $\mathcal{A}$'s advantage on those injective pairs (F, F^{-1}) , which satisfies $F^{-1}(F(x_1, x_2)) = (x_1, x_2)$ for all $(x_1, x_2) \in \mathcal{X}_1 \times \mathcal{X}_2$.

The proof of Theorem 2 proceeds via games G_0 - G_6 and showing $|\Pr[G_i \Rightarrow 1] - \Pr[G_{i-1} \Rightarrow 1]| \leq \text{negl}(\lambda)$ for $i \in [6]$. The pseudo-codes of G_0 - G_6 are shown in Fig. 5. To prove $|\Pr[G_3 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]| \leq \text{negl}(\lambda)$, we further consider games from $G_3 = G_{3.1}$ to $G_{3.8}$ whose pseudo-codes are shown in Fig. 6.

Game G_0 . This is the original IND-CCA experiment for a fixed (F, F^{-1}) , denoted by $\text{Exp}_{\text{KEM}}^{\text{IND-CCA}}(F, F^{-1})$, except that the challenger $\mathcal{C}$ chooses $x_2^* \leftarrow \psi_2$ at the beginning of the game. This is a conceptual change, so we have

$$\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A}) = \left| \Pr[\text{Exp}_{\text{KEM}}^{\text{IND-CCA}}(F, F^{-1}) \Rightarrow 1] - 1/2 \right| = |\Pr[G_0 \Rightarrow 1] - 1/2|.$$

Game G_1 . G_1 is the same as G_0 , except that the challenger $\mathcal{C}$ adds a new rejection rule in DECAP oracle after $\mathcal{A}$ obtains the challenge ciphertext $ct^* = (c^*, \sigma^*)$. In G_1 , the new rejection rule in DECAP is as follows.

- **Rejection rule:** DECAP($ct = (c, \sigma)$) will return $\perp$ if $c = c^*$.

G_1 and G_0 differs only when $\mathcal{A}$ makes a decryption query $ct = (c, \sigma)$ such that $c = c^*$ and $\sigma \neq \sigma^*$, but DECAP($ct = (c^*, \sigma)$) in G_0 does not return $\perp$. This is impossible due to the injectivity of F , since $(x_1^*, x_2^*) = F^{-1}(c^*) = F^{-1}(c) = (x_1, x_2)$ implies $(\sigma, k) = H(pk, x_1, x_2) = H(pk, x_1^*, x_2^*) = (\sigma^*, k^*)$. So we have

$$\Pr[G_1 \Rightarrow 1] = \Pr[G_0 \Rightarrow 1].$$

Game G_2 . G_2 is the same as G_1 , except that the challenger $\mathcal{C}$ changes the check rule in DECAP oracle. In G_2 , for $\mathcal{A}$'s decryption query DECAP($ct = (c, \sigma)$), after

$\mathcal{C}$ computes $(x_1, x_2) := F^{-1}(c)$, $\mathcal{C}$ will check whether $c = F(x_1, x_2)$ rather than check whether $(x_1, x_2) = \perp$ as did in G_1 . Note that if $(x_1, x_2) = \perp$, it must hold $c \neq F(x_1, x_2)$. So G_2 and G_1 differ only when $(x_1, x_2) := F^{-1}(c) \neq \perp$ but $F(x_1, x_2) \neq c$. By the rigidity of the underlying F , this event does not happen. So we have $\Pr[G_2 \Rightarrow 1] = \Pr[G_1 \Rightarrow 1]$.

Game G_3 . G_3 is the same as G_2 , except for the generation of (σ^*, k^*) in the challenge phase. In G_3 , after $\mathcal{A}$ submits (m_0, m_1) , $\mathcal{C}$ samples $(\sigma^*, k^*) \leftarrow_{\$} \{0, 1\}^{2\lambda}$, rather than computes $(\sigma^*, k^*) := H(pk, x_1^*, x_2^*)$ as in G_2 .

Now we bound the distinguishing advantage of G_3 and G_2 using the adaptive O2H lemma (cf. Lemma 3).

Let $\mathcal{C}_0^H$ be the challenger algorithm $\mathcal{C}$ in the pre-challenge phase in G_2/G_3 . That is, $\mathcal{C}_0^H$ samples $x_2^* \leftarrow \psi_2$, samples $b \leftarrow \{0, 1\}$, invokes $(m_0, m_1, st) \leftarrow \mathcal{A}(pk = F)$, and returns $\bar{m} := (x_2^*, m_b)$. Let $\bar{\mathcal{C}}$ be a part of the challenger algorithm in the challenge phase. That is, $\bar{\mathcal{C}}(x_2^*, m_b)$ computes $x_1^* = \text{Encode}(m_b, G(x_2^*))$, $c^* = F(x_1^*, x_2^*)$ and outputs $\bar{x} := (pk, x_1^*, x_2^*)$. Define $\mathcal{C}_1^H(st, \bar{x} = (pk, x_1^*, x_2^*), (\sigma^*, k^*) = H(pk, x_1^*, x_2^*)$ or $(\sigma^*, k^*) \leftarrow_{\$} \{0, 1\}^{2\lambda}$ as the rest of $\mathcal{C}$ in G_2/G_3 : $\mathcal{C}_1^H$ takes (σ^*, k^*) as input, obtains $ct^* = (c^*, \sigma^*)$ and $k_0^* := k^*$, and then $\mathcal{C}_1^H$ behaves like the part of $\mathcal{C}$ in the after-challenge phase in G_2 . That is, $\mathcal{C}_1^H$ invokes $b' \leftarrow \mathcal{A}(ct^* = (c^*, \sigma^*), k_b^*)$ and outputs $\beta := 1$ if $b = b'$ and $\beta := 0$ if $b \neq b'$.

It is easy to see that $(\mathcal{C}_0^H, \bar{\mathcal{C}}, \mathcal{C}_1^H)$ is just G_2 if $(\sigma^*, k^*) = H(pk, x_1^*, x_2^*)$ and it is just G_3 if $(\sigma^*, k^*) \leftarrow_{\$} \{0, 1\}^{2\lambda}$. Here $\mathcal{C}_0^H, \mathcal{C}_1^H$ use oracle H to answer all the H queries from $\mathcal{A}$ and the decryption oracle. Suppose $\mathcal{A}$ makes q'_H (so $q'_H \leq q_H$) queries to H after it obtains ct^* . Let $P_A^1 := \Pr[G_2 \Rightarrow 1]$ and $P_A^2 := \Pr[G_3 \Rightarrow 1]$. By Lemma 3, we have $|\Pr[G_3 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]|$

$$\leq 2(q'_H + q_d) \sqrt{\Pr[(\hat{x}_1, \hat{x}_2) = (x_1^*, x_2^*) : (\hat{x}_1, \hat{x}_2) \leftarrow \mathcal{C}' \text{ in } G_3]} + \frac{6\sqrt{q_H + q_d}}{2^{\kappa/4}}, \quad (3)$$

where $\mathcal{C}'$ behaves just like $\mathcal{C}$ in G_3 but measures $\mathcal{A}$'s j -th (post-challenge) query to H after $\mathcal{A}$ obtains (ct^*, k^*) and outputs the measurement outcome for $j \leftarrow_{\$} [q'_H]$.

Note that beyond $\mathcal{A}$'s direct hash queries to H , there are also DECAP oracle's queries to $H(pk, x_1, x_2)$ resulting from $\mathcal{A}$'s $\text{DECAP}(c, \sigma)$ queries and the intermediate values $(x_1, x_2) \leftarrow F^{-1}(c)$ from $\text{DECAP}(ct = (c, \sigma))$. However, hash query to $H(pk, x_1^*, x_2^*)$ never occurs in DECAP oracle in G_3 , since $\text{DECAP}(ct = (c, \sigma))$ returns $\perp$ if $c = c^*$ (which is equivalent to $(x_1, x_2) = (x_1^*, x_2^*)$ due to the injectivity of F). Therefore, $\mathcal{C}'$ has at most $q'_H + q_d$ hash queries to H , and hence (3) follows smoothly from Lemma 3.

Define Ext as the event that $(\hat{x}_1, \hat{x}_2) = (x_1^*, x_2^*)$. To bound $\Pr[\text{Ext in } G_3]$, we further hybrid G_3 from $G_{3.1}$ to $G_{3.8}$. We will show $|\Pr[\text{Ext in } G_{3.i}] - \Pr[\text{Ext in } G_{3.(i-1)}]| = \text{negl}(\lambda)$ and $\Pr[\text{Ext in } G_{3.8}] = \text{negl}(\lambda)$. Games $G_{3.1}$ - $G_{3.8}$ are shown in Fig. 6.

Game $G_{3.1}$. $G_{3.1}$ is the same as G_3 , except that $\mathcal{C}$ samples $j \leftarrow_{\$} [q'_H + q_d]$ and measures $\mathcal{C}$'s j -th post-challenge query to H to obtain $(\hat{x}_1, \hat{x}_2)$. It is easy to see that $\mathcal{C}$ in $G_{3.1}$ is just $\mathcal{C}'$ in G_3 . Therefore, $G_{3.1}$ is identical to G_3 . So (3) becomes

$$|\Pr[G_3 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]| \leq 2(q'_H + q_d) \cdot \sqrt{\Pr[\text{Ext in } G_{3.1}]} + \frac{6\sqrt{q_H + q_d}}{2^{\kappa/4}}.$$

Game $G_{3.2}$. $G_{3.2}$ is the same as $G_{3.1}$, except that challenger $\mathcal{C}$ uses extractable random oracle simulator $\text{eRO} = (\text{eRO.RO}, \text{eRO.E})$ to simulate random oracle H . More precisely, $\mathcal{C}$ uses eRO.RO to simulate H both for the hash evaluations in the DECAP oracle and for $\mathcal{A}$'s hash query to H . $\mathcal{C}$ also maintains a list $\mathcal{L}$ to store each tuple (x_1, x_2, σ) , where $(x_1, x_2) := F^{-1}(c)$ is generated in oracle $\text{DECAP}(ct = (c, \sigma))$. At the end of the game, $\mathcal{C}$ extracts (pk', x'_1, x'_2) by invoking $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$ for each $(x_1, x_2, \sigma) \in \mathcal{L}$ where $f((pk, x_1, x_2), (\sigma, k)) := \sigma$. Since the extraction is done at the end of the game, by Theorem 1 (1), we have

$$\Pr[\text{Ext in } G_{3.2}] = \Pr[\text{Ext in } G_{3.1}].$$

Meanwhile, we define the following event **Bad** in $G_{3.2}$:

$$\text{Bad} : \exists (x_1, x_2, \sigma) \in \mathcal{L} \text{ s.t. } (pk, x_1, x_2) \neq (pk', x'_1, x'_2) \wedge \text{eRO.RO}(pk, x_1, x_2) = \sigma,$$

where $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$. By Theorem 1 (3), we know that $\Pr[\text{Bad in } G_3] \leq \frac{300(q_H + q_d + 1)^3 \Gamma'(f) + 2}{2^{2\lambda}} + 8q_d^2 \sqrt{\frac{2\Gamma(f)}{2^\lambda}}$. Note that $\Gamma(f) = \Gamma'(f) = 2^\lambda$ for $f(x, y_1|y_2) = y_1$. Therefore, we have $\Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.2}] \geq \Pr[\text{Ext in } G_{3.2}] - \Pr[\text{Bad in } G_{3.2}] \geq \Pr[\text{Ext in } G_{3.2}] - \frac{300(q_H + q_d + 1)^3 + 2}{2^\lambda} - \frac{8\sqrt{2}q_d^2}{2^{\lambda/2}}$.

Game $G_{3.3}$. In $G_{3.3}$, for each $\text{DECAP}(ct = (c, \sigma))$, if $c \neq c^*$, then challenger $\mathcal{C}$ immediately extracts (pk', x'_1, x'_2) from $\text{eRO.E}(f, \sigma)$, instead of doing the extraction at the end of the game. By Theorem 1 (2.c), each swap of an invocation of eRO.RO with an invocation of eRO.E query affects the probability by at most $8\sqrt{2\Gamma(f)}/2^{2\lambda} = \frac{8\sqrt{2}}{2^{\lambda/2}}$. There are at most q_d queries to eRO.E and at most $(q_d + q_H)$ queries to eRO.RO . By a union bound we have

$$\Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.3}] \geq \Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.2}] - \frac{8\sqrt{2}q_d(q_d + q_H)}{2^{\lambda/2}}.$$

Game $G_{3.4}$. $G_{3.4}$ is the same as $G_{3.3}$, except for the simulation of DECAP oracle.

In $G_{3.4}$, upon a decryption query $\text{DECAP}(ct = (c, \sigma))$ query from $\mathcal{A}$, if $c \neq c^*$, challenger $\mathcal{C}$ computes $(x_1, x_2) := F^{-1}(c)$ and $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$, just like $G_{3.3}$. Next, a new rejection rule is added in $G_{3.4}$.

• **Rejection rule I:** If $(pk, x_1, x_2) \neq (pk', x'_1, x'_2)$, then $\mathcal{C}$ returns $\perp$ directly.

Otherwise, $\mathcal{C}$ computes $\sigma' := \text{eRO.RO}(pk', x'_1, x'_2)$, checks whether $c = F(x'_1, x'_2)$, and then proceeds with (x'_1, x'_2) just like $\mathcal{C}$ does in $G_{3.3}$ with (x_1, x_2) .

Note that $G_{3.4}$ and $G_{3.3}$ differs only when $\mathcal{A}$ makes a decryption query $\text{DECAP}(ct = (c, \sigma))$ s.t. $\mathcal{C}$ returns $m \neq \perp$ in $G_{3.3}$ but returns $\perp$ due to the rejection rule in $G_{3.4}$. This implies there exists $(pk', x'_1, x'_2) \neq (pk, x_1, x_2) \wedge \text{eRO.RO}(pk, x_1, x_2) = \sigma$ for a decryption query $\text{DECAP}(ct = (c, \sigma))$, which means event **Bad** happens in $G_{3.3}$. In other words, if event **Bad** does not happen in $G_{3.3}$, $G_{3.4}$ and $G_{3.3}$ are the same. So we have $\Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.4}] = \Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.3}]$. Obviously, $\Pr[\text{Ext in } G_{3.4}] \geq \Pr[\text{Ext} \wedge \neg \text{Bad in } G_{3.4}]$.

Game $G_{3.5}$. $G_{3.5}$ is the same as $G_{3.4}$, except that we replace the Rejection rule I with the following Rejection rule II.

- **Rejection rule II:** If $pk \neq pk'$, then $\mathcal{C}$ returns $\perp$ directly.

Otherwise, $\mathcal{C}$ computes $\sigma' := \text{eRO.RO}(pk', x'_1, x'_2)$, checks whether $c = F(x'_1, x'_2)$, and then proceeds just like $\mathcal{C}$ does in $\mathbf{G}_{3.4}$ with (x'_1, x'_2) .

Note that $\mathbf{G}_{3.5}$ and $\mathbf{G}_{3.4}$ differs only in the case that $\mathcal{A}$ makes a decryption query $\text{DECAP}(ct = (c, \sigma))$ s.t. $\mathcal{C}$ returns $m \neq \perp$ in $\mathbf{G}_{3.5}$ but returns $\perp$ due to Rejection rule I in $\mathbf{G}_{3.4}$. Let $(x_1, x_2) := F^{-1}(c)$, $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$. On the one hand, $\mathcal{C}$ returns $m \neq \perp$ in $\mathbf{G}_{3.5}$ implies that Rejection rule II does not hold, so $pk = pk'$. On the other hand, $\mathcal{C}$ returns $\perp$ due to the Rejection rule I in $\mathbf{G}_{3.4}$ means $(pk, x_1, x_2) \neq (pk', x'_1, x'_2)$. Therefore, it must hold that $(x_1, x_2) \neq (x'_1, x'_2)$. We also know that $F(x'_1, x'_2) = c$ since $\text{DECAP}(c, \sigma) \neq \perp$ in $\mathbf{G}_{3.5}$. Then we have $(x_1, x_2) := F^{-1}(c) = F^{-1}(F(x'_1, x'_2))$, which is contradictory to the injectivity of the underlying F . Therefore, the above case does not happen, and $\mathbf{G}_{3.5}$ is the same as $\mathbf{G}_{3.4}$. So we have $\Pr[\text{Ext in } \mathbf{G}_{3.5}] = \Pr[\text{Ext in } \mathbf{G}_{3.4}]$.

Thanks to eRO.E and Rejection rule II, $\mathcal{C}$ can answer DECAP queries without F^{-1} from $\mathbf{G}_{3.5}$ on.

Game $\mathbf{G}_{3.6}$. In $\mathbf{G}_{3.6}$, after $\mathcal{C}$ obtains $(\hat{x}_1, \hat{x}_2)$ by measuring $\mathcal{A}$'s j -th post-challenge hash query to H , $\mathcal{C}$ will make an additional query to $G(\hat{x}_2)$. This query $G(\hat{x}_2)$ does not affect the probability of Ext . So we have $\Pr[\text{Ext in } \mathbf{G}_{3.6}] = \Pr[\text{Ext in } \mathbf{G}_{3.5}]$. This additional query is used to remove the square root loss by applying Semi-O2H lemma in the following game.

Game $\mathbf{G}_{3.7}$. In $\mathbf{G}_{3.7}$, $\mathcal{C}$ introduces a new random oracle $\bar{G}$ such that $\bar{G}(x_2) = G(x_2)$ for all $x_2 \neq x_2^*$ and $\bar{G}(x_2^*) \leftarrow_s \mathcal{U}$. $\mathcal{C}$ still computes $x_1^* := \text{Encode}(m_b, G(x_2^*))$ with random oracle G . However, $\mathcal{C}$ uses the punctured oracle $\bar{G} \setminus \{x_2^*\}$ (cf. Definition 4) to respond any (quantum) random oracle query to G (both from $\mathcal{A}$'s queries and from oracle DECAP). Now that $\mathcal{A}$ has no information of $G(x_2^*)$, so $x_1^* := \text{Encode}(m_b, G(x_2^*))$ has the same distribution as $x_1^* := \text{Encode}(m_b, u)$ for $u \leftarrow_s \mathcal{U}$. By the semi-classical O2H lemma (Lemma 1), we have

$$\sqrt{\Pr[\text{Ext in } \mathbf{G}_{3.6}]} \leq \sqrt{\Pr[\text{Ext} \wedge \neg \text{Find in } \mathbf{G}_{3.7}]} + 2\sqrt{(q_G + q_d + 2)\Pr[\text{Find in } \mathbf{G}_{3.7}]}.$$

Here Find is defined as follows. For each $\mathcal{C}$'s (quantum) random oracle query $|x\rangle$ to $G \setminus \{x_2^*\}$, it first computes $|x\rangle|f_{x_2^*}(x)\rangle$ with an auxiliary qubit, where $f_{x_2^*}(x) = 1$ if $x_2^* = x_2$ and $f_{x_2^*}(x) = 0$ otherwise. Then it measures the auxiliary qubit $|f_{x_2^*}(x)\rangle$, and forwards the (collapsed) qubit $|x\rangle$ to hash function G . Find happens if any of the measurements of $|f_{x_2^*}(x)\rangle$ outputs 1.

Note that $\Pr[\text{Ext} \wedge \neg \text{Find in } \mathbf{G}_{3.7}] = 0$. The reason is as follows. Recall that $\mathcal{C}$ will query $\bar{G} \setminus \{x_2^*\}(\hat{x}_2)$ after it obtains $(\hat{x}_1, \hat{x}_2)$ by measuring $\mathcal{A}$'s j -th query to H . If Ext occurs, which means $(\hat{x}_1, \hat{x}_2) = (x_1^*, x_2^*)$, then this hash query $\bar{G} \setminus \{x_2^*\}(\hat{x}_2)$ must trigger the event Find . So we have $\Pr[\text{Ext} \wedge \neg \text{Find in } \mathbf{G}_{3.7}] = 0$ and thus

$$\Pr[\text{Ext in } \mathbf{G}_{3.6}] \leq 4(q_G + q_d + 2)\Pr[\text{Find in } \mathbf{G}_{3.7}].$$

Game $\mathbf{G}_{3.8}$. $\mathbf{G}_{3.8}$ is the same as $\mathbf{G}_{3.7}$ except for two changes. The first change in $\mathbf{G}_{3.8}$ is the generation of x_1^* . In $\mathbf{G}_{3.8}$, $\mathcal{C}$ samples $x_1^* \leftarrow \psi_1$ at the beginning of the game. Recall that in $\mathbf{G}_{3.7}$, $x_1^* := \text{Encode}(m_b, u)$ for $u \leftarrow_s \mathcal{U}$. By the ψ_1 -message

hiding property of underlying SOTP, we know that $x_1^* = \text{Encode}(m_b, u)$ follows distribution ψ_1 . Therefore, this is only a conceptual change.

The second change is that $\mathcal{C}$ uses $\overline{G}$ rather than the punctured oracle $\overline{G} \setminus \{x_2^*\}$ to respond to any (quantum) random oracle query to G . Besides, $\mathcal{C}$ randomly samples $i \leftarrow [q_G + q_d + 1]$, and measures the i -th query to G to obtain $\hat{x}_2$. By Lemma 2, we have $\Pr[\text{Find in } \mathbf{G}_{3.7}] \leq 4(q_G + q_d + 2) \cdot \Pr[\hat{x}_2 = x_2^* \text{ in } \mathbf{G}_{3.8}]$.

Now in $\mathbf{G}_{3.8}$, $\mathcal{C}$ can simulate the whole game only with $\mathbf{F}$ and c^* . So we are ready to prove $\Pr[\hat{x}_2 = x_2^* \text{ in } \mathbf{G}_{3.8}] \leq \text{Adv}_{\mathbf{F}, \mathbf{F}^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B})$ with a security reduction.

Let $\mathcal{B}$ be an algorithm that takes as input $(\mathbf{F}, c^*)$ with $c^* := \mathbf{F}(x_1^*, x_2^*)$ where $(x_1^*, x_2^*) \leftarrow (\psi_1, \psi_2)$, and aims to find x_2^* . To this end, $\mathcal{B}$ first samples $i \leftarrow [q_G + q_d + 1]$ and simulates $\mathbf{G}_{3.8}$ with $(\mathbf{F}, c^*)$ and random oracle $\overline{G}$ just before the i -th query to $\overline{G}$. Then $\mathcal{B}$ measures the argument of the i -th query in the computational basis and outputs $\hat{x}_2$. Clearly, up to now $\mathcal{B}$ perfectly simulates $\mathbf{G}_{3.8}$ for $\mathcal{A}$. If $\hat{x}_2 = x_2^*$ happens in $\mathbf{G}_{3.8}$, then $\mathcal{B}$ wins. Therefore, we have

$$\Pr[\hat{x}_2 = x_2^* \text{ in } \mathbf{G}_{3.8}] \leq \text{Adv}_{\mathbf{F}, \mathbf{F}^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B}).$$

Backtracking all the above (in)equalities from $\mathbf{G}_{3.8}$ to $\mathbf{G}_3$ and set $q := q_G + q_H + q_d + 2$, we obtain that

$$|\Pr[\mathbf{G}_3 \Rightarrow 1] - \Pr[\mathbf{G}_2 \Rightarrow 1]| \leq 2q \sqrt{16q^2 \text{Adv}_{\mathbf{F}, \mathbf{F}^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B}) + \frac{16\sqrt{2}q^2}{2^{\lambda/2}} + \frac{300q^3}{2^\lambda} + \frac{6\sqrt{q}}{2^{\kappa/4}}}.$$

Game $\mathbf{G}_4$. $\mathbf{G}_4$ is the same as $\mathbf{G}_3$, except that for two changes. Firstly, $\mathcal{C}$ uses the extractable random oracle simulator eRO to simulate random oracle H , and simulate DECAP without usage of $\mathbf{F}^{-1}$. Secondly, upon a decryption query $\text{DECAP}(ct = (c, \sigma))$, if $pk \neq pk'$, then $\mathcal{C}$ returns $\perp$ directly, where $(pk', x_1, x_2) := \text{eRO.E}(f, \sigma)$. With the same argument from $\mathbf{G}_{3.2}$ to $\mathbf{G}_{3.5}$, we have

$$|\Pr[\mathbf{G}_4 \Rightarrow 1] - \Pr[\mathbf{G}_3 \Rightarrow 1]| \leq \frac{16\sqrt{2}q^2}{2^{\lambda/2}} + \frac{300q^3}{2^\lambda}.$$

Now $\mathcal{C}$ can simulate the recovery of (x_1, x_2) in oracle DECAP without $\mathbf{F}^{-1}$.

Game $\mathbf{G}_5$. In $\mathbf{G}_5$, $\mathcal{C}$ introduces a new random oracle $\overline{G}$ satisfying $\overline{G}(x_2) = G(x_2)$ for all $x_2 \neq x_2^*$ and $\overline{G}(x_2^*) \leftarrow \mathcal{U}$. $\mathcal{C}$ still computes $x_1^* := \text{Encode}(m_b, G(x_2^*))$ with random oracle G . However, $\mathcal{C}$ uses the new oracle $\overline{G}$ to respond to (quantum) random oracle queries to G from $\mathcal{A}$ and the DECAP oracle. By the semi-classical O2H lemma (Lemma 1), we have

$$|\Pr[\mathbf{G}_5 \Rightarrow 1] - \Pr[\mathbf{G}_4 \Rightarrow 1]| \leq 2\sqrt{(q_G + 1)\Pr[\text{Find in } \mathbf{G}_5^{\overline{G} \setminus \{x_2^*\}}]},$$

where $\mathbf{G}_5^{\overline{G} \setminus \{x_2^*\}}$ is similar to $\mathbf{G}_5$ except that $\overline{G}$ is replaced by $\overline{G} \setminus \{x_2^*\}$. Find is the event that any of the measurement of $|f_{x_2^*}(x)\rangle$ returns 1 in $\mathbf{G}_5^{\overline{G} \setminus \{x_2^*\}}$.

Now in $\mathbf{G}_5$, $\mathcal{A}$ has no information of $G(x_2^*)$, so $G(x_2^*)$ is uniform. By the message hiding property of underlying SOTP, $\text{Encode}(m_0; G(x_2^*))$ must have the same distribution as that of $\text{Encode}(m_1; G(x_2^*))$. So we have $\Pr[\mathbf{G}_5 \Rightarrow 1] = 1/2$.

The remaining is to bound the probability of the event Find in $G_5^{\overline{G} \setminus \{x_2^*\}}$.

Game G_6 . G_6 is similar to G_5 , except that $\mathcal{C}$ first samples $i \leftarrow_{\$} [q_G]$ and measures the argument of the i -th query to $\overline{G}$ in the computational basis and outputs x'_2 . If $x'_2 = x_2^*$, then Find happens in G_6 . By Lemma 2, we have

$$\Pr [\text{Find in } G_5^{\overline{G} \setminus \{x_2^*\}}] \leq 4q_G \cdot \Pr [\text{Find in } G_6].$$

Now we prove $\Pr [\text{Find in } G_6] \leq \text{Adv}_{F, F^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B})$. Let $\mathcal{B}$ be an algorithm that takes as input (F, c^*) with $c^* := F(x_1^*, x_2^*)$ where $(x_1^*, x_2^*) \leftarrow (\psi_1, \psi_2)$. $\mathcal{B}$'s aim is to find x_2^* . To this end, $\mathcal{B}$ first samples $i \leftarrow_{\$} [q_G]$ and runs $\mathcal{C}$ in G_6 with (F, c^*) and random oracle $\overline{G}$ just before $\mathcal{C}$ makes the i -th query to $\overline{G}$. Then $\mathcal{B}$ measures the argument of the query in the computational basis and outputs x'_2 . Note that $\mathcal{B}$ perfectly simulates G_6 . If Find happens, then $x'_2 = x_2^*$ and $\mathcal{B}$ wins. So we have

$$\Pr [\text{Find in } G_6] \leq \text{Adv}_{F, F^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B}).$$

Backtracking all the above (in)equalities and set $q := q_G + q_H + q_d + 2$, we get the following bound for any (F, F^{-1}) with injectivity:

$$\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A}) \leq 16q^2 \sqrt{\text{Adv}_{F, F^{-1}, \psi}^{\text{PD-OW}}(\mathcal{B})} + 4q \sqrt{\frac{300q^3}{2^\lambda} + \frac{16\sqrt{2}q^2}{2^{\lambda/2}} + \frac{6\sqrt{q}}{2^{\kappa/4}}}.$$

For (F, F^{-1}) without injectivity, we have $\text{Adv}_{F, F^{-1}}^{\text{CCA}}(\mathcal{A}) \leq 1$. Taking the expectation over $(F, F^{-1}) \leftarrow \text{Gen}$ and applying Jensen's inequality, Thm. 2 follows. $\square$

GAME SETUP G_0 - G_6	$\text{DEC}_{ct^*}(ct)$ G_0 - G_6
01: $(pk, sk) := (F, F^{-1})$	17: If $ct = ct^*$: Return $\perp$
02: $b \leftarrow_{\$} \{0, 1\}$	18: Parse $ct^* := (c^*, \sigma^*)$
03: $x_1^* \leftarrow \psi_1$	19: Parse $ct := (c, \sigma)$
04: $x_2^* \leftarrow \psi_2$	20: If $c = c^*$: Return $\perp$
05: $(m_0, m_1, st) \leftarrow \mathcal{A}^{G, H, \text{DECAP}_\perp}(pk)$	21: $(x_1, x_2) := F^{-1}(c)$
06: $(m_0, m_1, st) \leftarrow \mathcal{A}^{G, \text{eRO}, \text{RO}, \text{DECAP}_\perp}(pk)$	22: $(pk', x_1, x_2) := \text{eRO.E}(f, \sigma)$
07: $(m_0, m_1, st) \leftarrow \mathcal{A}^{\overline{G} \setminus \{x_2^*\}, \text{eRO}, \text{RO}, \text{DECAP}_\perp}(pk)$	23: If $pk' \neq pk$: Return $\perp$
08: $x_1^* := \text{Encode}(m_b, G(x_2^*))$	24: If $(x_1, x_2) = \perp$: Return $\perp$
09: $c^* := F(x_1^*, x_2^*)$	25: If $F(x_1, x_2) \neq c$: Return $\perp$
10: $(\sigma^*, k^*) := H(pk, x_1^*, x_2^*)$	26: $(\sigma', k) := H(pk, x_1, x_2)$
11: $\sigma^* \leftarrow_{\$} \{0, 1\}^\lambda, k^* \leftarrow_{\$} \{0, 1\}^\lambda$	27: $(\sigma', k) := \text{eRO.RD}(pk, x_1, x_2)$
12: $ct^* := (c^*, \sigma^*), k_b^* := k^*, k_1^* \leftarrow_{\$} \{0, 1\}^\lambda$	28: If $\sigma' \neq \sigma$: Return $\perp$
13: $b' \leftarrow \mathcal{A}^{G, H, \text{DECAP}_{ct^*}}(st, ct^*, k_b^*)$	29: $r := G(x_2)$
14: $b' \leftarrow \mathcal{A}^{G, \text{eRO}, \text{RO}, \text{DECAP}_{ct^*}}(st, ct^*, k_b^*)$	30: $r := \overline{G}(x_2)$
15: $b' \leftarrow \mathcal{A}^{\overline{G} \setminus \{x_2^*\}, \text{eRO}, \text{RO}, \text{DECAP}_{ct^*}}(st, ct^*, k_b^*)$	31: If $x_2 = x_2^*$: Find := 1
16: Return $b = b'$	32: $m := \text{Decode}(x_1, r)$
	33: Return (m, k)

Fig. 5: G_0 - G_6 for the proof of Theorem 2. If a code line has no “//”, this line occurs in all games G_0 - G_6 .

GAME SETUP $G_{3.1}-G_{3.8}$	$DEC_{ct^*}(ct)$ $G_{3.1}-G_{3.3}$
01: $(pk, sk) := (F, F^{-1})$	29: If $ct = ct^*$: Return $\perp$
02: $b \leftarrow \{0, 1\}$, Ext := 0, Find := 0	30: Parse $ct^* := (c^*, \sigma^*)$
03: $x_1^* \leftarrow \psi_1$, $i \leftarrow [q_G + q_d + 1]$ // $G_{3.8}$	31: Parse $ct := (c, \sigma)$
04: $x_2^* \leftarrow \psi_2$	32: If $c = c^*$: Return $\perp$
05: $(m_0, m_1, st) \leftarrow \mathcal{A}^{G, H, \text{DECAP}_\perp}(pk)$ // $G_{3.1}$	33: $(x_1, x_2) := F^{-1}(c)$
06: $(m_0, m_1, st) \leftarrow \mathcal{A}^{G, \text{eRO.RO}, \text{DECAP}_\perp}(pk)$ // $G_{3.2}-G_{3.6}$	34: $\mathcal{L} := \mathcal{L} \cup \{(x_1, x_2, \sigma)\}$ // $G_{3.2}-G_{3.3}$
// H is replaced with eRO.RO	35: $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$ // $G_{3.3}$
07: $(m_0, m_1, st) \leftarrow \mathcal{A}^{\bar{G} \setminus \{x_2^*\}, \text{eRO.RO}, \text{DECAP}_\perp}(pk)$ // $G_{3.7}$	36: If $F(x_1, x_2) \neq c$: Return $\perp$
// G is replaced with $\bar{G} \setminus \{x_2^*\}$	37: $(\sigma', k) := H(pk, x_1, x_2)$ // $G_{3.1}$
08: $(m_0, m_1, st) \leftarrow \mathcal{A}^{\bar{G}, \text{eRO.RO}, \text{DECAP}_\perp}(pk)$ // $G_{3.8}$	38: $(\sigma', k) := \text{eRO.RO}(pk, x_1, x_2)$ // $G_{3.2}-G_{3.3}$
09: $x_1^* := \text{Encode}(m_b, G(x_2^*))$ // $G_{3.1}-G_{3.6}$	39: If $\sigma' \neq \sigma$: Return $\perp$
10: $u \leftarrow \mathcal{U}$, $x_1^* := \text{Encode}(m_b, u)$ // $G_{3.7}$	40: $r := G(x_2)$
11: $c^* := F(x_1^*, x_2^*)$	41: $m := \text{Decode}(x_1, r)$
12: $\sigma^* \leftarrow \{0, 1\}^\lambda$, $k^* \leftarrow \{0, 1\}^\lambda$	42: Return (m, k)
13: $ct^* := (c^*, \sigma^*)$, $k_0^* := k^*$, $k_1^* \leftarrow \{0, 1\}^\lambda$	
14: $j \leftarrow [q_H]$	$DEC_{ct^*}(ct)$ $G_{3.4}-G_{3.7}$
15: $b' \leftarrow \mathcal{A}^{G, H, \text{DECAP}_{ct^*}}(st, ct^*, k^*)$ // $G_{3.1}$	43: If $ct = ct^*$: Return $\perp$
16: $b' \leftarrow \mathcal{A}^{G, \text{eRO.RO}, \text{DECAP}_{ct^*}}(st, ct^*, k^*)$ // $G_{3.2}-G_{3.6}$	44: Parse $ct^* := (c^*, \sigma^*)$
// H is replaced with eRO.RO	45: Parse $ct := (c, \sigma)$
17: $b' \leftarrow \mathcal{A}^{\bar{G} \setminus \{x_2^*\}, \text{eRO.RO}, \text{DECAP}_{ct^*}}(st, ct^*, k^*)$ // $G_{3.7}$	46: If $c = c^*$: Return $\perp$
// G is replaced with $\bar{G} \setminus \{x_2^*\}$	47: $(x_1, x_2) := F^{-1}(c)$ // $G_{3.4}$
18: $b' \leftarrow \mathcal{A}^{\bar{G}, \text{eRO.RO}, \text{DECAP}_{ct^*}}(st, ct^*, k^*)$ // $G_{3.8}$	48: $\mathcal{L} := \mathcal{L} \cup \{(x_1, x_2, \sigma)\}$ // $G_{3.4}$
19: Measure $\mathcal{A}$'s j -th query to H to obtain $(\hat{x}_1, \hat{x}_2)$	49: $(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$
20: If $(\hat{x}_1, \hat{x}_2) = (x_1^*, x_2^*)$: Ext := 1	50: If $(pk, x_1, x_2) \neq (pk', x'_1, x'_2)$: // $G_{3.4}$
21: $\hat{\sigma} := G(\hat{x}_2)$ // $G_{3.6}$	Return $\perp$
22: $\hat{\sigma} := \bar{G} \setminus \{x_2^*\}(\hat{x}_2)$ // $G_{3.7}$	51: If $pk \neq pk'$: Return $\perp$ // $G_{3.5}-G_{3.8}$
23: If $\hat{x}_2 = x_2^*$: Find := 1 // $G_{3.7}$	52: If $F(x'_1, x'_2) \neq c$: Return $\perp$
24: $\hat{\sigma} := \bar{G}(\hat{x}_2)$ // $G_{3.8}$	53: $(\sigma', k) := \text{eRO.RO}(pk, x'_1, x'_2)$
25: For all $(x_1, x_2, \sigma) \in \mathcal{L}$:	54: If $\sigma' \neq \sigma$: Return $\perp$
$(pk', x'_1, x'_2) \leftarrow \text{eRO.E}(f, \sigma)$ // $G_{3.2}$	55: $r := G(x'_2)$ // $G_{3.4}-G_{3.6}$
26: Define event Bad : $\exists (x_1, x_2, \sigma) \in \mathcal{L}$ s.t. // $G_{3.2}-G_{3.4}$	56: $r := \bar{G} \setminus \{x_2^*\}(x'_2)$ // $G_{3.7}$
$(pk, x_1, x_2) \neq (pk', x'_1, x'_2) \wedge \text{eRO.RO}(pk, x_1, x_2) = \sigma$	57: If $x'_2 = x_2^*$: Find := 1 // $G_{3.7}$
27: Measure $\mathcal{A}$'s i -th query to $\bar{G}$ to obtain $\hat{x}_2$ // $G_{3.8}$	58: $r := \bar{G}(x'_2)$ // $G_{3.8}$
28: Return Ext	59: $m := \text{Decode}(x'_1, r)$
	60: Return (m, k)

Fig. 6: $G_{3.1}-G_{3.8}$ for the proof of Theorem 2. If a code line has no “//”, this line occurs in all games.

To show the anonymity (ANO-CCA security) for KEM, we prove a stronger PR-CCA security for KEM.

Theorem 3. Suppose that $\mathcal{F}$ is an η -almost injective function (η -ATIF) family and has rigidity, $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{M}, \mathcal{U}, \mathcal{X}_1, \psi_1)$ is a semi-generalized one-time pad, and ψ_2 is a distribution with collision entropy κ . Then scheme KEM from $\text{OAEP}^\dagger$ in Fig. 4 has PR-CCA security. More precisely, for any PR-CCA adversary $\mathcal{A}$ issuing q_G and q_H (quantum) random oracle queries to G and H , and q_d decryption queries, if $q = q_G + q_H + q_d + 2$, then we have

$$\text{Adv}_{\text{KEM}}^{\text{PR-CCA}}(\mathcal{A}) \leq \text{Adv}_{\mathcal{F}, \psi}^{\text{PR}}(\mathcal{B}) + 16q^2 \sqrt{\text{Adv}_{\mathcal{F}, \psi}^{\text{PD-OW}}(\mathcal{B})} + 4q \sqrt{\frac{300q^3}{2^\lambda} + \frac{16\sqrt{2}q^2}{2^{\lambda/2}} + \frac{6\sqrt{q}}{2^{\kappa/4}}} + \eta.$$

Proof. The proof outline is almost the same as Theorem 2.

Game G'_0 . G'_0 is the original $\text{Exp}_{\text{KEM}}^{\text{PR-CCA}}$. We have $\text{Adv}_{\text{KEM}}^{\text{PR-CCA}}(\mathcal{A}) = |\Pr[G'_0 \Rightarrow 1] - 1/2|$.

Game G'_1 . The hybrid from G'_0 to G'_1 is similar to the hybrid from G_0 to G_5 in the proof of Theorem 2, so we have

$$|\Pr[G'_1 \Rightarrow 1] - \Pr[G'_0 \Rightarrow 1]| \leq 16q^2 \sqrt{\text{Adv}_{\mathcal{F}, \psi}^{\text{PD-OW}}(\mathcal{B})} + 4q \sqrt{\frac{300q^3}{2^\lambda} + \frac{8\sqrt{2}q^2}{2^{\lambda/2}} + \frac{6\sqrt{q}}{2^{\kappa/4}}} + \eta.$$

In G'_1 , the challenger generates the challenge ciphertext $ct_0^* := (c_0^*, \sigma_0^*)$ and the challenge key k_b^* by $c_0^* := F(x_1^*, x_2^*)$, $\sigma_0^* \leftarrow_{\$} \{0, 1\}^\lambda$ and $k_b^* \leftarrow_{\$} \{0, 1\}^\lambda$, where $(x_1^*, x_2^*) \leftarrow (\psi_1, \psi_2)$. The challenger simulates H with eRO.RO and simulates G with $\bar{G}$. It simulates DECAP oracle with eRO.E instead of F^{-1} . The description of G'_1 is shown in Fig. 7.

Next we show that $|\Pr[G'_1 \Rightarrow 1] - \frac{1}{2}| \leq \text{Adv}_{\mathcal{F}, \psi}^{\text{PR}}(\mathcal{B})$. Let $\mathcal{B}$ be an algorithm that takes as input (F, c_b) , where $c_0 = F(x_1, x_2)$ when $b = 0$ and $c_1 \leftarrow_{\$} \mathcal{Y}$ when $b = 1$. $\mathcal{B}$ aims to guess the bit b . To this end, it simulates G'_1 for $\mathcal{A}$ as follows. $\mathcal{B}$ forwards $pk := F$ to $\mathcal{A}$ and simulates DECAP oracle with eRO. After $\mathcal{A}$ outputs (m, st) , $\mathcal{B}$ randomly samples $\sigma \leftarrow_{\$} \{0, 1\}^\lambda$ and $k_b^* \leftarrow_{\$} \{0, 1\}^\lambda$, outputs $ct_b^* = (c_b, \sigma)$ and k_b^* to $\mathcal{A}$. Finally, $\mathcal{B}$ just takes the guessing bit b' of $\mathcal{A}$ as its own output. Note that $\mathcal{B}$'s simulation of G'_1 is perfect for $\mathcal{A}$. Therefore, if $\mathcal{A}$ wins G'_1 , i.e., $G'_1 \Rightarrow 1$, then $\mathcal{B}$ breaks the pseudo-randomness of the ATIF family F . So we have $|\Pr[G'_1 \Rightarrow 1] - \frac{1}{2}| \leq \text{Adv}_{\mathcal{F}, \psi}^{\text{PR}}(\mathcal{B})$, and finish the proof of Theorem 3. $\square$

Now we show the robustness of our $\text{OAEP}^\dagger$ transform.

Theorem 4. If the ATIF family $\mathcal{F}$ and the hash function H both have collision resistant, then scheme KEM from $\text{OAEP}^\dagger$ in Fig. 4 has ROB-CCA security. More precisely, for any ROB-CCA adversary $\mathcal{A}$ issuing q_H (quantum) random oracle queries to H , we have $\text{Adv}_{\text{KEM}}^{\text{ROB-CCA}}(\mathcal{A}) \leq \frac{648(q_H+1)^3}{2^\lambda} + \text{negl}(\lambda)$.

Proof. Let $(pk_0, sk_0) = (F_0, F_0^{-1}) \leftarrow \text{Gen}$ and $(pk_1, sk_1) = (F_1, F_1^{-1}) \leftarrow \text{Gen}$. Then by the collision resistance of ATIF, we know that $pk_0 \neq pk_1$ overwhelmingly. Suppose $\mathcal{A}$ outputs a ciphertext $ct = (c, \sigma)$ and breaks the ROB-CCA security of

GAME G'_1	$\text{DEC}_{ct_b^*}(ct)$
01: $(pk, sk) := (F, F^{-1})$	13: If $ct = ct_b^*$: Return $\perp$
02: $b \leftarrow_{\$} \{0, 1\}$	14: Parse $ct_b^* := (c_b^*, \sigma_b^*)$
03: $x_1^* \leftarrow \psi_1, x_2^* \leftarrow \psi_2$	15: Parse $ct := (c, \sigma)$
04: $(m, st) \leftarrow \mathcal{A}^{\overline{G}, \text{eRO}, \text{RO}, \text{DECAP}_{\perp}}(pk)$	16: If $c = c_b^*$: Return $\perp$
05: $c_0^* := F(x_1^*, x_2^*)$	17: $(pk', x'_1, x'_2) := \text{eRO.E}(f, \sigma)$
06: $\sigma_0^* \leftarrow_{\$} \{0, 1\}^\lambda$	18: If $pk' \neq pk$: Return $\perp$
07: $ct_0^* := (c_0^*, \sigma_0^*), k_0^* \leftarrow_{\$} \{0, 1\}^\lambda$	19: If $F(x'_1, x'_2) \neq c$: Return $\perp$
08: $c_1^* \leftarrow_{\$} \mathcal{Y}$	20: $(\sigma', k) := \text{eRO.RO}(pk', x'_1, x'_2)$
09: $\sigma_1^* \leftarrow_{\$} \{0, 1\}^\lambda$	21: If $\sigma' \neq \sigma$: Return $\perp$
10: $ct_1^* := (c_1^*, \sigma_1^*), k_1^* \leftarrow_{\$} \{0, 1\}^\lambda$	22: $r := \overline{G}(x'_2)$
11: $b' \leftarrow \mathcal{A}^{\overline{G}, \text{eRO}, \text{RO}, \text{DECAP}_{ct_b^*}}(st, pk, ct_b^*, k_b^*)$	23: $m := \text{Decode}(x'_1, r)$
12: Return $b = b'$	24: Return (m, k)

Fig. 7: G'_1 for proof in Theorem 3.

KEM. Let $(x_1, x_2) := F_0^{-1}(c)$ and $(x'_1, x'_2) := F_1^{-1}(c)$. If $(\sigma, k) := H(pk_0, x_1, x_2)$ and $(\sigma', k') = H(pk_1, x'_1, x'_2)$, then we have $\sigma = \sigma'$. Further define $\tilde{H}(\cdot, \cdot, \cdot)$ as the first half output of H , then $\sigma = \tilde{H}(pk_0, x_1, x_2) = \tilde{H}(pk_1, x'_1, x'_2)$. This means that $\mathcal{A}$ has broken the collision resistance of the underlying hash function $\tilde{H}$, which happens with probability at most $\frac{648(q_H+1)^3}{2^\lambda}$ by Lemma 4. $\square$

4 Instantiations of KEM via OAEP[†]

4.1 Instantiations of ATIF from NTRU

We denote by $\mathcal{R}$ the polynomial ring $\mathbb{Z}_q[X]/(f(X))$, where $f(X)$ is a n -degree polynomial and q is an integer. We will use bold lower-case letters, like $\mathbf{g} = (g_0, g_1, \dots, g_{n-1})$, to represent polynomials, like $g_0 + g_1X + \dots + g_{n-1}X^{n-1}$, in $\mathcal{R}$. The infinity norm $\|\mathbf{g}\|_\infty$ represents $\max\{|g_0|, \dots, |g_{n-1}|\}$. Define CBD_1^n as the distribution over $\mathcal{R}$, where each coefficient is generated by first sampling $a, b \leftarrow_{\$} \{0, 1\}$ and then outputting $b - a \in \{-1, 0, 1\}$ with “ $-$ ” being bit-wise subtraction over $\mathbb{Z}$. Let $\mathcal{S}_d^n$ be a subset of $\mathcal{R}$, where $\mathbf{g} \in \mathcal{S}_d^n$ if $\mathbf{g}$ has d coefficients being 1, d coefficients being -1 , and all other coefficients being 0. We use Tern_d^n to denote the uniform distribution over $\mathcal{S}_d^n$. Denote by U_3^n a distribution over $\mathcal{R}$, where each coefficient of $\mathbf{g} \leftarrow_{\$} U_3^n$ is sampled uniformly from $\{-1, 0, 1\}$. For an integer h , we have $h \bmod q \in \{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\}$, where we shift $\mathbb{Z}_q$ from $\{0, 1, \dots, q-1\}$ to $\{-\frac{q-1}{2}, \dots, \frac{q-1}{2}\}$.

The $\text{NTRU}_{\mathcal{R}, \psi}$ assumption is a widely-used post-quantum assumption, which assumes that $\mathbf{h} := \mathbf{g}(p\mathbf{f}' + 1)^{-1} \in \mathcal{R}$ for $(\mathbf{f}', \mathbf{g}) \leftarrow \psi$ is indistinguishable from a random element $\mathbf{u} \leftarrow_{\$} \mathcal{R}$. Here we parse $\psi = (\psi_1, \psi_2)$, so $\mathbf{f}' \leftarrow \psi_1$ and $\mathbf{g} \leftarrow \psi_2$. We also require that for any $\mathbf{w} \in \mathcal{R}$, it holds that $\Pr[\mathbf{g} = \mathbf{w} : \mathbf{g} \leftarrow \psi_2] \leq \text{negl}(\lambda)$. This requirement is naturally satisfied by the NTRU assumption.

Definition 11 (The $\text{NTRU}_{\mathcal{R}, \psi}$ Assumption [27, 20]). Let q be a positive integer, and p be an integer co-prime to q . Let $\mathcal{R} := \mathbb{Z}_q[X]/(f(X))$ for some polynomial $f(X) \in \mathbb{Z}_q[x]$ and ψ be a distribution over $\mathcal{R}$. The advantage function of

Gen :	F(s, e) :	F ⁻¹ (c ∈ R) :
(f', g) ← ψ, f := 3f' + 1	If s ∉ {-1, 0, 1}^n: Return(⊥)	Parse F ⁻¹ = (f, h ⁻¹)
If f or g is not invertible in R:	If e ∉ {-1, 0, 1}^n: Return(⊥)	e := (cf mod q) mod 3
re-sample f', g	Parse F = h	s := h ⁻¹ (c - e)
h := 3gf ⁻¹	c := hs + e	If s ∉ {-1, 0, 1}^n: Return ⊥
Return (F, F ⁻¹) = (h, (f, h ⁻¹))	Return c	Return (s, e)

Fig. 8: Almost trapdoor injective function family $\mathcal{F}_{\text{ntru}} = (\text{Gen}, F, F^{-1})$ from NTRU.

an adversary $\mathcal{A}$ is defined as $\text{Adv}_{\mathcal{R}, \psi}^{\text{NTRU}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathbf{h}) \Rightarrow 1] - \Pr[\mathcal{A}(\mathbf{u}) \Rightarrow 1]|$, where $(\mathbf{f}', \mathbf{g}) \leftarrow \psi$, $\mathbf{h} := \mathbf{g}(\mathbf{p}\mathbf{f}' + 1)^{-1}$, and $\mathbf{u} \leftarrow_{\$} \mathcal{R}$. The $\text{NTRU}_{\mathcal{R}, \psi}$ assumption holds if $\text{Adv}_{\mathcal{R}, \psi}^{\text{NTRU}}(\mathcal{A}) = \text{negl}(\lambda)$ for all quantum polynomial-time algorithm $\mathcal{A}$.

Definition 12 (The Computation RLWE_{ℛ,ψ} Assumption). Let ψ be a distribution over $\mathcal{R}$. The computation Ring-LWE (RLWE_{ℛ,ψ}) assumption states that $\text{Adv}_{\mathcal{R}, \psi}^{\text{cRLWE}}(\mathcal{A}) = \text{negl}(\lambda)$ for all quantum polynomial time algorithm $\mathcal{A}$, where $\text{Adv}_{\mathcal{R}, \psi}^{\text{cRLWE}}(\mathcal{A}) := \Pr[\mathbf{e} = \mathbf{e}' : (\mathbf{s}, \mathbf{e}) \leftarrow \psi, \mathbf{h} \leftarrow_{\$} \mathcal{R}, \mathbf{e}' \leftarrow \mathcal{A}(\mathbf{h}, \mathbf{h}\mathbf{s} + \mathbf{e})]$.

Definition 13 (The Decision RLWE_{ℛ,ψ} Assumption). Let ψ be a distribution over $\mathcal{R}$. The decision Ring-LWE (RLWE_{ℛ,ψ}) assumption states that $\text{Adv}_{\mathcal{R}, \psi}^{\text{dRLWE}}(\mathcal{A}) := |\Pr[\mathcal{A}(\mathbf{h}, \mathbf{h}\mathbf{s} + \mathbf{e}) \Rightarrow 1] - \Pr[\mathcal{A}(\mathbf{h}, \mathbf{u}) \Rightarrow 1]|$ is negligible, where $\mathbf{s}, \mathbf{e} \leftarrow \psi$ and $\mathbf{h}, \mathbf{u} \leftarrow_{\$} \mathcal{R}$.

Now we construct an almost injective function family $\mathcal{F}_{\text{ntru}} = (\text{Gen}, F, F^{-1})$ and show its properties based on the $\text{NTRU}_{\mathcal{R}, \psi}$ and $\text{RLWE}_{\mathcal{R}, \psi}$ assumptions. See Fig. 8 for the construction of ATIF family $\mathcal{F}_{\text{ntru}}$.

Theorem 5. The ATIF family $\mathcal{F}_{\text{ntru}} : \{-1, 0, 1\}^n \times \{-1, 0, 1\}^n \rightarrow \mathcal{R}$ in Fig. 8 has partial one-wayness, pseudo-randomness, collision resistance and rigidity.

Proof. We prove these properties for $\mathcal{F}_{\text{ntru}}$ as follows.

Partial One-Wayness w.r.t. ψ . We show this property follows from the $\text{NTRU}_{\mathcal{R}, \psi}$ and $\text{RLWE}_{\mathcal{R}, \psi}$ assumptions.

First, we change the generation of $F = \mathbf{h}$ from $\mathbf{h} := \mathbf{g}(\mathbf{p}\mathbf{f}' + 1)^{-1}$ to $\mathbf{h} \leftarrow_{\$} \mathcal{R}$. This change is indistinguishable due to the NTRU assumption. Next, the computational RLWE assumption implies the hardness of recovering $\mathbf{e}$ from $\mathbf{h}$ and $\mathbf{c} := \mathbf{h}\mathbf{s} + \mathbf{e}$. Therefore, $\text{Adv}_{\mathcal{F}_{\text{ntru}}, \psi}^{\text{PD-OW}}(\mathcal{A}) \leq \text{Adv}_{\mathcal{R}, \psi}^{\text{NTRU}}(\mathcal{A}_1) + \text{Adv}_{\mathcal{R}, \psi}^{\text{cRLWE}}(\mathcal{A}_2)$ for any quantum algorithm $\mathcal{A}_1$ and $\mathcal{A}_2$. So the partial one-wayness of $\mathcal{F}_{\text{ntru}}$ follows from the NTRU and RLWE assumptions.

Pseudo-Randomness w.r.t. ψ & Collision Resistance. The pseudo-randomness of $\mathcal{F}_{\text{ntru}}$ also follows the NTRU and RLWE assumptions. We first make an indistinguishable change of $\mathbf{h}$ from $\mathbf{h} := \mathbf{g}(\mathbf{p}\mathbf{f}' + 1)^{-1}$ to $\mathbf{h} \leftarrow_{\$} \mathcal{R}$, where the indistinguishability is justified by the NTRU assumption. Next, the decision RLWE assumption implies the pseudo-randomness of $\mathcal{F}_{\text{ntru}}$, so $\text{Adv}_{\mathcal{F}_{\text{ntru}}, \psi}^{\text{PR}}(\mathcal{A}) \leq \text{Adv}_{\mathcal{R}, \psi}^{\text{NTRU}}(\mathcal{A}_1) + \text{Adv}_{\mathcal{R}, \psi}^{\text{dRLWE}}(\mathcal{A}_2)$.

$\mathcal{F}_{\text{ntru}}$ has collision resistance since $\Pr[\mathbf{h}_1 = \mathbf{h}_2] = \Pr[3\mathbf{g}_1\mathbf{f}_1^{-1} = 3\mathbf{g}_2\mathbf{f}_2^{-1}] = \Pr[\mathbf{g}_1 = \mathbf{g}_2\mathbf{f}_2^{-1}\mathbf{f}_1 : \mathbf{g}_1 \leftarrow \psi_2] = \text{negl}(\lambda)$.

Rigidity. For any $\mathbf{c} \in \mathcal{R}$, if $(\mathbf{s}, \mathbf{e}) := \mathbf{F}^{-1}(\mathbf{c}) \neq \perp$, then we have $\mathbf{s} = \mathbf{h}^{-1}(\mathbf{c} - \mathbf{e})$. So $\mathbf{c} = \mathbf{h}\mathbf{s} + \mathbf{e} = \mathbf{F}(\mathbf{s}, \mathbf{e})$ always holds.

Gen :	Encap($pk, m \in \mathcal{M}$) :	Decap(sk, ct) :
$(\mathbf{f}', \mathbf{g}) \leftarrow \psi = (\psi_1, \psi_2)$	Parse $pk = \mathbf{h}$	Parse $sk = (\mathbf{f}, \mathbf{h}^{-1}, pk)$ and $ct = (\mathbf{c}, \sigma)$
$\mathbf{f} := 3\mathbf{f}' + 1$	$\mathbf{e} \leftarrow \psi_2$	$\mathbf{e} := \mathbf{c}\mathbf{f} \bmod q \bmod 3, \mathbf{s} := \mathbf{h}^{-1}(\mathbf{c} - \mathbf{e})$
If $\mathbf{f}$ or $\mathbf{g}$ is not invertible in $\mathcal{R}$: restart	$\mathbf{s} := \text{Encode}(m, G(\mathbf{e}))$	If $\mathbf{s} \notin \{-1, 0, 1\}^n$: Return $\perp$
$\mathbf{h} := 3\mathbf{g}\mathbf{f}^{-1}$	$\mathbf{c} := \mathbf{h}\mathbf{s} + \mathbf{e}$	$(\sigma', k') := H(pk, \mathbf{s}, \mathbf{e})$
$pk := \mathbf{h}, sk := (\mathbf{f}, \mathbf{h}^{-1}, pk)$	$(\sigma, k) := H(pk, \mathbf{s}, \mathbf{e})$	If $\sigma' \neq \sigma$: Return $\perp$
Return (pk, sk)	$ct := (\mathbf{c}, \sigma)$	$m' := \text{Decode}(\mathbf{s}, G(\mathbf{e}))$
	Return (ct, k)	Return m', k'

Fig. 9: Scheme $\text{KEM}_{\text{NTRU}} = (\text{Gen}, \text{Encap}, \text{Decap})$ from $\mathcal{F}_{\text{ntru}}$ via our OAEP[†].

4.2 Schemes $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ via OAEP[†]

Following the generic construction of KEM from ATIF in Fig. 4, now we apply our OAEP[†] transform to the NTRU-based ATIF family $\mathcal{F}_{\text{ntru}}$, and then obtain an NTRU-based PKE scheme KEM_{NTRU} in Fig. 9.

To obtain specific KEM_{NTRU} schemes, we still have to specify the polynomial ring $\mathcal{R}$ and the associated distribution $\psi = (\psi_1, \psi_2)$. We suggest two choices for $(\mathcal{R}, \psi)$ associated with the NTRU assumption, which achieves the NIST post-quantum security level I (corresponding to Kyber-512 [10]). Each choice of $(\mathcal{R}, \psi)$ will determine a specific $\mathcal{F}_{\text{ntru}}$ with its own injective error η . Then we will equip $\mathcal{F}_{\text{ntru}}$ with a corresponding SOTP schemes $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{X}, \mathcal{U}, \mathcal{Y}, \psi_1)$. Accordingly, the specific $\mathcal{F}_{\text{ntru}}$ and SOTP scheme from the two choices of $(\mathcal{R}, \psi)$ yield two types of specific KEM_{NTRU} schemes, namely $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$, which are shown next.

(1) Type I: Scheme $\text{KEM}_{\text{NTRU}}^1$ with $\psi = (\text{CBD}_1^n, \text{CBD}_1^n)$

– **Distribution CBD_1^n and Its Collision Entropy.** Let CBD_1^n be a distribution over $\{-1, 0, 1\}^n$, where each bit has independently identical distribution that takes 0 with probability 1/2, takes 1 with probability 1/4 and takes -1 with probability 1/4.

For any two random variable $\mathbf{a}, \mathbf{b} \leftarrow \text{CBD}_1^n$, it holds that $\Pr[\mathbf{a} = \mathbf{b}] \leq 2^{-n}$. So the collision entropy of CBD_1^n is at least n bit.

– **Parameters ψ and $\mathcal{R}$ for the $\text{NTRU}_{\mathcal{R}, \psi}$ Assumption.** For our first type scheme $\text{KEM}_{\text{NTRU}}^1$, we choose $\psi = (\psi_1, \psi_2) = (\text{CBD}_1^n, \text{CBD}_1^n)$ and $\mathcal{R} = \mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$ with $n = 2^i 3^j$ for the $\text{NTRU}_{\mathcal{R}, \psi}$ assumption. Such an $\mathcal{R}$ admits efficient NTT operation for multiplication, as suggested in [34]. To match the NIST security level I, we need $n \geq 512$. We choose $n = 648 = 2^3 \cdot 3^4$, and then let $q = 7129$ so that the $\text{NTRU}_{\mathcal{R}, \psi}$ problem has quantum hardness of 144 bits according to the security evaluation provided in [8]. Accordingly, we have

$$n = 648, \quad q = 7129, \quad \mathcal{R} = \mathbb{Z}_{7129}/(X^{648} - X^{324} + 1), \quad \psi = (\text{CBD}_1^{648}, \text{CBD}_1^{648}) \quad (4)$$

Encode($\mathbf{m}, \mathbf{r}$) :	Decode($\mathbf{y}, \mathbf{r}$) :
Parse $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \in \{0, 1\}^n \times \{0, 1\}^n$	Parse $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \in \{0, 1\}^n \times \{0, 1\}^n$
$\mathbf{y} := (\mathbf{m} \oplus \mathbf{r}_1) - \mathbf{r}_2$	If $(\mathbf{y} + \mathbf{r}_2) \notin \{0, 1\}^n$: Return $\perp$
Return $\mathbf{y}$	$\mathbf{m} := (\mathbf{y} + \mathbf{r}_2) \oplus \mathbf{r}_1$
	Return m

Fig. 10: Instantiation for $\text{SOTP} = (\text{Encode}, \text{Decode}, \{0, 1\}^n, \{0, 1\}^{2n}, \mathcal{R}, \text{CBD}_1^n)$, where “ $-$ ” denote the bit-wise subtraction over $\mathbb{Z}$.

and the collision entropy of $\psi_2 = \text{CBD}_1^{648}$ is at least 648 bit.

– **Instantiation of $\mathcal{F}_{\text{ntru}}$.** With the above parameter choices for $\mathcal{R}$ and ψ , we obtain a specific ATIF family $\mathcal{F}_{\text{ntru}}$ (cf. Fig. 8 for the construction of $\mathcal{F}_{\text{ntru}}$). Now we can determine the injectivity error η for $\mathcal{F}_{\text{ntru}}$. For any $\mathbf{s} \in \{-1, 0, 1\}^n$, $\mathbf{e} \in \{-1, 0, 1\}^n$ and $\mathbf{c} := \mathbf{h}\mathbf{s} + \mathbf{e}$, if $\mathbf{F}^{-1}$ can recover $\mathbf{e}$ correctly, then $\mathbf{s}$ can also be correctly recovered with $\mathbf{s} := \mathbf{h}^{-1}(\mathbf{c} - \mathbf{e})$. Therefore, for any $\mathbf{s} \in \{-1, 0, 1\}^n$, $\mathbf{e} \in \{-1, 0, 1\}^n$ and $\mathbf{c} := \mathbf{h}\mathbf{s} + \mathbf{e}$, if $(\mathbf{g}, \mathbf{f}') \leftarrow \text{CBD}_1^n$, we have

$$\begin{aligned} \Pr [\mathbf{F}^{-1}(\mathbf{F}(\mathbf{s}, \mathbf{e})) \neq (\mathbf{s}, \mathbf{e})] &\leq \Pr [(3(\mathbf{g}\mathbf{s} + \mathbf{f}'\mathbf{e}) + \mathbf{e} \bmod q) \bmod 3 \neq \mathbf{e}] \\ &\leq \Pr [3(\mathbf{g}\mathbf{s} + \mathbf{f}'\mathbf{e}) + \mathbf{e}|_\infty > (q-1)/2] \leq \Pr [|\mathbf{g}\mathbf{s} + \mathbf{f}'\mathbf{e}|_\infty \geq (q-1)/6]. \end{aligned}$$

Recall that the polynomial multiplication is over $\mathcal{R} = \mathbb{Z}_q/(X^n - X^{n/2} + 1)$. Following the analysis in Sect. 3.2 in [34], the degree- i coefficient of $\mathbf{g}\mathbf{s} + \mathbf{f}'\mathbf{e}$ is the sum of (at worst) n independent random variables $ba + b'(a + a')$, where b and b' are fixed number in $\{-1, 0, 1\}$ from $(\mathbf{s}, \mathbf{e})$, and $a, a' \leftarrow \text{CBD}_1$ from $(\mathbf{g}, \mathbf{f}')$. Since the injectivity error considers the worst case, we consider the degree- i coefficient of $\mathbf{g}\mathbf{s} + \mathbf{f}'\mathbf{e}$ as the sum of n independent random variables $c = |a| + |a + a'|$, where $a, a' \leftarrow \text{CBD}_1$. The random variable c equals to $\{0, 1, 2, 3\}$ with probability $\{2/8, 3/8, 2/8, 1/8\}$, respectively. Let $\rho(i, j)$ be the probability that the sum of i independent random variables is j . Then we have

$$\rho(i, j) = \frac{2}{8}\rho(i-1, j) + \frac{3}{8}\rho(i-1, j-1) + \frac{2}{8}\rho(i-1, j-2) + \frac{1}{8}\rho(i-1, j-3).$$

With initial values $\rho(1, 0), \rho(1, 1), \rho(1, 2)$, and $\rho(1, 3)$, we can compute $\rho(n, j)$ efficiently. Finally, applying union bound to all n coefficients, we have

$$\eta = \Pr [\mathbf{F}^{-1}(\mathbf{F}(\mathbf{s}, \mathbf{e})) \neq (\mathbf{s}, \mathbf{e}) : (\mathbf{F}, \mathbf{F}^{-1}) \leftarrow \text{Gen}] \leq n \cdot \sum_{j=(q-1)/6}^{3n} \rho(n, j).$$

Given $n = 648$ and $q = 7129$, the injectivity error is $\eta \leq 2^{-159}$ for $\mathcal{F}_{\text{ntru}}$.

– **Instantiation of SOTP.** Now we give an instantiation for $\text{SOTP} = (\text{Encode}, \text{Decode}, \mathcal{X} := \{0, 1\}^n, \mathcal{U} := \{0, 1\}^{2n}, \mathcal{Y} := \mathcal{R}, \psi_1 := \text{CBD}_1^n)$ in Fig. 10. *Correctness of SOTP.* For all $\mathbf{m} \in \{0, 1\}^n$ and $\mathbf{r} \in \{0, 1\}^{2n}$, we have

$$\text{Decode}(\text{Encode}(\mathbf{m}, \mathbf{r}), \mathbf{r}) = ((\mathbf{m} \oplus \mathbf{r}_1) - \mathbf{r}_2 + \mathbf{r}_2) \oplus \mathbf{r}_1 = \mathbf{m}.$$

CBD₁ⁿ-Message hiding of SOTP. For any $\mathbf{m} \in \{0, 1\}^n$, if $\mathbf{r}_1, \mathbf{r}_2 \leftarrow_s \{0, 1\}^n$, then $\mathbf{m} \oplus \mathbf{r}_1$ is uniform over $\{0, 1\}^n$ and $\mathbf{y} := (\mathbf{m} \oplus \mathbf{r}_1) - \mathbf{r}_2$ follows the distribution of CBD₁ⁿ.

The above parameter configuration and instantiations of $\mathcal{F}_{\text{ntru}}$ and SOTP yields scheme KEM_{NTRU}¹. According to Theorems 2, 3, 4, we know that KEM_{NTRU}¹ enjoys good securities, as shown in Corollary 1.

Corollary 1. *Given the parameters in Eq.(4) and the above instantiations of $\mathcal{F}_{\text{ntru}}$ and SOTP, the KEM_{NTRU} constructed from $\mathcal{F}_{\text{ntru}}$ via our OAEP[†] transform yields the first type of PKE scheme KEM_{NTRU}¹, which enjoys IND-CCA security, PR-CCA security, ANO-CCA security, and ROB-CCA security.*

(2) Type II of KEM_{NTRU}: Scheme KEM_{NTRU}² with $\psi = (U_3^n, \text{Tern}_d^n)$

– **Distributions U_3^n and Tern_d^n , and the Collision Entropy of Tern_d^n .** Let $\mathcal{S}_3^n := \{\mathbf{a} \mid \mathbf{a} = (a_1, \dots, a_n) \in \{-1, 0, 1\}^n\}$ and U_3^n be the uniform distribution over $\mathcal{S}_3^n$. Let $\mathcal{S}_d^n := \{\mathbf{a} \mid \mathbf{a} = (a_1, \dots, a_n) \in \{-1, 0, 1\}^n, \exists i_1, \dots, i_d, i'_1, \dots, i'_d, s.t. a_{i_1} = \dots = a_{i_d} = 1, a_{i'_1} = \dots = a_{i'_d} = -1, \forall i \in [n] \setminus (\{i_1, \dots, i_d\} \cup \{i'_1, \dots, i'_d\}), a_i = 0\}$. Let Tern_d^n be a uniform distribution over $\mathcal{S}_d^n$. So the collision entropy of Tern_d^n is $\log_2 |\mathcal{S}_d^n| = \log_2 \frac{n!}{d!d!(n-2d)!}$.

– **Parameters ψ and $\mathcal{R}$ for the NTRU _{$\mathcal{R}, \psi$} Assumption.** For our second type scheme KEM_{NTRU}², we choose $\psi = (\psi_1, \psi_2) = (U_3^n, \text{Tern}_d^n)$, and choose $n = 677$, $q = 2048$ and $\mathcal{R} = \mathbb{Z}_{2048}/(X^{677} - 1)$, which share the same $(n, q, \mathcal{R})$ as those of the NTRU-HPS-2048677 in [18]. Such a parameter configuration yields quantum hardness of 170 bit according to the security evaluation provided in [8]. Accordingly, we have

$$n = 677, \quad q = 2048, \quad \mathcal{R} = \mathbb{Z}_{2048}/(X^{677} - 1), \quad \psi = (U_3^{677}, \text{Tern}_{127}^{677}), \quad (5)$$

and the collision entropy of $\psi_2 = \text{Tern}_{127}^{677}$ is $\log_2 \frac{677!}{127!127!(677-254)!} > 700$ bits. The $\mathcal{F}_{\text{NTRU}}$ equipped with the above parameters becomes the deterministic PKE scheme used in NTRU-HPS-2048677 [18], where Tern_d^n was introduced to avoid correctness error of NTRU-HPS-2048677.

– **Instantiation of $\mathcal{F}_{\text{ntru}}$.** With the above parameter choices for $\mathcal{R}$ and ψ , we obtain a specific ATIF family $\mathcal{F}_{\text{ntru}}$ (cf. Fig.8 for the construction of $\mathcal{F}_{\text{ntru}}$). Now we can determine the injectivity error η for $\mathcal{F}_{\text{ntru}}$. By [18], NTRU-HPS-2048677 has perfect correctness. Therefore, $\mathcal{F}_{\text{ntru}}$ has no injectivity error either, so $\eta = 0$.

– **Instantiation of SOTP.** Now we give an instantiation for SOTP = (Encode, Decode, $\mathcal{X} := \{-1, 0, 1\}^n, \mathcal{U} := \{-1, 0, 1\}^n, \mathcal{Y} := \mathcal{R}, \psi_1 := U_3^n$), which couples the above NTRU _{$\mathcal{R}, \psi$} . Here Encode($\mathbf{m}, \mathbf{r}$) outputs $\mathbf{y} := (\mathbf{m} + \mathbf{r}) \bmod 3$, and Decode($\mathbf{y}, \mathbf{r}$) outputs $\mathbf{m} := (\mathbf{y} - \mathbf{r}) \bmod 3$. Clearly, Decode(Encode($\mathbf{m}, \mathbf{r}$), $\mathbf{r}$) = $((\mathbf{m} + \mathbf{r}) - \mathbf{r}) \bmod 3 = \mathbf{m}$ for all $\mathbf{m} \in \mathcal{X}$ and $\mathbf{r} \in \mathcal{U}$. So SOTP has correctness. Meanwhile, the distribution $(\mathbf{m} + \mathbf{r}) \bmod 3$ for $\mathbf{r} \leftarrow_s \{-1, 0, 1\}^n$ is exactly U_3^n , so SOTP is U_3^n -Message hiding.

The above parameter configuration and instantiations of $\mathcal{F}_{\text{ntru}}$ and SOTP yields scheme KEM_{NTRU}². According to Theorems 2, 3, 4, we know that KEM_{NTRU}² enjoys good securities, as shown in Corollary 2.

Corollary 2. *Given the parameters in Eq.(5) and the above instantiations of $\mathcal{F}_{\text{ntru}}$ and SOTP, the KEM_{NTRU} constructed from $\mathcal{F}_{\text{ntru}}$ via our $\text{OAEP}^\dagger$ transform yields the second type of PKE scheme $\text{KEM}_{\text{NTRU}}^2$, which enjoys IND-CCA security, PR-CCA security, ANO-CCA security, and ROB-CCA security.*

The implementations for the PKE schemes $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ are provided at https://github.com/LVYOUyw/OAEP_WORKSPACE. The implementation detail of NTT (for our $\text{KEM}_{\text{NTRU}}^1$ scheme) is shown in Appendix A.

Remark 1. In our implementation, we put the hashed pk (say $G'(pk)$) rather than pk into sk to save space for sk . This is a common trick which is used in Kyber and NTRU+. We require G' is collision-resistant. This trick does not affect the security proofs for IND-CCA, PR-CCA, ROB-CCA.

4.3 Instantiations of ATIF from other post-quantum assumptions

Besides NTRU, there are also other ATIF instantiations from isogenies and from lattices.

- **ATIF from isogenies.** The isogeny-based one-way injective trapdoor function F underlying scheme FESTA [11] satisfies rigidity and partial one-wayness, so it is a natural candidate for ATIF. Our $\text{OAEP}^\dagger$ can convert this F to an isogeny-based CCA-secure PKE/KEM scheme.
- **ATIF from lattices.** The (Ring) LWE assumption from lattices directly results in an ATIF. Let $(\mathbf{A}, \mathbf{T}) \leftarrow \text{TrapGen}$ be the lattice trapdoor generation algorithm in [36]. Then the function $F(\mathbf{s}, \mathbf{e}) := \mathbf{A}^T \mathbf{s} + \mathbf{e}$ is a one-way injective trapdoor function with trapdoor $\mathbf{T}$. This F has rigidity and partial one-wayness, and hence is also a natural instantiation of ATIF based on LWE from lattices. Our $\text{OAEP}^\dagger$ can convert this F to a lattice-based CCA-secure PKE/KEM scheme.

With the above instantiations of ATIF, our $\text{OAEP}^\dagger$ yields two CCA-secure PKE/KEM schemes from isogenies and from lattices respectively. However, they are not as efficient as the CCA-secure PKE/KEM schemes $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ from $\mathcal{F}_{\text{ntru}}$ in subsect. 4.2, so here we do not elaborate them any more. Nevertheless, the various instantiations from different types of post-quantum assumptions show the generality and good applications of our $\text{OAEP}^\dagger$. The future research might exploit more efficient PQC instantiations based on our $\text{OAEP}^\dagger$ transform.

Acknowledgements. We would like to thank the reviewers for their valuable comments. This work was supported by National Natural Science Foundation of China (Grant Nos. 625B2114, 62572295, 62632012, 62522212), National Cryptologic Science Fund of China (Grant No. 2025NCSF01009), National Cyber Security-National Science and Technology Major Project (Grant No. 2026ZD1500900), and National Key R&D Program of China (2025YFA1017100).

References

1. Apple, <https://support.apple.com/guide/security/how-imessage-sends-and-receives-messages-sec70e68c949/1/web/1>
2. Aws cloudhsm, https://docs.aws.amazon.com/cloudhsm/latest/userguide/key_mgmt_util-wrapKey.html
3. Aws-kms, <https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-get-public-key-and-token.html>
4. draft-fluhrer-cfrg-ntru-02, <https://www.ietf.org/archive/id/draft-fluhrer-cfrg-ntru-02.html>
5. Kpqc-bulletin (2024), <https://groups.google.com/g/kpqc-bulletin/c/LC9IpjasXac>
6. Kpqc competition (2025), https://www.kpqc.or.kr/competition_02.html
7. Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Berlin, Heidelberg (Feb 2010). https://doi.org/10.1007/978-3-642-11799-2_28
8. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. J. Math. Cryptol. **9**(3), 169–203 (2015), <http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml>
9. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 269–295. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26951-7_10
10. Avanzi, R., Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-kyber algorithm specifications and supporting documentation <https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf>
11. Basso, A., Maino, L., Pope, G.: FESTA: Fast encryption from supersingular torsion attacks. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 98–126. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8739-9_4
12. Bellare, M., Boldyreva, A., Desai, A., Pointcheval, D.: Key-privacy in public-key encryption. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 566–582. Springer, Berlin, Heidelberg (Dec 2001). https://doi.org/10.1007/3-540-45682-1_33
13. Bellare, M., Rogaway, P.: Optimal asymmetric encryption. In: De Santis, A. (ed.) EUROCRYPT’94. LNCS, vol. 950, pp. 92–111. Springer, Berlin, Heidelberg (May 1995). <https://doi.org/10.1007/BFb0053428>
14. Bernstein, D.J., Cottaar, J., Gandomenico, E.D., Hövelmanns, K., Hülsing, A., Kudinov, M., Lange, T., Mahzoun, M., Meijers, M., Pellegrini, A., Ravagnani, A., Ritsch, S., Schäge, S., Tang, T., Trimoska, M., Vorstermans, M., Weber, F.J.: Report on evaluation of KpqC round-2 candidates. Cryptology ePrint Archive, Paper 2024/2077 (2024), <https://eprint.iacr.org/2024/2077>
15. Bernstein, D.J., Persichetti, E.: Towards KEM unification. Cryptology ePrint Archive, Report 2018/526 (2018), <https://eprint.iacr.org/2018/526>
16. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 61–90. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-36033-7_3

17. Bjørstad, T.E., Dent, A.W., Smart, N.P.: Efficient kems with partial message recovery. In: Galbraith, S.D. (ed.) *Cryptography and Coding, 11th IMA International Conference, Cirencester, UK, December 18-20, 2007, Proceedings. Lecture Notes in Computer Science*, vol. 4887, pp. 233–256. Springer (2007). https://doi.org/10.1007/978-3-540-77272-9_15, https://doi.org/10.1007/978-3-540-77272-9_15
18. Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J.M., Schwabe, P., Whyte, W., Zhang, Z.: *Algorithm specifications and supporting documentation*. Brown University and Onboard security company, Wilmington USA (2019), <https://ntru.org>.
19. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Online-extractability in the quantum random-oracle model. In: Dunkelman, O., Dziembowski, S. (eds.) *EUROCRYPT 2022, Part III. LNCS*, vol. 13277, pp. 677–706. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_24
20. Duman, J., Hövelmanns, K., Kiltz, E., Lyubashevsky, V., Seiler, G., Unruh, D.: A thorough treatment of highly-efficient NTRU instantiations. In: Boldyreva, A., Kolesnikov, V. (eds.) *PKC 2023, Part I. LNCS*, vol. 13940, pp. 65–94. Springer, Cham (May 2023). https://doi.org/10.1007/978-3-031-31368-4_3
21. Ebrahimi, E.: Post-quantum security of plain OAEP transform. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) *PKC 2022, Part I. LNCS*, vol. 13177, pp. 34–51. Springer, Cham (Mar 2022). https://doi.org/10.1007/978-3-030-97121-2_2
22. Ebrahimi, E.: From plaintext-extractability to IND-CCA security. In: Smith, B., Wu, H. (eds.) *SAC 2022. LNCS*, vol. 13742, pp. 419–436. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-58411-4_19
23. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. In: Wiener, M.J. (ed.) *CRYPTO’99. LNCS*, vol. 1666, pp. 537–554. Springer, Berlin, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1_34
24. Gaborit, P., Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Persichetti, E., Zémor, G., Bos, J., et al.: *Hamming quasi-cyclic (hqc) (2025)*
25. Ge, J., Shan, T., Xue, R.: Tighter QCCA-secure key encapsulation mechanism with explicit rejection in the quantum random oracle model. In: Handschuh, H., Lysyanskaya, A. (eds.) *CRYPTO 2023, Part V. LNCS*, vol. 14085, pp. 292–324. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38554-4_10
26. Grubbs, P., Maram, V., Paterson, K.G.: Anonymous, robust post-quantum public key encryption. In: Dunkelman, O., Dziembowski, S. (eds.) *EUROCRYPT 2022, Part III. LNCS*, vol. 13277, pp. 402–432. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_15
27. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. Lecture Notes in Computer Science*, vol. 1423, pp. 267–288. Springer (1998). <https://doi.org/10.1007/BFB0054868>, <https://doi.org/10.1007/BFB0054868>
28. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) *TCC 2017, Part I. LNCS*, vol. 10677, pp. 341–371. Springer, Cham (Nov 2017). https://doi.org/10.1007/978-3-319-70500-2_12
29. Hövelmanns, K., Hülsing, A., Majenz, C.: Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform. In: Agrawal, S., Lin, D. (eds.) *ASI-*

- ACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 414–443. Springer, Cham (Dec 2022). https://doi.org/10.1007/978-3-031-22972-5_15
30. Jiang, H., Zhang, Z., Chen, L., Wang, H., Ma, Z.: IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 96–125. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96878-0_4
 31. Kim, J., Park, J.H.: Ntru+: Compact construction of ntru using simple encoding method (2022), <https://sites.google.com/view/ntruplus>
 32. Kim, J., Park, J.H.: NTRU+: Compact construction of NTRU using simple encoding method. Cryptology ePrint Archive, Report 2022/1664 (2022), <https://eprint.iacr.org/2022/1664>
 33. Liao, H., Ge, J., Cao, S., Xue, R.: New proof for plain OAEP: Post-quantum security without parameter restrictions or collision-resistance. Cryptology ePrint Archive, Report 2025/1288 (2025), <https://eprint.iacr.org/2025/1288>
 34. Lyubashevsky, V., Seiler, G.: NTTTRU: Truly fast NTRU using NTT. IACR TCHES **2019**(3), 180–201 (2019). <https://doi.org/10.13154/tches.v2019.i3.180-201>, <https://tches.iacr.org/index.php/TCHES/article/view/8293>
 35. Maram, V., Xagawa, K.: Post-quantum anonymity of Kyber. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part I. LNCS, vol. 13940, pp. 3–35. Springer, Cham (May 2023). https://doi.org/10.1007/978-3-031-31368-4_1
 36. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_41
 37. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 520–551. Springer, Cham (Apr / May 2018). https://doi.org/10.1007/978-3-319-78372-7_17
 38. Shoup, V.: OAEP reconsidered. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 239–259. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_15
 39. Targhi, E.E., Unruh, D.: Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 192–216. Springer, Berlin, Heidelberg (Oct / Nov 2016). https://doi.org/10.1007/978-3-662-53644-5_8
 40. Ueno, R., Xagawa, K., Tanaka, Y., Ito, A., Takahashi, J., Homma, N.: Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs. IACR TCHES **2022**(1), 296–322 (2022). <https://doi.org/10.46586/tches.v2022.i1.296-322>
 41. Unruh, D.: Non-interactive zero-knowledge proofs in the quantum random oracle model. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 755–784. Springer, Berlin, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6_25
 42. Xagawa, K.: Anonymity of NIST PQC round 3 KEMs. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 551–581. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_20
 43. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Inf. Comput. **15**(7&8), 557–567 (2015). <https://doi.org/10.26421/QIC15.7-8-2>, <https://doi.org/10.26421/QIC15.7-8-2>

Appendix

A Implementation of NTT for $\text{KEM}_{\text{NTRU}}^1$.

Table 3: Combination of NTT Layers

n	q	Radix-2 for cyclotomic trinomial	Radix-3	Radix-2	Inertia degree
648	7129	1	3	2	3

We utilize NTT to optimize the polynomial multiplication within the ring $\mathcal{R} = \mathbb{Z}_q[X]/(X^n - X^{n/2} + 1)$, where $n = 2^i 3^j$. Our approach builds upon the Radix-2 NTT layer for cyclotomic trinomial, introduced in [34], which demonstrates an efficient direct product decomposition of $\mathcal{R}$ into $\mathbb{Z}_q[X]/(X^{n/2} - \eta) \times \mathbb{Z}_q[X]/(X^{n/2} - (1 - \eta))$, where η denotes a sixth primitive root of unity modulo q . This layer reduces the problem to the decomposition of two separate quotient rings $\mathbb{Z}_q[X]/(X^{n/2} - \eta)$ and $\mathbb{Z}_q[X]/(X^{n/2} - (1 - \eta))$, which can be further processed using ordinary Radix-3 NTT layers and Radix-2 NTT layers in sequence, until the rings are decomposed into direct products of fields. At this point, we aim for an inertia degree of 2 or 3, minimizing the cost of multiplication in the NTT domain.

For our implementation, we adapt the NTT framework from [32] to suit our polynomial ring $\mathcal{R}$. The main differences lie in the number of different NTT layers and the final inertia degree, which are shown in Table.3.

Table of Contents

OAEP [†] Transform in the Post-Quantum World	1
<i>You Lyu^{id}, Shengli Liu^{(✉)id}, Shuai Han^{id}, and Bohang Chen^{id}</i>	
1 Introduction	1
2 Preliminary	10
2.1 (General) KEM and Its Security Notions	11
2.2 Lemmas for Quantum Random Oracles	11
3 The OAEP [†] Transform	14
3.1 ATIF and SOTP	15
3.2 OAEP [†] : KEM from ATIF and SOTP	16
4 Instantiations of KEM via OAEP [†]	25
4.1 Instantiations of ATIF from NTRU	25
4.2 Schemes $\text{KEM}_{\text{NTRU}}^1$ and $\text{KEM}_{\text{NTRU}}^2$ via OAEP [†]	27
4.3 Instantiations of ATIF from other post-quantum assumptions ...	30
A Implementation of NTT for $\text{KEM}_{\text{NTRU}}^1$	34