Selective Opening Security in the Quantum Random Oracle Model, Revisited*

Jiaxin Pan [1,2] Runzhi Zeng [2]

October 30, 2023

[1] University of Kassel, Kassel, Germany
jiaxin.pan@uni-kassel.de
[2] Department of Mathematical Sciences,
NTNU Norwegian University of Science and Technology, Trondheim, Norway
runzhi.zeng@ntnu.no

Abstract

We prove that two variants of the Fujisaki-Okamoto (FO) transformations are selective opening secure (SO) against chosen-ciphertext attacks in the quantum random oracle model (QROM), assuming that the underlying public-key encryption scheme is one-way secure against chosen-plaintext attacks (OW-CPA). The two variants we consider are $\mathsf{FO}^{\not\perp}$ (Hofheinz, Hövelmanns, and Kiltz, TCC 2017) and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ (Jiang et al., CRYPTO 2018). This is the *first* correct proof in the QROM.

The previous work of Sato and Shikata (IMACC 2019) showed the SO security of $\mathsf{FO}^{\not\perp}$ in the QROM. However, we identify a subtle gap in their work. To close this gap, we propose a new framework that allows us to adaptively reprogram a QRO with respect to multiple queries that are computationally hard to predict. This is a property that can be easily achieved by the classical ROM, but is very hard to achieve in the QROM. Hence, our framework brings the QROM closer to the classical ROM.

Under our new framework, we construct the *first tightly* SO secure PKE in the QROM using lossy encryption. Our final application is proving $\mathsf{FO}^{\not\perp}$ and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ are bi-selective opening (Bi-SO) secure in the QROM. This is a stronger SO security notion, where an adversary can additionally corrupt some users' secret keys.

Keywords: Selective opening security, quantum random oracle model, Fujisaki-Okamoto transformation, tight security

1 Introduction

Public-key encryption (PKE) schemes are a central topic in cryptography. Their widely accepted security notion is indstinguishability against chosen-ciphertext attacks (IND-CCA), which states that confidentiality holds even if an adversary $\mathcal{A}$ can adaptively decrypt ciphertexts of its choice, except the challenge ciphertext. This is a security notion in the single-user, single-challenge setting, namely, only one user's public key and one challenge ciphertext are exposed to an adversary.

Its multi-user, multi-challenge counterpart is an arguably more realistic setting. Selective opening (SO) security [BHY09, BHK12] is a notion in a multi-challenge setting, where an adversary is given multiple challenge ciphertexts under a single public key and aims at learning some information about the encrypted messages. On top of that, the adversary can open a subset of the challenge ciphertexts and reveal the corresponding messages and randomness used to generate those ciphertexts. SO security guarantees the confidentiality of the remaining unopened challenge ciphertexts. The recent notion, Bi-SO security [LYHW21], can be viewed as a stronger variant of the SO security in a multi-user setting, where the adversary is additionally given multiple users' public keys and it can corrupt some of their secret keys.

*Partially supported by the Research Council of Norway under Project No. 324235

The aforementioned opening capability is motivated by the fact that cryptographic information is technically hard and expensive to erase in practice and an adversary may break into an encrypter's computer and learn the used randomness. In some applications, such as secure multi-party computation, it is even required to reveal the messages and randomness to make a user's computation publicly verifiable.

Technically speaking, it is challenging to construct a SO secure PKE. At a first glance, one may think that IND-CCA security implies SO security, since each ciphertext is generated using independent randomness. However, this is not true in general. We refer [HR14] for an overview and useful further reading. We highlight that, from a provable-security point of view, to answer an opening query, a security reduction should be able to 'explain' how it generates a challenge ciphertext by returning the randomness, but in many cases the reduction do not even know the randomness itself. Hybrid arguments are one of the examples, namely, the reduction cannot explain a ciphertext where a challenge is embedded. This is also the inherent reason why the recent updated proof of Sato and Shikata [SS22] is incorrect. In the recent years, a great amount of effort has been put into defining the right notion of SO security [BHY09, BHK12, HR14] and construct efficient SO-secure public-key encryption schemes [FHKW10, HLOV11, Hof12, HJKS15, HP16, LYHW21].

Notions of Selective Opening Security. Currently, there are two types of notions have been studied in the literature, the indistinguishability-based (IND-based) ones (weak-IND-SO and full-IND-SO) [BHY09, BHK12] and the simulation-based (SIM-based) one (SIM-SO) [BHY09]. They are not polynomial-time equivalent to each other. In this paper we only consider the SIM-based one. Informally, SIM-SO security states that for every SO adversary its output can be efficiently simulated by a simulator that sees only the opened messages. Unlike its IND-based counterpart, SIM-SO does not require the message distribution chosen by the adversary to be efficiently resamplable, conditioned on the opened messages (cf. [BHY09]). Previous work showed that SIM-SO-CCA and full-IND-SO-CCA notions are the strongest SO security [BHK12, BDWY12, HR14]. However, only SIM-SO-CCA has been realized so far [FHKW10, HLOV11, Hof12, HJKS15, HP16]. It is similar for Bi-SO security, and only SIM-based notion is considered so far [LYHW21]. For simplicity, we will not write 'SIM' in the following.

Our Goal: Selective Opening Security in the QROM. SO secure PKE schemes are constructed in idealized models [HJKS15, HP16] and in the standard model [BHY09, FHKW10, HLOV11, Hof12]. Constructions in idealized models are more efficient and hence more relevant to practice. In particular, this paper considers schemes in the random oracle model (ROM).

The increasingly threat that quantum computers can break most widely deployed public-key cryptosystems has driven research in the direction of building post-quantum secure public-key primitives, including PKE schemes and key encapsulation mechanisms (KEMs). Currently, the National Institute of Standards and Technology (NIST) in the US has come to a conclusion for the post-quantum standards. Kyber [SAB+20], NTRU [CDH+20], and Saber [DKR+20] were three finalists in the last round for the KEM/PKE category. They all use variants of the Fujisak-Okamoto (FO) transformation [FO99a, FO99b, FO13, HHK17]. It is interesting to consider whether these FO transformations are secure in the SO setting.

The FO transformation turns a relatively weak PKE (e.g. a One-Way CPA secure one) into an IND-CCA secure one. Recently, the FO transformation and its variants have been widely analyzed in both the classical ROM and the quantum (accessible) ROM (QROM) [TU16, HHK17, SXY18, JZC+18, KSS+20], but mostly with a focus on establishing IND-CCA security. An exception is the work of Heuer et al.[HJKS15] which studied the SO security of the FO transformation in the ROM.

For post-quantum security, proofs in the QROM are more desirable than those in the (classical) ROM, since it models quantum adversaries in a more realistic manner. In this setting, a quantum adversary interacts with a classical network, where "online" primitives (such as encryption) are classical, and computes "offline" primitives (such as hashing) on its own in superposition.

The work of Sato and Shikata [SS19] proved the SO security of the FO transformation in the QROM. To the best of our knowledge, this is the only work considers SO security in the QROM. However, we identified a subtle gap in their security proof[1]. Even worse, this gap cannot be closed, even if we relax the notion to the weaker, non-adaptive SO security as in [LLHG18], where an adversary is not allowed to adaptively open a challenge ciphertext, but commits all its opening indices after seeing the challenge ciphertexts. From a technical point of view, closing the gap in [SS19] requires new proof

[1] The authors confirmed this to us.

techniques in the QROM that allow a security reduction to adaptively reprogram multiple RO-queries in one security game without changing the view of an adversary, where the reprogrammed points are computationally hidden. This is a property not achievable by existing well-known techniques, such as [Unr14b, Unr14a, KSS+20, GHHM21]. We provide more discussion about it in Section 1.2.

1.1 Our Contributions

We revise the selective opening security in the QROM and prove that two "implicit rejection" variants of the FO transformation (namely, $\mathsf{FO}^{\not\perp}$ [JZC+18] and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ [HHK17]) are SO-CCA secure if the underlying PKE is one-way CPA (OW-CPA) secure in the QROM. Here we consider PKE schemes, namely, combining KEM $\mathsf{FO}^{\not\perp}$ (or $\mathsf{U}^{\not\perp}_{\mathsf{m}}$) with one-time pad and a message authentication code (MAC). The one with $\mathsf{FO}^{\not\perp}$ is the same scheme considered in [SS19], but ours is the first correct proof in the QROM. Since the proofs for $\mathsf{FO}^{\not\perp}$ and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ are similar, we leave the one for $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ in Supp. Mat. H, and there we only prove the Bi-SO-CCA for $\mathsf{U}^{\not\perp}_{\mathsf{m}}$, since it implies SO-CCA security.

Our core technical contribution is a computational adaptive reprogramming framework in the QROM that enables a security reduction to *adaptively* and *simultaneously* reprogram polynomially many RO-queries which are computationally hidden from a quantum adversary. This is a property cannot be provided by previous techniques in the QROM, such as the (adaptive) one-way to hiding (O2H) lemma [Unr14b, Unr14a], the semi-classical O2H lemma [AHU19], and the measure-rewind-measure O2H lemma [KSS+20]. Our framework brings the QROM closer to the classical ROM, and it generalizes and improves the adaptive reprogramming framework by Grilo et al.[GHHM21].

Tight SO Security from Lossy Encryption in the QROM. Our second contribution is a tightly SO-CCA secure PKE from lossy encryption [BHY09, HJR16]. This is the *first* tight scheme in the QROM. A recent work of Pan, Wagner, and Zeng has constructed the first tightly multi-user (without corruptions), multi-challenge IND-CCA in the QROM [PWZ23], but it did not get extended to the (stronger) SO setting. Another related work is also due to Pan and Zeng [PZ22], where a compact and tightly SO-CCA secure PKE is proposed in the classical random oracle model. However, it is unclear if it can be transformed to the QROM. Our result on tight SO security is established in the QROM, and it improves both aforementioned work.

Bi-SO Security of FO Transformations. As another application of our framework, we prove that the aforementioned variants of FO transformation, namely, $\mathsf{FO}^{\not\perp}$ and $\mathsf{U}^{\not\perp}_{\mathsf{m}}$, are furthermore Bi-SO-CCA secure [LYHW21] in the QROM, assuming OW-CPA security of the underlying PKE scheme. This notion is stronger than the SO-CCA security, since it additionally allows secret key corruption for the adversaries. The only known Bi-SO-CCA secure construction is in the classical ROM. Our work is the first one in the QROM.

Impacts on the NIST Finalists. The NIST finalists Kyber and Saber use tweaked verions of transformation $\mathsf{FO}^{\not\perp}$, and NTRU uses $\mathsf{U}^{\not\perp}_{\mathsf{m}}$. Hence, analysis of these FO transformations is more fundamental than directly analyzing these concrete schemes. Although our results strongly indicate that the NIST finalists are SO-CCA secure and Bi-SO-CCA in the QROM, we leave the formal proof of it as a future direction, and we are optimistic that our approaches can be extended naturally in achieving it.

1.2 Technical Details

We provide some details about our technical contribution, computational adaptive reprogramming framework.

Our Starting Point. The work of Heuer et al.[HJKS15] is the first one proving that practical PKEs via the OAEP and FO transformation are SO-CCA secure in the (classical) ROM. Their work considered the original FO transformation [FO13]. Motivated by Heuer et al.'s work, we can show that the combination of $\mathsf{FO}^{\not\perp}$ and one-time pad is SO-CPA secure in the classical ROM by adaptively reprogramming the ROs. Here we describe some key idea. Note that our final goal is SO-CCA, but for the simplicity of our discussion here, we only consider SO-CPA.

A ciphertext of message m in the $\mathsf{FO}^{\not\perp}$ transformation, (e, d), is defined as follow:

$$\begin{aligned} e &:= \mathsf{Enc}_0(\mathsf{pk}, r; G(r)) \quad \text{for} \quad r \xleftarrow{\$} \mathcal{M}' \\ d &:= H(r, e) \oplus m \end{aligned} \tag{1}$$

where Enc_0 is the randomized encryption algorithm of a OW-CPA secure PKE with message space $\mathcal{M}'$, $G(r)$ is the explicit randomness used in Enc_0, and G, H are two hash functions with suitable domains and ranges. Public and secret keys of $\mathsf{FO}^{\not\perp}$ is the same as those of the OW-CPA secure PKE, and the decryption is defined in the straightforward way. We refer Figure 6 for the full description.

Efficient Openability in the ROM. To show the SO-CPA security, we require "efficient openability" of ciphertexts [BHY09, FHKW10]. This property states that one can generate some ciphertexts and later they can be efficiently opened to arbitrary messages by using some trapdoor (in the standard model) or reprogramming ROs (in the ROM) in a suitable way. In the classical ROM, our ciphertexts (defined by Equation (1)) have efficient openability. More precisely, a security reduction $\mathcal{R}$ can choose random r_i^*, R_i^*, and d_i^* and return the challenge ciphertexts $(\mathsf{Enc}_0(\mathsf{pk}, r_i^*; R_i^*), d_i^*)_{1\le i\le \mu}$ to the SO-CPA adversary $\mathcal{A}$. For these challenge ciphertexts, the reduction $\mathcal{R}$ can open a ciphertext $(\mathsf{Enc}_0(\mathsf{pk}, r_i^*; R_i^*), d_i^*)$ to arbitrary message m_i by reprogramming $G(r_i^*) := R_i^*$ and $H(r_i^*, e_i^*) := d_i^* \oplus m_i$. Moreover, $\mathcal{R}$ will embed the OW-CPA challenge to one of the unopened ciphertexts. Here, r_i^* are only computationally hidden from the adversary.

For the SO-CPA security, the aforementioned reprogramming is required to be *adaptive*, since an adversary can submit an opening query adaptively. Moreover, a SO-CPA adversary can submit multiple opening queries in one security game or hybrid. Therefore, our reprogramming strategy should be able to reprogram multiple RO-queries in one security game. We call this last requirement as multi-point reprogramming. We stress that hybrid arguments are already not useful for SO security. This is because a standard hybrid argument will embed a OW-CPA challenge into the SO-CPA ciphertexts one-by-one. After it is embedded to the i-th ciphertext, $G(r_i^*)$ cannot be reprogrammed to R_i^*, since R_i^* is unknown to the reduction $\mathcal{R}$. Thus, the opening query cannot be correctly answered.

Existing Approaches in the QROM. Reprogramming a quantum (accessible) RO is highly non-trivial, since a query in superposition can be viewed as a query that might contain all possible input values at once. To correctly reprogram a value to a particular QRO query, it needs to measure and extract classical preimages of a quantum query, which will cause a change in the adversary's view. Although many works have been done to provide reprogrammability in the QROM [Unr14a, Unr14b, AHU19, KSS+20, GHHM21], reprogramming in the QROM is still much more challenging than in the ROM.

For the SO security, the situation is more complicated. Essentially, existing approaches (such as [Unr14a, Unr14b, AHU19, KSS+20, GHHM21]) cannot easily achieve the requirements for SO security in the QROM. We use the semi-classical O2H lemma [AHU19] as an example to elaborate on this. Fix a random set $S \subseteq \mathcal{X}$. Let $H, H' : \mathcal{X} \to \mathcal{Y}$ be two different ROs such that, for all $x \in \mathcal{X} \setminus S$, $H(x) = H'(x)$ (denoted by $H \setminus S = H' \setminus S$). The semi-classical O2H lemma states that a quantum adversary $\mathcal{A}$ cannot tell the difference between H and H' by giving only quantum access to them, unless $\mathcal{A}$ finds an element from S. Here set S needs to be defined before defining H and H'.

In the work of Sato and Shikata [SS19], their security proofs viewed S as the set containing all the randomness used in the opened ciphertexts (cf. the step between Game_1 and Game_2 in [SS19, Section 3.1] and the one between Game_5 and Game_6 in [SS19, Section 3.2]). Essentially, S is equivalent to the set of opening indices which are adaptively decided by the adversary $\mathcal{A}$. However, to use the semi-classical O2H lemma, S must be fixed at the beginning of the security game, even before generating the public key. Therefore, this technical gap in their proofs cannot be closed, and it will be the case, even if we consider the weaker, non-adaptive variant of SO security as in [LLHG18], namely, an adversary cannot adaptively open challenge ciphertexts, but commits to opening indices after receiving the challenge ciphertexts.

The recent measure-rewind-measure O2H lemma [KSS+20] has a similar flavor as the semi-classical O2H lemma, and it does not allow to define S adaptively. The adaptive O2H lemma [Unr14a] allow us to reprogram a single query adaptively. However, we require adaptive reprogramming multiple queries for SO security, since if we only reprogram wrt one opening query, an adversary can distinguish the simulation by opening multiple ciphertexts.

Our Approach. To solve the technical difficulties, we propose the computational adaptive reprogramming framework. It is more general than the algorithmic O2H lemma [Unr14a] and the adaptive reprogramming framework [GHHM21] in the sense that our framework allows a reduction to reprogram polynomial many RO queries in the QROM. Different to the work of Grilo et al., our reprogrammed points can be only computationally hidden from the adversary.

In a nutshell, our framework considers two security games, NonAda and Ada. The RO H' in

NonAda will never be reprogrammed, but the RO H in Ada will be adaptively reprogrammed for multiple times according to the adversary's behavior. We require $H' \setminus S = H \setminus S$, but S can be modified adaptively by a security reduction. Intuitively, an adversary $\mathcal{A}$ can distinguish NonAda and Ada if it queries $x \in S$. This event can be detected easily in the classical setting, but is problematic in the quantum setting. Our high-level approach is to bound the probability of this event by randomly measuring $\mathcal{A}$. Details are given in Section 3. We stress that our approach is not a "hybrid argument" extension of the existing techniques. In fact, as pointed out by Bellare, Hofheinz, and Yilek [BHY09], it is unknown if a simple hybrid argument is useful in proving SO security. Very unfortunately, the latest revision[2] of [SS19] is a concrete example for why it does not work. The proof of their Lemma 1 is essentially a hybrid argument. A counterexample is simply: Imagine an adversary opens each index with probability $1/2$, then their OPEN oracle will abort with overwhelming probability and thus their hybrid argument.

More Comparison with Related Work. Recently, Grilo et al. proposed the adaptive reprogramming framework [GHHM21] and used it to give a QROM proof for Fiat-Shamir's signatures. The main difference between our work and Grilo et al.'s work is that their framework requires the reprogramming points to have high statistical entropy, while our framework requires the reprogramming points are computationally hard to find (which cover the case of statistical entropy). When proving the SO security of the FO transformation, their framework cannot be used since the reprogramming points are computationally hidden by OW-CPA security of some underlying PKEs.

We also compare our framework to the measure-and-reprogram framework of Don, Fehr, and Majenz [DFM20] and the lifting theorem in [YZ21] that are used to prove security of the Fiat-Shamir (FS) signature in the QROM. In a nutshell, the difference between our frameworks is similar to that between the security proofs of the FO encryption and FS signature in the classical setting. More precisely, in the proof of FO encryption, we argue that it is infeasible for an adversary to learn the reprogramming points and thus we can reprogram the random oracle without changing the adversary's view. However, in the proof of FS signature, an adversary can learn the reprogramming points, since they are the hash values of signing messages and some (public) commitments of the Σ protocol. Hence, the measure-and-reprogram framework is conceptually different to us and cannot be used in proving SO or Bi-SO security in the QROM. The lifting theorem (cf. [YZ21, Theorem 4.2]) has a similar flavor as the measure-and-reprogram framework.

Finally, we are aware of a recent revision of the Sato-Shikata work [SS22], but it uses a hybrid argument and, as explained earlier, hybrid arguments are not useful even in the classical ROM. As a simple counter-example, in the proof of their Theorem 1, imagine an adversary that opens each ciphertext with probability $1/2$ (or some non-negligible probability). Then their $\mathsf{Hybrid}^{(i^*)}$ aborts with probability $1/2$.

Future Work. We leave exploring more applications of our computational adaptive reprogramming framework as a future direction, since reprogramming a (quantum) random oracle on multiple computationally hidden points is an interesting technique and we are optimistic that it may yields new applications. Moreover, we are optimistic that our approach can work for the simulatable DEM framework of SO secure PKEs. We leave a formal treatment of it as another future direction.

2 Preliminaries

Let n be an integer. $[n]$ denotes the set $\{1, ..., n\}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets and $f : \mathcal{X} \to \mathcal{Y}$ be a function. $f(\mathcal{X}) := \{f(x)|x \in \mathcal{X}\}$. $x \xleftarrow{\$} \mathcal{X}$ denotes sampling a uniform element x from $\mathcal{X}$ at random. If S is a subset of $\mathcal{X}$, then $\mathcal{X} \backslash S$ denotes the set $\{x \in \mathcal{X}|x \notin S\}$. Let $\mathcal{A}$ be an algorithm. If $\mathcal{A}$ is probabilistic, then $y \leftarrow \mathcal{A}(x)$ means that the variable y is assigned to the output of $\mathcal{A}$ on input x. If $\mathcal{A}$ is deterministic, then we write $y := \mathcal{A}(x)$. We write $\mathcal{A}^{\mathcal{O}}$ to indicate that $\mathcal{A}$ has classical access to oracle $\mathcal{O}$. We write $\mathbf{T}(\mathcal{A}_0) \approx \mathbf{T}(\mathcal{A}_1)$ if the running times of $\mathcal{A}_0$ and $\mathcal{A}_1$ are polynomially close to each other. All (quantum) algorithms are (quantum) probabilistic polynomial time, unless we state it.

Games. We use code-based games [BR06] to define and prove security. We implicitly assume that Boolean flags are initialized to false, numerical types are initialized to 0, sets are initialized to $\emptyset$, while strings are initialized to the empty string ϵ. $\Pr[\mathbf{G}^{\mathcal{A}} \Rightarrow 1]$ denotes the probability that the final output

[2] https://eprint.iacr.org/archive/2022/617/20230108:160413

$\mathbf{G}^{\mathcal{A}}$ of game $\mathbf{G}$ running an adversary $\mathcal{A}$ is 1. Let $\mathtt{Ev}$ be an (classical and well-defined) event. We write $\Pr[\mathtt{Ev} : \mathsf{G}]$ to denote the probability that $\mathtt{Ev}$ occurs during the game $\mathbf{G}$.

2.1 Public-Key Encryption

A Public Key Encryption (PKE) scheme PKE consists of three algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and a message space $\mathcal{M}$ that is assumed to be efficiently recognizable. The three algorithms work as follows:

- The key generation algorithm KG, on input the security parameter λ, outputs a public and secret key pair $(\mathsf{pk}, \mathsf{sk})$. pk also defines a finite randomness space $\mathcal{R} := \mathcal{R}(\mathsf{pk})$ and a ciphertext space $\mathcal{C} := \mathcal{C}(\mathsf{pk})$. For sake of simplicity, in this paper, we ignore the input λ and simply write the process as $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}$.
- The encryption algorithm Enc, on input pk and a message $m \in \mathcal{M}$, outputs a ciphertext $c \in \mathcal{C}$. We also write $c := \mathsf{Enc}(\mathsf{pk}, m; r)$ to indicate the randomness $r \in \mathcal{R}$ explicitly.
- The (deterministic) decryption algorithm Dec, on input sk and a ciphertext c, outputs a message $m' \in \mathcal{M}$ or a rejection symbol $\perp \notin \mathcal{M}$.

Definition 2.1 (PKE Correctness). A PKE scheme $\mathsf{PKE} := (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ is $(1-\delta)$-correct if

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

where the expectation is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}$ and randomness of Enc. PKE has perfect correctness if $\delta = 0$.

Definition 2.2 (Collision Probability of Key Generation). Let

$$\eta_{\mathsf{PKE}} := \max\left[\Pr\left[\mathsf{pk}_0 = \mathsf{pk}_1 : (\mathsf{pk}_0, \mathsf{sk}_0) \leftarrow \mathsf{KG}, (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}\right]\right]$$

be the collision probability of KG of PKE. The maximum is taken over all $\mathsf{pk}_0, \mathsf{pk}_1$. In this paper, we assume that for any OW-CPA-secure PKE, $\eta_{\mathsf{PKE}} = \mathsf{negl}(\lambda)$

We focus on two security notions for PKE: onewayness under chosen-plaintext attacks ($\mathsf{OW\text{-}CPA}$) and selective-opening security under chosen-ciphertext-attacks (SO-CCA). Let $\mathsf{PKE} := (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme with message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$.

Definition 2.3 ($\mathsf{OW\text{-}CPA}$). For an adversary $\mathcal{A}$, its advantage against $\mathsf{OW\text{-}CPA}$ security of PKE is defined as

$$\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) := \Pr\Big[m' = m^* : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}, m^* \xleftarrow{\$} \mathcal{M}, \\ c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*), m' \leftarrow \mathcal{A}(\mathsf{pk}, c^*)\Big].$$

PKE is $\mathsf{OW\text{-}CPA}$ secure if for all PPT adversaries $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \mathsf{negl}(\lambda)$.

We also use MAC schemes that have one-time strong existential unforgeability under chosen message attack (otSUF-CMA) as building block. Let $\mathsf{MAC} := (\mathsf{Tag}, \mathsf{Vrfy})$ be an one-time MAC scheme with key space $\mathcal{K}^{\mathtt{mac}}$. The otSUF-CMA security game is given in Figure 1.

Definition 2.4 ($\mathsf{otSUF\text{-}CMA}$). For a forger $\mathcal{F}$, its advantage against otSUF-CMA security of MAC is defined as

$$\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{PKE}}(\mathcal{F}) := \Pr[\mathsf{otSUF\text{-}CMA}^{\mathcal{F}}_{\mathsf{MAC}} \Rightarrow 1]$$

MAC is otSUF-CMA secure if for all $\mathcal{F}$, $\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{PKE}}(\mathcal{F}) = \mathsf{negl}(\lambda)$.

One-time MAC schemes can be constructed by using pair-wise independent hash function family, and they are otSUF-CMA secure against *unbounded* adversaries. Here Tag cannot be queried with quantum superposition.

(Adaptive) Selective Opening Security. Selective Opening (SO) security preserves confidentiality even if an adversary opens the randomnesses of some ciphertexts. We use simulation-based approach

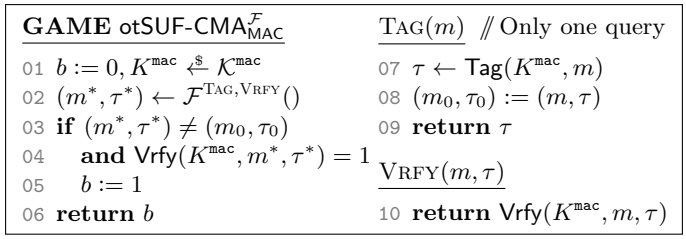

Figure 1: Security games one-time MAC schemes

to define SO security as in [HJKS15]. We consider the SO security against Chosen-Plaintext Attacks (SO-CPA) and Chosen-Ciphertext Attacks (SO-CCA), respectively.

We note that a non-adaptive variant of SO security has been used in [LLHG18], where an adversary must declare the opening index set I after receiving the challenge ciphertexts, while our SO security is *adaptive* in the sense that Open can be asked adaptively. Intuitively, our adaptive security is harder to achieve, since an adversary can change its opening queries after seeing the answers of previous ones.

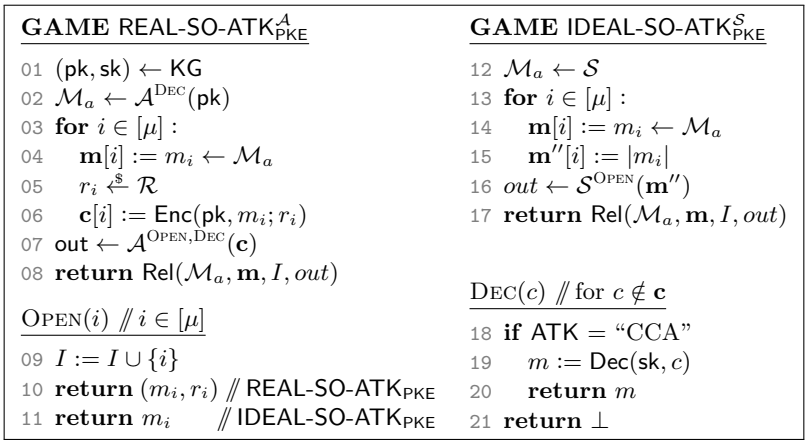

Figure 2: The SO security games for PKE schemes.

Definition 2.5 (SO security). Let PKE be a PKE scheme with message space $\mathcal{M}$ and randomness space $\mathcal{R}$ and $\mathcal{A}$ be an adversary against PKE. For security parameter λ, $\mu := \mu(\lambda) > 0$ is a polynomially bounded function. Let Rel be a relation. We consider two games defined in Figure 2, where $\mathcal{A}$ is run in REAL-SO-ATK$_{\mathsf{PKE}}$ and a SO simulator $\mathcal{S}$ in IDEAL-SO-ATK$_{\mathsf{PKE}}$. $\mathcal{M}_a$ is a distribution over $\mathcal{M}$ chosen by $\mathcal{A}$, and $\mathcal{A}$ is not allowed to issue Open queries before it outputs $\mathcal{M}_a$ and receives challenge ciphertexts $\mathbf{c}$. Messages sampled from $\mathcal{M}_a$ may be dependent on each other. Dec is not available in SO-CPA security.

We define the SO-ATK (ATK = 'CPA' or 'CCA') advantage function

$$\mathsf{Adv}^{\mathsf{SO\text{-}ATK}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) := \left| \Pr\left[\mathsf{REAL\text{-}SO\text{-}ATK}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}SO\text{-}ATK}^{\mathcal{S}}_{\mathsf{PKE}} \Rightarrow 1\right] \right|,$$

PKE is SO-ATK secure if, for every adversary $\mathcal{A}$ and every PPT relation Rel, there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{SO\text{-}ATK}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \leq \mathsf{negl}(\lambda)$.

(Adaptive) Bi-Selective-Opening Security. In this paper, we also consider a stronger SO security definition: Bi-SO-ATK [LYHW21]. This security definition considers a multi-user setting and allows the adversary to corrupt some users (namely, obtains their secret keys) adaptively. The Bi-SO-ATK definition in [LYHW21] is non-adaptive, that is, the SO adversary is required to tell the game simulator which users it wants to corrupted and which challenge ciphertexts it wants to open at once. In this paper, we enhance the security definition to be adaptive. The adversary can adaptively issues Open queries and Corrupt queries in any order. The enhanced definition is also simulation-based. If $\mathcal{A}$ corrupts a

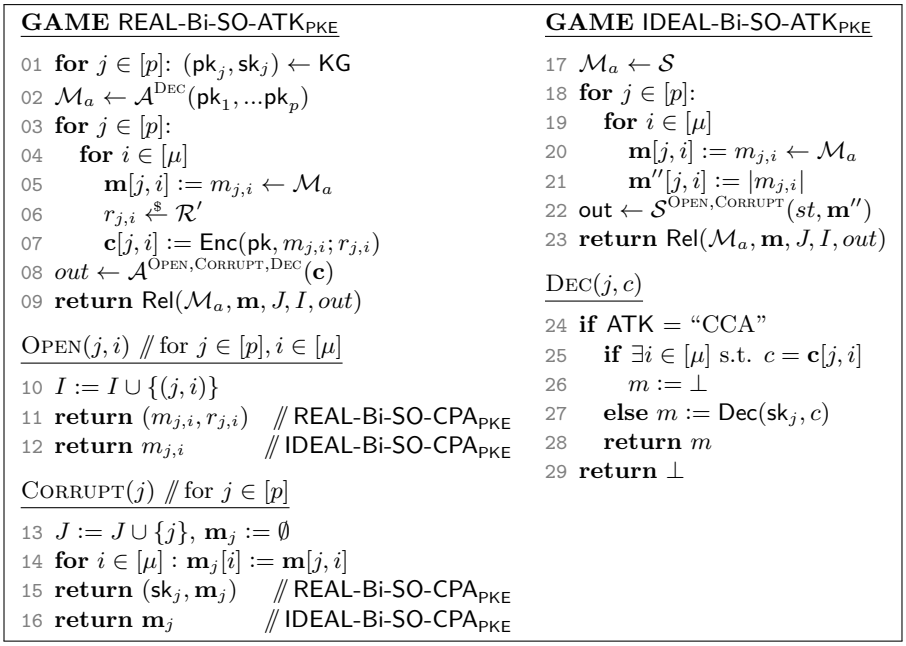

Figure 3: The Bi-SO-ATK security game for PKE schemes

user j, then the messages of challenge ciphertexts that encrypted by j are also revealed (see Items 15 and 16).

Definition 2.6 (Bi-SO security). Let PKE be a PKE scheme and $\mathcal{A}$ be a Bi-SO adversary against PKE. For security parameter λ, let $\mu := \mu(\lambda)$ and $p := p(\lambda)$ that are both polynomially bounded. Let Rel be a relation. We consider two games defined in Figure 3, where $\mathcal{A}$ is run in REAL-Bi-SO-ATK$_{\mathsf{PKE}}$ and a Bi-SO simulator $\mathcal{S}$ in IDEAL-Bi-SO-ATK$_{\mathsf{PKE}}$. $\mathcal{M}_a$ is a distribution over $\mathcal{M}$ chosen by $\mathcal{A}$, and $\mathcal{A}$ is not allowed to issue Open or Corrupt queries before it outputs $\mathcal{M}_a$ and receives challenge ciphertexts $\mathbf{c}$. Messages sampled from $\mathcal{M}_a$ may be dependent on each other. Dec is not available in Bi-SO-CPA security.

We define the Bi-SO-ATK (ATK = 'CPA' or 'CCA') advantage function

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}ATK}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, p, \mu, \mathsf{Rel})\\ &\quad := \left|\Pr\left[\mathsf{REAL\text{-}Bi\text{-}SO\text{-}ATK}^{\mathcal{A}}_{\mathsf{PKE}} \Rightarrow 1\right] - \Pr\left[\mathsf{IDEAL\text{-}Bi\text{-}SO\text{-}ATK}^{\mathcal{S}}_{\mathsf{PKE}} \Rightarrow 1\right]\right|.\end{aligned}$$

PKE is adaptive Bi-SO-ATK secure if, for any adversary $\mathcal{A}$ and PPT relation Rel, there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}ATK}}_{\mathsf{PKE}}(\mathcal{A}, \mathcal{S}, p, \mu, \lambda) = \mathsf{negl}(\lambda)$.

Security in the quantum random oracle model. The (Bi-)SO security of PKE schemes containing hash functions can be analyzed in the quantum random oracle model (cf. Section 2.2). If we model a hash function H as quantum random oracle, then the adversary $\mathcal{A}$ has quantum access to H during the SO security games (e.g., Figure 7).

2.2 Quantum computation

We refer to [NC16] for detailed background about quantum mechanism. Here we only recall some necessary notations and lemmas.

Pure quantum states can be described by qubits. For a λ-bit-string x, $|x\rangle \in \mathbb{C}^{2^\lambda}$ denotes the (pure) quantum state of x encoded in the standard computational basis. Quantum register is used to store multiple qubits. In this paper, we assume that any polynomially long object x can be encoded as a (unique) bit string, and if we "store" x in a quantum register X, $|x\rangle$ is the quantum state of this register. A λ-qubits quantum superposition state $|\phi\rangle$ can be written as $\sum_{x\in\{0,1\}^\lambda} \alpha_x|x\rangle$ where $\sum_{x\in\{0,1\}^\lambda} |\alpha_x|^2 = 1$.

By performing measurement on a quantum state, we obtain classical information about the state, and the state collapses after measurement. Let $|x\rangle$ be an quantum state, $x' \leftarrow \mathsf{Measure}(|x\rangle)$ denote the

process that $|x\rangle$ is measured and the measurement outcome is x'. We assume that all measurement are performed with respect to the standard computational basis.

Let $\mathcal{O}:\mathcal{X}\to\mathcal{Y}$ be an random oracle with sets $\mathcal{X},\mathcal{Y}$. We implicitly assume that the elements in $\mathcal{X}$ and $\mathcal{Y}$ are expressed as bit strings. In quantum random oracle model (QROM)[BDF+11], the oracle $\mathcal{O}$ are described as the unitary transformation $U_{\mathcal{O}}:|x\rangle|y\rangle\to|x,y\oplus\mathcal{O}(x)\rangle$, and the adversary can query random oracles on quantum states. For an quantum adversary $\mathcal{A}$, the notation $\mathcal{A}^{|\mathcal{O}\rangle}$ indicates that $\mathcal{A}$ has quantum access to the $U_{\mathcal{O}}$. Without loss of generality, we directly write $\mathcal{O}$ to denote the unitary $U_{\mathcal{O}}$.

In this paper, we say an event is classical if it can be determined by only using classical algorithm (namely, without using any quantum mechanism).

Lemma 2.7 gives a probabilistic bound for adversary (has a quantum access to oracles) to distinguish $h(s,\cdot)$ and h', where s is secret, h and h' are QRO and have the same image set. When the image is large enough, the adversary cannot distinguish these two oracles.

Lemma 2.7 (Lemma 2.2 in [SXY18]). *Let k be an integer. Let $h:\mathcal{X}'\times\mathcal{X}\to\mathcal{Y}$ and $h':\mathcal{X}\to\mathcal{Y}$ be two independent random oracles. If an unbounded time quantum adversary $\mathcal{A}$ that queries h at most q_h times, then we have*

$$\left|\Pr\left[1\leftarrow\mathcal{A}^{|h\rangle,|h(s,\cdot)\rangle}()\middle|s\xleftarrow{\$}\mathcal{X}'\right]-\Pr\left[1\leftarrow\mathcal{A}^{|h\rangle,|h'\rangle}()\right]\right|\leq 2q_h/\sqrt{|\mathcal{X}'|}$$

3 Computational adaptive reprogramming in the QROM

We propose a computational adaptive reprogramming framework in the QROM. In Supp. Mat. A, we will review Unruh's adaptive O2H lemma [Unr14a] and discuss why our lemma (namely, Lemma 3.1) cannot be proved by using hybrid arguments of Unruh's adaptive O2H lemma.

Let $\mathcal{A}$ be an adversary that has quantum access to $\mathcal{H}:\mathcal{X}\to\mathcal{Y}$ and takes in_0 as input and terminates by outputting out_n. During its execution, $\mathcal{A}$ outputs some out_i and then takes in_{i+1} as input ($0\leq i\leq n-1$). We view $\mathcal{A}$ as a $(n+1)$-stage adversary, $(\mathcal{A}_0,...,\mathcal{A}_n)$, where $\mathcal{A}_i$ takes in_i as input and outputs out_i. Here $\mathsf{in}_0,\mathsf{out}_0,\mathsf{in}_1,...,\mathsf{in}_n$, and out_n can be arbitrary classical information. In this paper, we consider post-quantum setting where adversaries have quantum access to hash functions. The classical information $\mathsf{in}_0,\mathsf{out}_0,\mathsf{in}_1,...,\mathsf{in}_n,\mathsf{out}_n$ capture the interaction between $\mathcal{A}$ and the security game simulator, and they will be specified in a concrete use of our framework.

We write $\mathcal{A}=(\mathcal{A}_0,...,\mathcal{A}_n)$ to divide $\mathcal{A}$ into $n+1$ stages for better analysis. By writing $\mathsf{out}_i\leftarrow\mathcal{A}_i(\mathsf{in}_i)$ we mean that at stage i $\mathcal{A}$ receives input in_i and outputs out_i at the end of the stage. The index indicates the stage number of $\mathcal{A}$. So, all $\mathcal{A}_i$ are the same adversary $\mathcal{A}$ in different stages, and they share the quantum registers of $\mathcal{A}$. The same notation (of dividing $\mathcal{A}$ into different stages) is also used in Unruh's adaptive O2H lemma [Unr14a].

Games NonAda and Ada (as in Figure 4) are used to define our framework. $\mathcal{A}$ has quantum access to $\mathcal{H}$ which is either H or H_i. In NonAda, H will never get reprogrammed, while in Ada different stages of $\mathcal{A}$ will have access to different ROs H_i. That is, $\mathcal{A}_i$ queries H_i, and according to $\mathcal{A}_i$'s output out_i H_i will be reprogrammed and become H_{i+1} (cf. Items 07, 17 and 18). To formalize this, we define three algorithms $\mathtt{INIT}$, $\mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ in Figure 4 as:

- The initialization algorithm $\mathtt{INIT}$ outputs $((\mathbf{s},\mathsf{in}_0),\mathsf{H},\mathsf{H}_0)$ (cf. Items 01 and 11), where $\mathbf{s}$ is some parameter that used in a security reduction, in_0 is the initial input to $\mathcal{A}$, and H and H_0 are two random oracles. Here the tuple $((\mathbf{s},\mathsf{in}_0),\mathsf{H},\mathsf{H}_0)$ may have an arbitrary joint distribution.
- $\mathsf{F}_{\mathbf{s}}$ takes out_i as input and computes $(\mathsf{in}_{i+1},\mathsf{in}'_{i+1})$, where in_{i+1} is the input to $\mathcal{A}_{i+1}$ and in'_{i+1} is the information for reprogramming H_i. Here in'_{i+1} is used to capture the fact that $\mathcal{H}$ can be reprogrammed according to $\mathcal{A}_i$'s behavior, and the algorithm $\mathsf{Repro}_{\mathbf{s}}$ (described below) will take it as input. To make our lemma general and useful for a wider class of applications, we only require that $\mathsf{F}_{\mathbf{s}}$ does not have access to random oracles.
- $\mathsf{Repro}_{\mathbf{s}}$ is defined to reprogram $\mathcal{H}$ in Ada (cf. Item 17). $\mathsf{Repro}_{\mathbf{s}}$ takes in'_i and H_{i-1} as input. It returns a random oracle H_i which is from reprogramming H_{i-1}. The concrete reprogramming operation of $\mathsf{Repro}_{\mathbf{s}}$ depends on the concrete use of our framework. Here we only require $\mathsf{Repro}_{\mathbf{s}}$ to be deterministic.

Let S_i be a set such that $\mathsf{H}\setminus S_i=\mathsf{H}_i\setminus S_i$ (namely, for all $x\in\mathcal{X}$, if $x\in S_i$, then $\mathsf{H}(x)\neq\mathsf{H}_i(x)$). $\mathcal{A}$ can only distinguish Ada and NonAda, if it queries a $x\in S_i$ (where $i\in\{0,...,n\}$). Since $\mathcal{A}$'s QRO queries

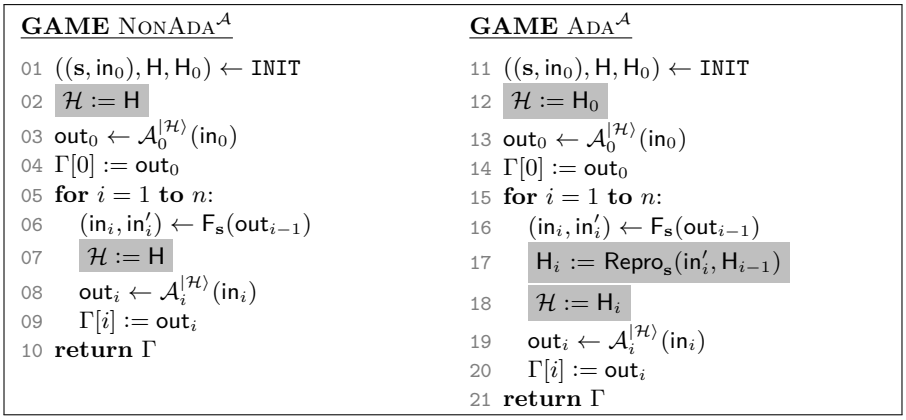

Figure 4: Games NonAda and Ada used in Lemma 3.1. The main difference between two games is highlighted with gray box. In both games, $\mathcal{A}$ is divided into $n+1$ stages, namely, $(\mathcal{A}_0, ..., \mathcal{A}_n)$. The input and output of $\mathcal{A}$ in each stage are classical information because we consider post-quantum settings. The list Γ stores $\mathcal{A}$'s outputs in each stage. $\mathsf{F}_\mathbf{s}$ is a deterministic algorithm that provides inputs for each stage of $\mathcal{A}$. $\mathsf{Repro}_\mathbf{s}$ is a deterministic algorithm that reprograms QROs. For a concise presentation, we assume that $\mathcal{A}_i$ takes $\mathcal{A}_{i-1}$'s final state as its initial state. In our framework, H_0 can be different to H.

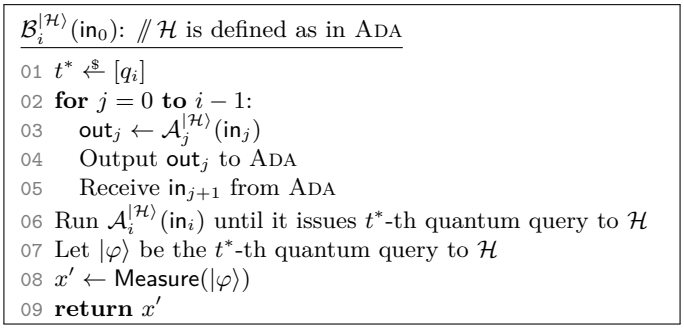

Figure 5: Algorithm $\mathcal{B}_i$ (used in Lemma 3.1) plays Game Ada (where $i \in [n]$). $\mathcal{B}_i$ proceeds identically with $(\mathcal{A}_1, ..., \mathcal{A}_i)$, except that $\mathcal{B}_i$ measures the t^*-th QRO query issued by $\mathcal{A}_i$ and then outputs the measurement outcome.

are superposition states, we need to define extractor $\mathcal{B}_i$ as in Figure 5 to bound the difference between NonAda and Ada. This follows the works in [Unr14a, SXY18, KSS+20].

Lemma 3.1 formalizes our framework. Its proof is postponed in Supp. Mat. C.

Lemma 3.1 *Let $\mathcal{A}$ be an adversary that can be divided into $(n+1)$ stages as in Figure 4 and has quantum access to random oracle $\mathcal{H}$ ($= \mathsf{H}$ in NonAda or H_i in Ada). Let Ev be a classical event that may be raised by $\mathcal{A}$ in NonAda or Ada. Suppose that $\mathcal{A}$ queries $\mathcal{H}$ at most q_i times in its i-th stage and at most $q := q_0 + \cdots q_n$ times in total during the game. Then for all algorithms $\mathtt{INIT}$, $\mathsf{F}_\mathbf{s}$, and $\mathsf{Repro}_\mathbf{s}$ (as described earlier), there exists adversaries $\mathcal{B}_i$ for $i \in \{0, ..., n\}$ (shown in Figure 5) such that*

$$\left|\Pr\left[\mathsf{Ev} : \text{NonAda}^{\mathcal{A}}\right] - \Pr\left[\mathsf{Ev} : \text{Ada}^{\mathcal{A}}\right]\right| \leq \sum_{k=0}^{n}\sum_{i=0}^{k} 2q_i \Pr\sqrt{\left[x' \leftarrow \mathcal{B}_i^{\mathcal{H}} \ s.t. \ x' \in S_i : \text{Ada}^{\mathcal{B}_i}\right]}, \tag{2}$$

where S_i is a set such that $\mathsf{H}\backslash S_i = \mathsf{H}_i\backslash S_i$. Such an S_i is defined by the operations in $\mathsf{Repro}_\mathbf{s}$. $\Pr\left[\mathsf{Ev} : \text{NonAda}^{\mathcal{A}}\right]$ and $\Pr\left[\mathsf{Ev} : \text{Ada}^{\mathcal{A}}\right]$ are the probabilities that $\mathcal{A}$ triggers Ev in NonAda and in Ada, respectively.

Discussions on Lemma 3.1. In Ada, reprogramming the RO is captured by algorithm $\mathsf{Repro}_\mathbf{s}$. How the reprogramming is done will be specified in a concrete use of Lemma 3.1. This is to make our framework general. The difference between NonAda and Ada is that between H and H_i caused by $\mathsf{Repro}_\mathbf{s}$.

Concretely, in i-th stage, $\mathsf{Repro}_{\mathbf{s}}$ will define a set S_i such that $\mathsf{H} \setminus S_i = \mathsf{H}_i \setminus S_i$. For any $k \in \{0, ..., n\}$, if $\mathcal{A}$ queries $\mathcal{H}$ with an $x \in \cup_{0 \le i \le k} S_k$ before the end of its k-th stage, then $\mathcal{A}$ can distinguish NonAda and Ada. To bound this in the quantum setting, our approach is to randomly measure $\mathcal{A}$'s queries to $\mathcal{H}$, which is captured by $\mathcal{B}_i$ (in Figure 5). The advantage of $\mathcal{A}$ distinguishing NonAda and Ada is bounded by the probability that $\mathcal{B}_i$'s output falls into S_i.

More Discussions on F and Repro in Figure 4. When defining our framework, we do not make any requirement on the efficiencies of $\mathsf{F}_{\mathbf{s}}$ and $\mathsf{Repro}_{\mathbf{s}}$. However, when we use this framework to construct (efficient) reduction, $\mathsf{F}_{\mathbf{s}}$ and $\mathsf{Repro}_{\mathbf{s}}$ are required to be efficient (namely, running in quantum probabilistic polynomial time) and the description of QRO is polynomially bounded [BDF+11, Zha12, KLS18]. For instance, we can use a $2q$-independent hash function [Zha12] and the list of reprogramming points (which are inputs to the hash and polynomial-bounded) to describe this QRO.

Why our Framework covers the Work of Grilo et al.. By specifying $\mathsf{F}_{\mathbf{s}}$ and $\mathsf{Repro}_{\mathbf{s}}$, we can describe Grilo et al.'s framework using our framework. In Grilo et al.'s framework [GHHM21], the i-th output of $\mathcal{A}$ is a distribution $\mathsf{out}_i := p_i$. $\mathsf{F}_{\mathbf{s}}$ can be defined as, on input p_i, it samples a reprogramming point (x_i, x_i') from p_i and an independently random y_i and outputs $(\mathsf{in}_{i+1} := (x_i, x_i'), \mathsf{in}'_{i+1} := (x_i, x_i', y_i))$[3]. $\mathsf{Repro}_{\mathbf{s}}$ can be defined as, on input $\mathsf{in}'_{i+1} := (x_i, x_i', y_i)$, it reprograms the QRO $\mathcal{H} := \mathcal{H}[(x_i, x_i') \to y_i]$ and returns the reprogrammed QRO. Their framework implicitly requires that the probability bound for $\mathcal{A}$ to learn x_i, x_i' (before seeing them) is information-theoretic. Namely, p_i should have enough entropy. Some important advantage of our framework, compared with Grilo et al.'s [GHHM21], are as follows:

- Grilo et al.'s framework requires the reprogramming points have high entropy and it is hard to find them even for unbounded adversary, while our framework does not have such restrictions. If $\mathcal{A}$ is a PPT adversary, our framework provides efficient extractors $\mathcal{B}_i$'s to bound the difference of $\mathcal{A}$ in NonAda and Ada. In our proofs, we need to instantiate $\mathtt{INIT}, \mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ efficiently. This $\mathcal{B}_i$ can be used to do a reduction in breaking some computational hard problem, for instance, the OW-CPA security. However, the Grilo et al.framework cannot be used to do any efficient reduction.
- Our framework allows NonAda and Ada to start from different QROs, while the Grilo et al.framework starts from the same QRO. Starting from different QROs allows us to consider more complicated cases of adaptive reprogramming. All security proofs in this paper are examples for this, and for SO and Bi-SO security we require this.
- Our framework also supports delayed analysis. In some complicated proofs, the difference between non-reprogramming and reprogramming games cannot be immediately bounded, and we may need extra game sequences to postpone such a bound. Our framework supports delayed analysis, since we can use extra game sequences to bound the winning probability of $\mathcal{B}_i$ (i.e. $\mathcal{B}_i$ outputs $x \in S_i$). In particular, our tightly-secure SO-CCA PKE scheme in Section 5 requires delayed analysis.

4 Selective Opening Security of Fujisaki-Okamoto's PKE in the QROM

We prove the selective-opening (SO) security of two Fujisaki-Okamoto(FO)-style PKE schemes in the QROM. As a warm-up, our first scheme is SO secure against chosen-plaintext attacks (SO-CPA), and the scheme follows the idea of hybrid encryption. It offers a simple example about how to use our framework. Our second scheme is SO secure against chosen-ciphertext attacks (SO-CCA). It is the same scheme as in [SS19, Section 3.2], but our proof is showing adaptive SO-CCA security, while the original proof in [SS19] has a subtle gap and the gap still exists even if we consider the non-adaptive security notion (cf. discussion in Introduction).

In both schemes, let $\mathsf{PKE} := (\mathsf{KG}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ be a $(1-\delta)$-correct PKE scheme with message space $\mathcal{M}'$, ciphertext space $\mathcal{C}'$, and randomness space $\mathcal{R}'$. Let $G : \mathcal{M}' \to \mathcal{R}'$ be a hash function.

4.1 Selective Opening Security against Chosen-Plaintext Attacks

Let $H : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M}$ be a hash function. Our first PKE scheme $\mathsf{wPKE} = (\mathsf{wKG}, \mathsf{wEnc}, \mathsf{wDec})$ (where 'w' stands for weak) with message space $\mathcal{M}$ and is defined as in Figure 6. Theorem 4.1 states that wPKE is adaptive SO-CPA secure when modeling G and H as QROs.

[3] The randomness for sampling can be included in $\mathbf{s}$, since it is captured by the game simulator.

wKG	wEnc(pk, $m \in \mathcal{M}$)	wDec(sk, (e, d))
01 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}_0$	03 $r \xleftarrow{\$} \mathcal{M}'$	08 $r' := \mathsf{Dec}_0(\mathsf{sk}, e)$
02 **return** $(\mathsf{pk}, \mathsf{sk})$	04 $e := \mathsf{Enc}_0(\mathsf{pk}, r; G(r))$	09 $K := H(r', e)$
	05 $K := H(r, e)$	10 $m := K \oplus d$
	06 $d := K \oplus m$	11 **return** m
	07 **return** (e, d)	

Figure 6: A SO-CPA secure PKE scheme $\mathsf{wPKE} = (\mathsf{wKG}, \mathsf{wEnc}, \mathsf{wDec})$

Game $\mathbf{G}_0$-$\mathbf{G}_3$		OPEN(i)	
01 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}_0$		17 $I := I \cup \{i\}$	
02 $\mathcal{M}_a \leftarrow \mathcal{A}^{\lvert G \times H \rangle}(\mathsf{pk})$		18 **return** (m_i, r_i)	
03 **for** $i \in [\mu]$		$H(r, e)$	
04 $\quad \mathbf{m}[i] := m_i \leftarrow \mathcal{M}_a, r_i \xleftarrow{\$} \mathcal{R}'$		19 **if** $\exists i \in I$ s.t. $(r, e) = (r_i, e_i)$	// $\mathbf{G}_2$-$\mathbf{G}_3$
05 $\quad R_i := G(r_i)$		20 $\quad$ **return** K_i	// $\mathbf{G}_2$
06 $\quad R_i \xleftarrow{\$} \mathcal{R}'$	// $\mathbf{G}_2$-$\mathbf{G}_3$	21 $\quad$ **return** $d_i \oplus m_i$	// $\mathbf{G}_3$
07 $\quad e_i := \mathsf{Enc}_0(\mathsf{pk}, r_i; R_i)$		22 **return** $h(r, e)$	
08 $\quad K_i := H(r_i, e_i)$	// $\mathbf{G}_1$	$G(r)$	
09 $\quad K_i \xleftarrow{\$} \mathcal{M}$	// $\mathbf{G}_2$	23 **if** $\exists i \in I$ s.t. $r = r_i$	// $\mathbf{G}_2$-$\mathbf{G}_3$
10 $\quad d_i := K_i \oplus m_i$	// $\mathbf{G}_1$-$\mathbf{G}_2$	24 $\quad$ **return** R_i	// $\mathbf{G}_2$-$\mathbf{G}_3$
11 $\quad d_i \xleftarrow{\$} \mathcal{M} \backslash \{d_1, ..., d_{i-1}\}$	// $\mathbf{G}_3$	25 **return** $g(r)$	
12 $\quad \mathbf{c}[i] := (e_i, d_i)$			
13 **if** $\exists i \neq j$ s.t. $K_i = K_j$			
14 $\quad$ **abort**	// $\mathbf{G}_1$-$\mathbf{G}_2$		
15 $out \leftarrow \mathcal{A}^{\text{OPEN}, \lvert G \times H \rangle}(\mathbf{c})$			
16 **return** $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$			

Figure 7: Games $\mathbf{G}_0$-$\mathbf{G}_3$ for proving Theorem 4.1.

Theorem 4.1 *If* PKE *is OW-CPA secure, then* wPKE *in Figure 6 is adaptive SO-CPA secure (Definition 2.5). Concretely, for security parameter λ and $\mu := \mu(\lambda)$ (polynomially bounded), for any SO-CPA adversary $\mathcal{A}$ and relation* Rel*, there exist a simulator $\mathcal{S}$ and an adversary $\mathcal{B}^{ow}$ such that $\mathbf{T}(\mathcal{S}) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}^{ow})$ and*

$$\mathsf{Adv}^{\mathsf{SO\text{-}CPA}}_{\mathsf{wPKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \leq 2(n_{\text{OP}} + 1)^2 q \sqrt{2\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{ow})} + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}},$$

where μ, q_G, q_H, and n_{OP} are the maximum numbers of $\mathcal{A}$'s challenge ciphertexts, $\mathcal{A}$'s queries to G, H, and OPEN*, respectively. $q = q_G + q_H$.*

Proof. Let $h : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M}$ and $g : \mathcal{M}' \to \mathcal{R}'$ be two internal quantum-accessible random oracles that are used to respond queries to H and G, respectively. Following the convention in [KLS18, SXY18], in our proof we simulate H and G using two internal quantum-accessible random oracles $h : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M}$ and $g : \mathcal{M}' \to \mathcal{R}'$, respectively.

Our proof consists a sequence of games defined in Figure 7. We will use our framework in Section 3 to finish the proof. To fit into the syntax of our framework, we combine G and H as one random oracle $G \times H$ such that $G \times H(r', r, e) := (G(r'), H(r, e))$. If $\mathcal{A}$ only queries $G(r')$, we view it as querying $G \times H(r', r, e)$ for some dummy (r, e) and ignoring $H(r, e)$ in the response. $\mathcal{A}$ can query $G \times H$ at most $q = q_H + q_G$ times. This was also used in [JZC+18]. $\mathbf{G}_0$ is equivalent to $\mathsf{REAL\text{-}SO\text{-}CPA}_{\mathsf{wPKE}}$, thus

$$\Pr\left[\mathsf{REAL\text{-}SO\text{-}CPA}^{\mathcal{A}}_{\mathsf{wPKE}} \Rightarrow 1\right] = \Pr\left[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1\right]$$

Game $\mathbf{G}_1$: If in the challenge ciphertexts there exist K_i and K_j for $i \neq j$ such that $K_i = K_j$, then we abort the game. Such K_i and K_j collide only if r_i and r_j collide or $H(r_i, e_i)$ and $H(r_j, e_j)$ collide with different r_i and r_j. By birthday bounds, and we have

$$\left|\Pr\left[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|}$$

Game $\mathbf{G}'_1$-$\mathbf{G}'_2$

$((\mathbf{s}, \mathsf{in}_0), \mathsf{H}, \mathsf{H}_0) \leftarrow \texttt{INIT}$
Initialize $\mathcal{A}_{1,0}$ with
  the final state of $\mathcal{A}_0$ in $\texttt{INIT}$
$\mathcal{H} := \mathsf{H}$ // $\mathbf{G}'_1$
$\mathcal{H} := \mathsf{H}_0$ // $\mathbf{G}'_2$
$\mathsf{out}_0 \leftarrow \mathcal{A}_{1,0}^{|\mathcal{H}\rangle}(\mathsf{in}_0)$
$\Gamma[0] := \mathsf{out}_0$
for $i = 1$ **to** n_{OP}:
  $(\mathsf{in}_i, \mathsf{in}'_i) := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_{i-1})$
  $\mathsf{H}_i := \mathsf{Repro}_{\mathbf{s}}(\mathsf{in}'_i, \mathsf{H}_{i-1})$ // $\mathbf{G}'_2$
  $\mathcal{H} := \mathsf{H}_i$ // $\mathbf{G}'_2$
  $\mathsf{out}_i \leftarrow \mathcal{A}_{1,i}^{|\mathcal{H}\rangle}(\mathsf{in}_i)$
  $\Gamma[i] := \mathsf{out}_i$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, \Gamma[n_{\mathrm{OP}}])$

$\underline{\texttt{INIT}}$

$I := \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}_0$
$\mathcal{M}_a \leftarrow \mathcal{A}_0^{|g \times h\rangle}(\mathsf{pk})$
Let g' and h' be internal QROs.
for $i \in [\mu]$:
  $\mathbf{m}[i] := m_i \leftarrow \mathcal{M}_a, \mathbf{r}[i] := r_i \xleftarrow{\$} \mathcal{M}'$
  $R_i := g(r_i)$, $\mathbf{R}[i] := R_i$
  $e_i := \mathsf{Enc}_0(\mathsf{pk}, r_i; R_i)$
  $K_i := h(r_i, e_i)$, $d_i := K_i \oplus m_i$ // By $\mathbf{G}_1$, all K_i's are different.
  $\mathbf{c}[i] := (e_i, d_i)$
$\mathbf{s} := (\mathcal{M}_a, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c})$, $\mathsf{in}_0 := \mathbf{c}$
$S_0 := \{r_i\}_{i \in [\mu]} \times \{(r_i, e_i)\}_{i \in [\mu]}$
$G := g, H := h$
Let $G_0 \times H_0$ be a QRO such that $G_0 \times H_0(x) := \begin{cases} g \times h(x), & (x \notin S_0) \\ g' \times h'(x), & (\text{else}) \end{cases}$
// Namely, $(G_0 \times H_0) \backslash S_0 = (G \times H) \backslash S_0$
return $((\mathbf{s}, \mathsf{in}_0), (G \times H), (G_0 \times H_0))$

$\underline{\mathsf{F}_{\mathbf{s}}(\mathsf{out})}$

parse $i := \mathsf{out}$
parse $(\mathcal{M}_a, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}) := \mathbf{s}$
$I := I \cup \{i\}$
$r_i := \mathbf{r}[i], m_i := \mathbf{m}[i], (e_i, d_i) := \mathbf{c}[i]$
$\mathsf{in} := (m_i, r_i)$, $\mathsf{in}' := (m_i, r_i, e_i, d_i)$
return $(\mathsf{in}, \mathsf{in}')$

$\underline{\mathsf{Repro}_{\mathbf{s}}(\mathsf{in}', (G \times H))}$

parse $(m, r, e, d) := \mathsf{in}'$
$G' := G[r \to R]$
$H' := H[(r, e) \to d \oplus m]$
  // Namely, we set $H(r_i, e_i) := K_i$
  // and denote the new oracle as H'
return $G' \times H'$

Figure 8: Constructions of $\texttt{INIT}, \mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ and games $\mathbf{G}'_1$ and $\mathbf{G}'_2$. $G' := G[r_i \to R_i]$ (similarly, $H' := H[(r_i, e_i) \to K_i]$) means that we set $G'(r_i) := R_i$ and $G'(r) := G(r)$ for $r \neq r_i$. Oracles $g, g' : \mathcal{M}' \to \mathcal{R}'$, and $h, h' : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M}$ are four independent internal quantum-accessible random oracles.

Game $\mathbf{G}_2$: R_i and K_i in the challenge ciphertexts are chosen randomly, instead of using G and H. If $\mathcal{A}$ queries $\textsc{Open}(i)$, then we reprogram G and H such that $G(r_i) := R_i$ and $H(r_i, e_i) := K_i$.

In the following, we use Lemma 3.1 to bound the difference between $\mathbf{G}_1$ and $\mathbf{G}_2$. In $\mathbf{G}_2$, $\mathcal{A}$'s $\textsc{Open}$ queries will make QRO $G \times H$ reprogrammed, while in $\mathbf{G}_1$, QRO $G \times H$ does not get reprogrammed. So, we can view $\mathbf{G}_1$ and $\mathbf{G}_2$ as concrete cases of $\textsc{NonAda}$ and $\textsc{Ada}$, respectively. For simplicity, we denote $\mathcal{A} := (\mathcal{A}_0, (\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\mathrm{OP}}}))$, where $\mathcal{A}_0$ is the initial stage of $\mathcal{A}$ and cannot query $\textsc{Open}$, and $(\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\mathrm{OP}}})$ is the stage that $\mathcal{A}$ receives the challenge ciphertexts $\mathbf{c}$ and can query $\textsc{Open}$. Let $\mathcal{A}_1 := (\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\mathrm{OP}}})$. $\mathcal{A}_1$'s initial state is the final state of $\mathcal{A}_0$. $\mathcal{A}_{1,k}$ is defined with respect to $\textsc{Open}$ queries:

- Before any $\textsc{Open}$ query (i.e., at the 0-th stage), $\mathcal{A}_{1,0}$ takes $\mathsf{in}_0 := \mathbf{c}$ as input and outputs the first opening index $\mathsf{out}_0 := (i_1)$.
- At k-th stage ($1 \leq k \leq n_{\mathrm{OP}} - 1$), $\mathcal{A}_{1,k}$ receives $\mathsf{in}_k = (m_{i_k}, r_{i_k})$ as the result of the $(k-1)$-th $\textsc{Open}$ query and finishes the stage by outputting the $(k+1)$-th opening index $\mathsf{out}_k := (i_{k+1})$
- Finally, at the n_{OP} stage, $\mathcal{A}_{1,n_{\mathrm{OP}}}$ receives $\mathsf{in}_{n_{\mathrm{OP}}} = (m_{i_{n_{\mathrm{OP}}}}, r_{i_{n_{\mathrm{OP}}}})$ and terminates by outputting $\mathsf{out}_{n_{\mathrm{OP}}} = \mathit{out}$ (the final output of SO adversary).

To formally show why $\mathbf{G}_1$ and $\mathbf{G}_2$ are concrete cases of $\textsc{NonAda}$ and $\textsc{Ada}$, respectively, in Figure 8, we define $\texttt{INIT}$, $\mathsf{F}_{\mathbf{s}}$, $\mathsf{Repro}_{\mathbf{s}}$, $\mathbf{G}'_1$ and $\mathbf{G}'_2$. Games $\mathbf{G}'_1$ and $\mathbf{G}'_2$ are only defined to show how our proof

follows the syntax of our framework. They have the same forms as NonAda and Ada.

Now we argue that $\mathbf{G}_1$ and $\mathbf{G}_2$ are concrete cases of NonAda and Ada, respectively. Namely, $\mathbf{G}_1$ and $\mathbf{G}_2$ in Figure 7 are equivalent to $\mathbf{G}'_1$ and $\mathbf{G}'_2$ in Figure 8, respectively. Firstly, algorithm INIT in Figure 8 run the codes from Item 01 to Item 12 in Figure 7. Since in $\mathcal{A}_0$'s view, $\mathbf{G}_1$ is the same as $\mathbf{G}_2$ (it does not see any challenge ciphertexts), the distribution of $\mathcal{M}_a$ and $\mathbf{m}$ in $\mathbf{G}_1$ is the same as the one in $\mathbf{G}_2$, and thus the output of INIT and the final state of $\mathcal{A}_0$ in INIT in $\mathbf{G}'_1$ are the same as those in $\mathbf{G}'_2$. Secondly, $\mathsf{F}_\mathbf{s}$ simulates the Open oracle and $\mathsf{Repro}_\mathbf{s}$ simulates the reprogramming operations on G and H. In $\mathbf{G}'_1$, G and H will not be reprogrammed, but in $\mathbf{G}'_2$, G and H will be reprogrammed, according to $\mathcal{A}$'s output. This is the same as in $\mathbf{G}_2$.

Moreover, when running $\mathcal{A}_{1,k}$, our $\mathsf{Repro}_\mathbf{s}$ defines a set

$$S_k := \{(r,(r',e')) \mid \exists i \in [\mu] \backslash I_k \text{ s.t. } r = r_i \text{ or } (r',e') = (r_i,e_i)\} \tag{3}$$

where $I_k := \{i_1,...,i_k\}$ is the opening index set I in $\mathcal{A}_1$'s k-th stage. Answers of $G \times H$ on S_k are only different in $\mathbf{G}_1$ (i.e., NonAda) and $\mathbf{G}_2$ (i.e., Ada). For $k=0$, S_0 is defined at line 35 and $I_0 = \emptyset$.

Now we consider the probability that $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out) = 1$. I and out are determined by $\mathcal{A}_1$. $\mathcal{M}_a$ is output by $\mathcal{A}_0$, and $\mathbf{m}$ is determined by $\mathcal{M}_a$. Since in $\mathcal{A}_0$'s view, $\mathbf{G}_1$ is the same as $\mathbf{G}_2$ (since it does not see challenge ciphertexts), thus the distribution of $\mathcal{M}_a$ and $\mathbf{m}$ in $\mathbf{G}_1$ is the same as the one in $\mathbf{G}_2$. Therefore, the probability difference between the classical event that $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out) = 1$ in $\mathbf{G}_1$ and the similar event in $\mathbf{G}_2$, is determined by the probability difference between the event that $\mathcal{A}_1$ outputs a particular (I, out) (i.e., Γ in Figure 8) in $\mathbf{G}_1$ and the similar event in $\mathbf{G}_2$. Therefore, we have

$$\left|\Pr\left[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1\right]\right| \leq \left|\Pr\left[\mathbf{G}_1'^{\mathcal{A}_1} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_2'^{\mathcal{A}_1} \Rightarrow 1\right]\right| + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}} \tag{4}$$

This bound includes a term $\frac{2\mu q}{\sqrt{|\mathcal{R}'|}}$, since $\mathcal{A}_0$ also has quantum access to $|G \times H\rangle$, and this term is the probability that the first stage (i.e., $\mathcal{A}_{1,0}$) of $\mathcal{A}_1$ learns r_i before seeing challenge ciphertexts. Such probability is only information-theoretic.

We now use Lemma 3.1 to bound Equation (4). Since $\mathbf{G}'_1$ is a NonAda game and $\mathbf{G}'_2$ is an Ada game, by Lemma 3.1, there exist adversaries $\mathcal{B}_i$ $(0 \leq i \leq n_{\mathrm{OP}})$, which take $\mathsf{in}_0 = \mathbf{c}$ as its input and output $x \in S_k$ where the set S_i is defined in (3), such that

$$\left|\Pr\left[\mathbf{G}_1'^{\mathcal{A}_1} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_2'^{\mathcal{A}_1} \Rightarrow 1\right]\right| \leq \sum_{k=0}^{n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[x \leftarrow \mathcal{B}_i \text{ s.t. } x \in S_i : \mathbf{G}_2'^{\mathcal{B}_i}\right]} \tag{5}$$

Here $\mathcal{B}_i$ proceeds the same as $(\mathcal{A}_{1,0},...,\mathcal{A}_{1,i})$ except that it randomly measures a QRO query issued by $\mathcal{A}_{1,i}$. Moreover, since $\mathcal{A}_{1,0}$'s initial state is the final state of $\mathcal{A}_0$, $\mathcal{B}_i$ starts with state of $\mathcal{A}_0$ (cf. Item 07).

Based on $\mathcal{B}_i$, we construct an adversary $\mathcal{B}_i^{\mathrm{ow}}$ (in Figure 9) to breaks OW-CPA security of PKE. By the construction of $\mathcal{B}_i^{\mathrm{ow}}$, if $\mathcal{A}_1$ does not open t^*, and r or r' equals the solution of e^*, then $\mathcal{B}_i^{\mathrm{ow}}$ wins. So the winning probability for $\mathcal{B}_i^{\mathrm{ow}}$ to breaks the OW-CPA challenge is:

$$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_i^{\mathrm{ow}}) = \frac{1}{2}\frac{\mu - n_{\mathrm{OP}}}{\mu}\frac{1}{\mu - n_{\mathrm{OP}}}\Pr\left[x \leftarrow \mathcal{B}_i \text{ s.t. } x \in S_i\right],$$

and thus we have

$$\Pr\left[x \leftarrow \mathcal{B}_i \text{ s.t. } x \in S_i : \mathbf{G}_2'^{\mathcal{B}_i}\right] \leq 2\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_i^{\mathrm{ow}}) \tag{6}$$

Let $\mathcal{B}^{\mathrm{ow}}$ be the adversary that has highest advantage against PKE among $\{\mathcal{B}_i^{\mathrm{ow}}\}_{i\in\{0,...,n\}}$. Then equation (6) can be written as:

$$\Pr\left[x \leftarrow \mathcal{B}_i \text{ s.t. } x \in S_i : \mathbf{G}_2'^{\mathcal{B}_i}\right] \leq 2\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}^{\mathrm{ow}}), \text{ for } \forall i \in [\mu] \tag{7}$$

By combining Equations (4) to (7), we have

$$\left|\Pr\left[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1\right]\right| \leq 2(n_{\mathrm{OP}}+1)^2 q \sqrt{2\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}^{\mathrm{ow}})} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}$$

$\underline{\mathcal{B}_i^{\mathrm{ow}}(\mathsf{pk}^*, e^*)}$ // (pk^*, e^*) is a OW-CPA challenge of PKE

$I := \emptyset$
$((\mathbf{s}, \mathsf{in}_0), (G \times H), (G_0 \times H_0)) \leftarrow \mathtt{INIT}$ // $\mathtt{INIT}$ is defined in Figure 8 and // it uses pk^* instead of KG_0
parse $(\mathcal{M}_a, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}) := \mathbf{s}$
parse $\mathbf{c} := \mathsf{in}_0$
$t^* \xleftarrow{\$} [\mu], (e_{t^*}, d_{t^*}) := \mathbf{c}[t^*]$
$\mathbf{c}[t^*] := (e^*, d_{t^*})$, $\mathsf{in}_0 := \mathbf{c}$ // embed the challenge
Initialize $\mathcal{B}_i$ with $\mathcal{A}_0$'s final state in $\mathtt{INIT}$.
if $i = 0$: **goto** line 18
$\mathsf{out}_0 \leftarrow \mathcal{B}_i^{|G_0 \times H_0\rangle}(\mathsf{in}_0)$
if $\mathsf{out}_0 = t^*$: **abort**
$(\mathsf{in}_1, \mathsf{in}_1') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_0)$ // $\mathsf{F}_{\mathbf{s}}$ is defined in Figure 8
$(G_1 \times H_1) := \mathsf{Repro}_{\mathbf{s}}(\mathsf{in}_1', (G_0 \times H_0))$ // $\mathsf{Repro}_{\mathbf{s}}$ is defined in Figure 8
for $j = 1$ **to** $i - 1$:
  $\mathsf{out}_j \leftarrow \mathcal{B}_i^{|G_j \times H_j\rangle}(\mathsf{in}_j)$
  if $\mathsf{out}_j = t^*$: **abort**
  $(\mathsf{in}_{j+1}, \mathsf{in}_{j+1}') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_j)$
  $(G_{j+1} \times H_{j+1}) := \mathsf{Repro}_{\mathbf{s}}(\mathsf{in}_{j+1}', (G_j \times H_j))$
$(r_0', (r_1', e')) \leftarrow \mathcal{B}_i^{|G_i \times H_i\rangle}(\mathsf{in}_i)$ // perform measurement
$b \xleftarrow{\$} \{0,1\}, r^* := r_b'$ // randomly choose a solution
return r^*

Figure 9: The constructions of OW-CPA adversaries $\mathcal{B}_i^{\mathrm{ow}}$ for $i \in \{0, ..., n_{\mathrm{OP}}\}$. $\mathcal{B}_i^{\mathrm{ow}}$ simulates $\mathbf{G}_2'$ (which is a concrete case of ADA in Figure 4) for $\mathcal{B}_i$ to break PKE. F and Repro are defined as in Figure 8.

Game $\mathbf{G}_3$: We change the generation of K_i and d_i. Now we firstly sample d_i uniformly at random, and replace all K_i as $d_i \oplus m_i$. This change is conceptual since in $\mathbf{G}_2$, all K_i are independently and uniformly random. In $\mathbf{G}_1$, we excluded any collision of K_i, so, in $\mathbf{G}_3$, it is equivalent to sample d_i in a collision-free way. Therefore, we have

$$\Pr\left[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1\right]$$

CONSTRUCTION OF SO SIMULATOR. We construct a SO simulator $\mathcal{S}$ that is simulating $\mathbf{G}_3$ for $\mathcal{A}$ and interacts with the $\mathsf{IDEAL\text{-}SO\text{-}CPA}^{\mathcal{S}}_{\mathsf{wPKE}}$ game. The simulation process is shown in Figure 10. Obviously, $\mathcal{S}$ can perfectly simulates $\mathbf{G}_3$. So, we have

$$\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{IDEAL\text{-}SO\text{-}CPA}^{\mathcal{S}}_{\mathsf{wPKE}} \Rightarrow 1]$$

In conclusion, for any SO-CPA adversary $\mathcal{A}$, there exists efficient simulator $\mathcal{S}$ such that

$$\begin{aligned}&\left|\Pr[\mathsf{REAL\text{-}SO\text{-}CPA}^{\mathcal{A}}_{\mathsf{wPKE}} \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CPA}^{\mathcal{S}}_{\mathsf{wPKE}} \Rightarrow 1]\right| \\ &\quad \leq 2(n_{\mathrm{OP}}+1)^2 q\sqrt{2\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{ow})} + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}.\end{aligned}$$

□

4.2 Selective Opening Security against Chosen-Ciphertext Attacks

Let $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ be a MAC scheme with key space $\mathcal{K}^{\mathtt{mac}}$, and let $H : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ be a hash function, where $\mathcal{C}$ is the ciphertext space of PKE. The second PKE scheme $\mathsf{sPKE} = (\mathsf{sKG}, \mathsf{sEnc}, \mathsf{sDec})$ (Figure 11) is a combination of a modular Fujisaki-Okamoto's transformation $\mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ [JZC+18, HHK17], one-time pad, and the one-time MAC scheme MAC. It has similar structure with the scheme in [HJKS15, SS19].

This scheme is adaptive SO-CCA secure when modeling G and H as QROs, as stated in Theorem 4.2. The main difference between the proof of Theorem 4.2 and the one of Theorem 4.1 is that the simulator needs to simulate the decryption oracle for the adversary. We use the encrypt-then-hash technique

$\mathcal{S}^{\text{OPEN}'}$

Chooses QROs g, h at random
$I = \emptyset$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{wKG}$
$\mathcal{M}_a \leftarrow \mathcal{A}(\mathsf{pk})$
Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$
for $i \in [\mu]$
  $r_i \xleftarrow{\$} \mathcal{M}', R_i \xleftarrow{\$} \mathcal{R}'$
  $e_i := \mathsf{Enc}_0(\mathsf{pk}, r_i; R_i)$
  $d_i \xleftarrow{\$} \mathcal{M} \backslash \{d_1, ..., d_{i-1}\}$
  $\mathbf{c}[i] := (e_i, d_i)$
$out \leftarrow \mathcal{A}^{\text{OPEN}, |G \times H\rangle}(\mathbf{c})$
return out

OPEN(i)

$I := I \cup \{i\}$
Queries OPEN$'$ on i and receives m_i
return (m_i, r_i)

$H(r, e)$

if $\exists i \in I$ s.t. $(r, e) = (r_i, e_i)$
  return $d_i \oplus m_i$
return $h(r, e)$

$G(r)$

if $\exists i \in I$ s.t. $r = r_i$
  return R_i
return $g(r)$

Figure 10: The simulator $\mathcal{S}$ of the proof of Theorem 4.1.

sKG

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KG}_0$
$k \xleftarrow{\$} \mathcal{M}'$
$\mathsf{pk}' := \mathsf{pk}$
$\mathsf{sk}' := (\mathsf{sk}, k)$
return $(\mathsf{pk}', \mathsf{sk}')$

sEnc$(\mathsf{pk}, m \in \mathcal{M})$

$r \xleftarrow{\$} \mathcal{M}'$
$e := \mathsf{Enc}_0(\mathsf{pk}, r; G(r))$
$(K, K^{\mathtt{mac}}) := H(r, e)$
$d := K \oplus m$
$\tau := \mathsf{Tag}(K^{\mathtt{mac}}, d)$
return (e, d, τ)

sDec$((\mathsf{sk}, k), (e, d, \tau))$

$r' := \mathsf{Dec}_0(\mathsf{sk}, e)$
if $r' = \perp$
  or $e \neq \mathsf{Enc}_0(\mathsf{pk}, r'; G(r'))$
  $(K, K^{\mathtt{mac}}) := H(k, e)$
else $(K, K^{\mathtt{mac}}) := H(r', e)$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \perp$
return m

Figure 11: A SO-CCA secure PKE scheme $\mathsf{sPKE} = (\mathsf{sKG}, \mathsf{sEnc}, \mathsf{sDec})$

(widely used in CCA proof of PKE [JZC+18, KSS+20, SXY18]) to simulate decryption oracle without using the secret key and add a MAC verification in the decryption so that the adversary cannot forge valid MAC codes for any unopened cipheretext. The proof of Theorem 4.2 is given in Supp. Mat. F.

Theorem 4.2 *If* PKE *is OW-CPA secure and δ-correct, and* MAC *is otSUF-CMA secure, then the PKE scheme* sPKE *in Figure 11 is adaptive SO-CCA secure (Definition 2.5). Concretely, for security parameter λ and integer $\mu := \mu(\lambda)$ (polynomially bounded) for any SO-CCA adversary $\mathcal{A}$ and relation* Rel*, there exist a simulator $\mathcal{S}$ and adversaries $\mathcal{B}^{ow}$ and $\mathcal{F}$ such that $\mathbf{T}(\mathcal{S}) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}^{ow}) \approx \mathbf{T}(\mathcal{F})$ and*

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{SO\text{-}CCA}}_{\mathsf{sPKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \\
&\leq 3\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + 6(n_{\text{OP}} + 1)^2 q \sqrt{2\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{ow}) + \mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} \\
&+ \frac{2q_H}{\sqrt{2^k}} + 16(\mu + n_{\text{DEC}} + q + 1)^2 \delta + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + \frac{6\mu q}{\sqrt{|\mathcal{R}'|}} + \frac{\mu n_{\text{DEC}}}{|\mathcal{C}'| - n_{\text{DEC}}} + \frac{(2+\mu)q}{\sqrt{|\mathcal{M}'|}}
\end{aligned}$$

where μ, q_G, q_H, n_{OP}, and n_{DEC} are the maximum numbers of $\mathcal{A}$'s challenge ciphertexts, $\mathcal{A}$'s queries to $G, H,$ OPEN*, and* DEC*, respectively. $q = q_G + q_H$.*

5 Tight SO-CCA Security from Lossy Encryption

In this section, we show that if the underlying PKE is a lossy encryption [BHY09, HJR16], then the construction in Figure 11 is tightly SO-CCA secure. We recall the notion of lossy encryption from [HJR16].

Definition 5.1 (Lossy Encryption [HJR16]). Let $\mathsf{PKE}_1 := (\mathsf{KG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a PKE scheme with message space $\mathcal{M}'$ and randomness space $\mathcal{R}'$. PKE_1 is *lossy* if it has the following properties:

- PKE_1 is correct according to Definition 2.1.
- *Key indistinguishability:* We say PKE_1 has key indistinguishability if there is an algorithm LKG_1 such that, for any adversary $\mathcal{B}$, the advantage function

$$\mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{PKE}_1}(\mathcal{B}) := |\Pr[\mathcal{B}(\mathsf{pk}_1) \Rightarrow 1] - \Pr[\mathcal{B}(\mathsf{lpk}_1) \Rightarrow 1]|$$

  is negligible, where $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}_1$ and $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1$.
- *Lossiness:* Let $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1$ and m, m' be arbitrary messages in $\mathcal{M}'$, the statistical distance between $\mathsf{Enc}_1(\mathsf{lpk}_1, m)$ and $\mathsf{Enc}_1(\mathsf{lpk}_1, m')$ is negligible.
- *Weak Openability:* Let $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{KG}_1$, m and m' be arbitrary messages, and r be arbitrary randomness. For ciphertext $c := \mathsf{Enc}_1(\mathsf{lpk}_1, m; r)$, there exists an algorithm open_1 such that $\mathsf{open}_1(\mathsf{lsk}_1, \mathsf{lpk}_1, c, r, m')$ outputs r' where $c = \mathsf{Enc}_1(\mathsf{lpk}_1, m'; r')$ and r' is distributed uniformly. open_1 can be inefficient.

The lossiness definition can be extended to a multi-challenge version using a hybrid argument. Since it is only a statistical property, the hybrid argument will not affect tightness of the computational advantage.

Definition 5.2 (Multi-Challenge Lossiness). Let $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1$, μ be the number of challenge, and $m_1, m'_1, ..., m_\mu, m'_\mu$ be arbitrary messages in $\mathcal{M}'$. Multi-challenge *Lossiness* requires that statistical distance between $\{\mathsf{Enc}_1(\mathsf{lpk}_1, m_i)\}_{i\in[\mu]}$ and $\{\mathsf{Enc}_1(\mathsf{lpk}_1, m'_i)\}_{i\in[\mu]}$ is negligible. We write the distance as $\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1}$.

5.1 Construction

Let $\mathsf{PKE}_1 = (\mathsf{KG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$ be a lossy encryption with message space $\mathcal{M}'$, randomness space $\mathcal{R}'$, ciphertext space $\mathcal{C}'$, and an opening algorithm open_1. Let $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ be a MAC scheme with key space $\mathcal{K}^{\mathtt{mac}}$, and $G : \mathcal{M}' \to \mathcal{R}', H : \mathcal{R}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ be two hash functions. Our PKE scheme $\mathsf{sPKE} = (\mathsf{sKG}, \mathsf{sEnc}, \mathsf{sDec})$ is defined in Figure 12, which has the same structure with the scheme in Figure 11.

$\underline{\mathsf{sKG}}$	$\underline{\mathsf{sEnc}(\mathsf{pk} = \mathsf{pk}_1, m \in \mathcal{M})}$	$\underline{\mathsf{sDec}((\mathsf{sk}_1, k), (e, d, \tau))}$
01 $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}_1$	06 $r \xleftarrow{\$} \mathcal{M}'$	12 $r' := \mathsf{Dec}_1(\mathsf{sk}_1, e)$
02 $k \xleftarrow{\$} \mathcal{M}'$	07 $e := \mathsf{Enc}_1(\mathsf{pk}_1, r; G(r))$	13 **if** $r' = \perp$
03 $\mathsf{pk} := \mathsf{pk}_1$	08 $(K, K^{\mathtt{mac}}) := H(r, e)$	**or** $e \neq \mathsf{Enc}_1(\mathsf{pk}_1, r'; G(r'))$
04 $\mathsf{sk} := (\mathsf{sk}_1, k)$	09 $d := K \oplus m$	14 $(K, K^{\mathtt{mac}}) := H(k, e)$
05 **return** $(\mathsf{pk}, \mathsf{sk})$	10 $\tau \leftarrow \mathsf{Tag}(K^{\mathtt{mac}}, (e, d))$	15 **else** $(K, K^{\mathtt{mac}}) := H(r', e)$
	11 **return** (e, d, τ)	16 **if** $\mathsf{Vrfy}(K^{\mathtt{mac}}, (e, d), \tau) = 1$
		17 $m := K \oplus d$
		18 **else** $m := \perp$
		19 **return** m

Figure 12: A PKE scheme $\mathsf{sPKE} = (\mathsf{sKG}, \mathsf{sEnc}, \mathsf{sDec})$ based on lossy encryption PKE_1.

Theorem 5.3 shows that sPKE is tightly SO-CCA secure when modeling G and H as QROs. Although there is a loss μ to the otSUF-CMA security of the underlying MAC, if one can use a perfectly otSUF-CMA secure MAC (e.g., the efficient one implicitly in [KPW15]), it will not affect the security loss of sPKE and thus sPKE is tight.

Theorem 5.3 *If PKE_1 is a lossy encryption scheme and $(1-\delta)$-correct, and MAC is otSUF-CMA secure, then the PKE scheme sPKE in Figure 12 is adaptive SO-CCA secure (Definition 2.5). Concretely, for security parameter λ and integer $\mu := \mu(\lambda)$ (which is polynomially bounded) for any SO-CCA adversary $\mathcal{A}$ and relation Rel, there exist a simulator $\mathcal{S}$ and an adversary $\mathcal{F}$ with $\mathbf{T}(\mathcal{S}) \approx \mathbf{T}(\mathcal{A})$, $\mathbf{T}(\mathcal{F}) \approx \mathbf{T}(\mathcal{A})$, and*

$$\mathsf{Adv}^{\mathsf{SO\text{-}CCA}}_{\mathsf{sPKE}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel})$$

$$
\begin{aligned}
&\leq \mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{PKE}_1}(\mathcal{A}) + 3\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) \\
&\quad + 6(n_{\mathrm{OP}}+1)^2 q \sqrt{\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1} + \frac{\mu q}{|\mathcal{M}'|}} + 16(\mu + n_{\mathrm{DEC}} + q + 1)^2\delta \\
&\quad + \frac{(2+\mu)q}{\sqrt{|\mathcal{M}'|}} + \frac{6\mu q}{\sqrt{|\mathcal{R}'|}} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{\mu^2}{\mathcal{R}'} + \frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + \frac{\mu n_{\mathrm{DEC}}}{|\mathcal{C}' - n_{\mathrm{DEC}}|} + \frac{\mu^2}{|\mathcal{M}|}
\end{aligned}
$$

where μ, $q_G, q_H, n_{\mathrm{OP}}$, and n_{DEC} are the maximum numbers of $\mathcal{A}$'s challenge ciphertexts, $\mathcal{A}$'s queries to G, H, OPEN, and DEC, respectively. $q = q_G + q_H$.

The proof of Theorem 5.3 is given in Supp. Mat. D. Roughly, we firstly use the encrypt-then-hash technique [JZC+18, KSS+20, SXY18] to change security games so that the simulator can simulate decryption oracle without using secret key. Then, we switch the public key of PKE_1 to the lossy mode. By the key indistinguishability of PKE_1, the adversary cannot detect such modification, and the simulation of decryption oracle still works. However, we cannot use the lossiness of PKE_1 now, since there are several correlations between challenge ciphertexts and the QROs. Therefore, at the end of the proof, we use our adaptive reprogramming framework in Section 3 and delayed analysis to derelate QROs and challenge ciphertexts, and argue that the adversary cannot learn any information of unopened challenge ciphertexts.

INSTANTIATION FROM LWE. The Regev encryption scheme as defined in [GPV08] is essentially a lossy encryption, and we can use it to instantiate our generic construction in Figure 12. For completeness, we describe the lossy encryption in Supp. Mat. E. Our resulting SO-CCA secure PKE is unfortunately only almost tight, since the LWE-based lossy encryption loses a factor depending on the security parameter.

6 Bi-SO security in the QROM

In this section, we show that two PKE schemes are Bi-SO-CCA secure in the QROM. The first scheme is based on a modular FO transformation $\mathsf{FO}^{\not\perp}$ [JZC+18, HHK17] (Section 6.1). The second scheme is based on another modular FO transformation $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ [HHK17] (Section 6.2).

6.1 Bi-SO Security of $\mathsf{FO}^{\not\perp}$

We show that a multi-user version of sPKE (Figure 11) is Bi-SO-CCA-secure in the QROM. Using the same building blocks $\mathsf{PKE} = (\mathsf{KG}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ and MAC as sPKE, we propose $\mathsf{sPKE}_{\mathsf{bi}}$ (in Figure 13). This scheme can be viewed as a combination of a modular FO transformation $\mathsf{FO}^{\not\perp}[\mathsf{PKE}, G, H]$ in [JZC+18, HHK17], one-time pad, and the a MAC scheme MAC. Moreover, in $\mathsf{sPKE}_{\mathsf{bi}}$, each user includes its public key as an input to the hash functions G, H, H'.

Theorem 6.1 shows that $\mathsf{sPKE}_{\mathsf{bi}}$ is Bi-SO-CCA secure when modeling G and H as QROs. The proof of Theorem 6.1 is more complicated than the proofs of Theorem 4.2, since we also need to simulate CORRUPT oracle. But the proof idea is similar: we change the games so that the game simulator can use the encrypt-then-hash technique to simulate DEC (as we did in the proof of Theorem 4.2). To use our framework, we divide $\mathcal{A}_1$ with respect to CORRUPT and DEC, since the operations of CORRUPT also reprograms $G \times H$. The proof of Theorem 6.1 is given in Supp. Mat. G.

Theorem 6.1 *If PKE is OW-CPA secure, then the PKE scheme $\mathsf{sPKE}_{\mathsf{bi}}$ in Figure 13 is adaptive Bi-SO-CCA secure (Definition 2.6). Concretely, for any Bi-SO-CCA adversary $\mathcal{A}$ and relation Rel, there exist a simulator $\mathcal{S}$ and adversaries $\mathcal{B}^{ow}$ and $\mathcal{F}$ such that $\mathbf{T}(\mathcal{S}) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}^{ow}) \approx \mathbf{T}(\mathcal{F})$ and*

$$
\begin{aligned}
&\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}CCA}}_{\mathsf{sPKE}_{\mathsf{bi}}}(\mathcal{A}, \mathcal{S}, p, \mu, \mathsf{Rel}) \\
&\leq 6(n_{\mathrm{CO}} + n_{\mathrm{OP}} + 1)^2 q \sqrt{2p\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{ow}) + p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} + p\eta_{\mathsf{KG}_0} \\
&\quad + 3p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + \frac{p\mu n_{\mathrm{DEC}}}{|\mathcal{C}'| - n_{\mathrm{DEC}}} + \frac{p^2\mu^2 + p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{\mathcal{R}'} + \frac{p^2\mu^2}{|\mathcal{M}|} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} \\
&\quad + \frac{6p\mu q}{\sqrt{|\mathcal{R}'|}} + 16p(\mu + n_{\mathrm{DEC}} + q + q_{H'} + 1)^2\delta + \frac{2(n_{\mathrm{CO}}+1)^2\sqrt{pq_{H'}} + 2pq_{H'} + p\mu q}{\sqrt{|\mathcal{M}'|}}
\end{aligned}
$$

$\underline{\mathsf{sKG}_{\mathsf{bi}}}$	$\underline{\mathsf{sEnc}_{\mathsf{bi}}(\mathsf{pk}, m \in \mathcal{M})}$	$\underline{\mathsf{sDec}_{\mathsf{bi}}((\mathsf{pk},\mathsf{sk},k),(e,d,\tau))}$
01 $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}_0$	06 $r \xleftarrow{\$} \mathcal{M}'$	12 $r' := \mathsf{Dec}_0(\mathsf{sk}, e)$
02 $k \xleftarrow{\$} \mathcal{M}'$	07 $e := \mathsf{Enc}_0(\mathsf{pk}, r; G(\mathsf{pk}, r))$	13 **if** $r' = \bot$
03 $\mathsf{pk}' := \mathsf{pk}$	08 $(K, K^{\mathtt{mac}}) := H(\mathsf{pk}, r, e)$	14 **or** $e \neq \mathsf{Enc}_0(\mathsf{pk}, r'; G(\mathsf{pk}, r'))$
04 $\mathsf{sk}' := (\mathsf{pk},\mathsf{sk},k)$	09 $d := K \oplus m$	15 $(K, K^{\mathtt{mac}}) := H'(\mathsf{pk}, k, e)$
05 **return** $(\mathsf{pk}',\mathsf{sk}')$	10 $\tau \leftarrow \mathsf{Tag}(K^{\mathtt{mac}}, d)$	16 **else** $(K, K^{\mathtt{mac}}) := H(\mathsf{pk}, r', e)$
	11 **return** (e, d, τ)	17 **if** $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
		18 $m := K \oplus d$
		19 **else** $m := \bot$
		20 **return** m

Figure 13: A Bi-SO-CCA secure PKE scheme $\mathsf{sPKE}_{\mathsf{bi}} = (\mathsf{sKG}, \mathsf{sEnc}, \mathsf{sDec})$

where p, μ, $q_G, q_H, q_{H'}, n_{\mathrm{OP}}, n_{\mathrm{CO}}$, and n_{DEC} are the number of user in the games, the maximal number of challenge ciphertexts per users, $\mathcal{A}$'s queries to $G, H, H', \textsc{Open}$, $\textsc{Corrupt}$, and $\textsc{Dec}$, respectively. $q = q_G + q_H$.

6.2 Bi-SO security of $\mathsf{U}^{\not\perp}_{\mathsf{m}}$

Let $\mathsf{PKE} = (\mathsf{KG}_0, \mathsf{Enc}_0, \mathsf{Dec}_0)$ be a deterministic PKE scheme with public space $\mathcal{PK}'$, plaintext space $\mathcal{M}'$, ciphertext space $\mathcal{C}'$, and plaintext distribution $\mathcal{D}_{\mathcal{M}'}$. Lett MAC be a one-time MAC as in $\mathsf{sPKE}_{\mathsf{bi}}$. Let $H, H' : \mathcal{PK}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ be two hash functions. We define $\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}$ as in Figure 14. $\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}$ can be viewed as a combination of $\mathsf{U}^{\not\perp}_{\mathsf{m}}$ [HHK17], one-time pad and one-time MAC. Similar to $\mathsf{sPKE}_{\mathsf{bi}}$, each user includes its public key into the input of hash functions.

$\underline{\mathsf{sKG}^{\mathsf{m}}_{\mathsf{bi}}}$	$\underline{\mathsf{sEnc}^{\mathsf{m}}_{\mathsf{bi}}(\mathsf{pk}, m \in \mathcal{M})}$	$\underline{\mathsf{sDec}^{\mathsf{m}}_{\mathsf{bi}}((\mathsf{pk},\mathsf{sk},k),(e,d,\tau))}$
01 $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}_0$	06 $r \leftarrow \mathcal{D}_{\mathcal{M}'}$	12 $r' = \mathsf{Dec}_0(\mathsf{sk}, e)$
02 $k \xleftarrow{\$} \mathcal{M}'$	07 $e := \mathsf{Enc}_0(\mathsf{pk}, r)$	13 **if** $r' = \bot$
03 $\mathsf{pk}' := \mathsf{pk}$	08 $(K, K^{\mathtt{mac}}) := H(\mathsf{pk}, r)$	14 $(K, K^{\mathtt{mac}}) := H'(\mathsf{pk}, k, e)$
04 $\mathsf{sk}' := (\mathsf{pk},\mathsf{sk},k)$	09 $d := K \oplus m$	15 **else** $(K, K^{\mathtt{mac}}) := H(\mathsf{pk}, r')$
05 **return** $(\mathsf{pk}',\mathsf{sk}')$	10 $\tau \leftarrow \mathsf{Tag}(K^{\mathtt{mac}}, d)$	16 **if** $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
	11 **return** (e, d, τ)	17 $m = K \oplus d$
		18 **else** $m = \bot$
		19 **return** m

Figure 14: A Bi-SO-CCA secure PKE scheme $\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}} = (\mathsf{sKG}^{\mathsf{m}}_{\mathsf{bi}}, \mathsf{sEnc}^{\mathsf{m}}_{\mathsf{bi}}, \mathsf{sDec}^{\mathsf{m}}_{\mathsf{bi}})$

Here we consider a variant of OW-CPA security: $\mathcal{D}_{\mathcal{M}'}$-OW-CPA security, namely, OW-CPA security with challenge messages chosen following $\mathcal{D}_{\mathcal{M}'}$. The definition of $\mathcal{D}_{\mathcal{M}'}$-OW-CPA is given in Definition H.1. Moreover, we require that PKE is *rigid* correct [BP18], namely, for all $(\mathsf{pk}, \mathsf{sk})$ generated from KG_0, ciphertext e, and plaintext r, $(e = \mathsf{Enc}_0(\mathsf{pk}, r))$ if and only if $(\mathsf{Dec}_0(\mathsf{sk}, e) = r)$. Theorem 6.2 shows that $\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}$ is Bi-SO-CCA secure when modeling G and H as QROs. The proof of Theorem 6.2 is similar to Theorem 6.1, and is given in Supp. Mat. H.

Theorem 6.2 *Let PKE be a deterministic PKE with perfect correctness and rigidity. If PKE is $\mathcal{D}_{\mathcal{M}'}$-OW-CPA secure, then the PKE scheme $\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}$ in Figure 14 is adaptive Bi-SO-CCA secure (Definition 2.6). Concretely, for any Bi-SO-CCA adversary $\mathcal{A}$ and relation Rel, there exist a simulator $\mathcal{S}$ and adversaries $\mathcal{B}^{ow}$ and $\mathcal{F}$ such that $\mathbf{T}(\mathcal{S}) \approx \mathbf{T}(\mathcal{A}) \approx \mathbf{T}(\mathcal{B}^{ow}) \approx \mathbf{T}(\mathcal{F})$ and*

$$
\begin{aligned}
&\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}CCA}}_{\mathsf{sPKE}_{\mathsf{bi}}}(\mathcal{A}, \mathcal{S}, p, \mu, \mathsf{Rel}) \\
&\leq 6(n_{\mathrm{CO}} + n_{\mathrm{OP}} + 1)^2 q \sqrt{2p\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}'}}(\mathcal{B}^{ow}) + p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} \\
&\quad + 3p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + \frac{6p\mu q}{2^{\epsilon_{\mathcal{D}_{\mathcal{M}'}}}} + \frac{p\mu n_{\mathrm{DEC}}}{|\mathcal{C}'| - n_{\mathrm{DEC}}} + \frac{p^2\mu^2 + p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{|\mathcal{M}|} \\
&\quad + p\eta_{\mathsf{KG}_0} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + \frac{2(n_{\mathrm{CO}} + 1)^2 \sqrt{pq_{H'}} + 2pq_{H'} + p\mu q}{\sqrt{|\mathcal{M}'|}}
\end{aligned}
$$

where p, μ, $q_H, q_{H'}, n_{\text{OP}}, n_{\text{CO}}$, *and* n_{DEC} *are the maximum numbers of user in the games and* $\mathcal{A}$*'s challenge ciphertexts per users,* $\mathcal{A}$*'s queries to* H, H', OPEN, CORRUPT, *and* DEC, *respectively.* $\epsilon_{\mathcal{D}_{\mathcal{M}'}}$ *is the minimum entropy of* $\mathcal{D}_{\mathcal{M}'}$.

References

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. pages 269–295, 2019. (Cited on page 3, 4, 23.)

[AP08] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. Cryptology ePrint Archive, Report 2008/521, 2008. https://eprint.iacr.org/2008/521. (Cited on page 38.)

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. pages 41–69, 2011. (Cited on page 9, 11, 25.)

[BDWY12] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. pages 645–662, 2012. (Cited on page 2.)

[BHK12] Florian Böhl, Dennis Hofheinz, and Daniel Kraschewski. On definitions of selective opening security. pages 522–539, 2012. (Cited on page 1, 2.)

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. pages 1–35, 2009. (Cited on page 1, 2, 3, 4, 5, 16.)

[BP18] Daniel J. Bernstein and Edoardo Persichetti. Towards KEM unification. Cryptology ePrint Archive, Report 2018/526, 2018. https://ia.cr/2018/526. (Cited on page 19, 49.)

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. pages 409–426, 2006. (Cited on page 5.)

[CDH+20] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa. NTRU. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. (Cited on page 2, 49.)

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. pages 602–631, 2020. (Cited on page 5.)

[DKR+20] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. SABER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. (Cited on page 2.)

[FHKW10] Serge Fehr, Dennis Hofheinz, Eike Kiltz, and Hoeteck Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. pages 381–402, 2010. (Cited on page 2, 4.)

[FO99a] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. pages 53–68, 1999. (Cited on page 2.)

[FO99b] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. pages 537–554, 1999. (Cited on page 2.)

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. 26(1):80–101, January 2013. (Cited on page 2, 3.)

[GHHM21] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. pages 637–667, 2021. (Cited on page 3, 4, 5, 11, 23, 28.)

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. pages 197–206, 2008. (Cited on page 18, 37, 38.)

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. pages 341–371, 2017. (Cited on page 2, 3, 15, 18, 19.)

[HJKS15] Felix Heuer, Tibor Jager, Eike Kiltz, and Sven Schäge. On the selective opening security of practical public-key encryption schemes. pages 27–51, 2015. (Cited on page 2, 3, 7, 15.)

[HJR16] Dennis Hofheinz, Tibor Jager, and Andy Rupp. Public-key encryption with simulation-based selective-opening security and compact ciphertexts. pages 146–168, 2016. (Cited on page 3, 16.)

[HKSU20] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. pages 389–422, 2020. (Cited on page 34.)

[HLOV11] Brett Hemenway, Benoît Libert, Rafail Ostrovsky, and Damien Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. pages 70–88, 2011. (Cited on page 2.)

[Hof12] Dennis Hofheinz. All-but-many lossy trapdoor functions. pages 209–227, 2012. (Cited on page 2.)

[HP16] Felix Heuer and Bertram Poettering. Selective opening security from simulatable data encapsulation. pages 248–277, 2016. (Cited on page 2.)

[HR14] Dennis Hofheinz and Andy Rupp. Standard versus selective opening security: Separation and equivalence results. pages 591–615, 2014. (Cited on page 2.)

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. pages 96–125, 2018. (Cited on page 2, 3, 12, 15, 16, 18.)

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. pages 552–586, 2018. (Cited on page 11, 12.)

[KPW15] Eike Kiltz, Jiaxin Pan, and Hoeteck Wee. Structure-preserving signatures from standard assumptions, revisited. pages 275–295, 2015. (Cited on page 17.)

[KSS+20] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. pages 703–728, 2020. (Cited on page 2, 3, 4, 10, 16, 18, 25.)

[LLHG18] Lin Lyu, Shengli Liu, Shuai Han, and Dawu Gu. Tightly SIM-SO-CCA secure public key encryption from standard assumptions. pages 62–92, 2018. (Cited on page 2, 4, 7.)

[LYHW21] Junzuo Lai, Rupeng Yang, Zhengan Huang, and Jian Weng. Simulation-based bi-selective opening security for public key encryption. pages 456–482, 2021. (Cited on page 1, 2, 3, 7.)

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. pages 700–718, 2012. (Cited on page 37.)

[NC16] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information (10th Anniversary edition)*. Cambridge University Press, 2016. (Cited on page 8, 23, 26.)

[PWZ23] Jiaxin Pan, Benedikt Wagner, and Runzhi Zeng. Tighter security for generic authenticated key exchange in the qrom. In *ASIACRYPT 2023*, LNCS. Springer, Heidelberg, December 2023. https://eprint.iacr.org/2023/1380. (Cited on page 3, 34, 45.)

[PZ22] Jiaxin Pan and Runzhi Zeng. Compact and tightly selective-opening secure public-key encryption schemes. pages 363–393, 2022. (Cited on page 3.)

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. pages 84–93, 2005. (Cited on page 37.)

[SAB+20] Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions. (Cited on page 2.)

[SS19] Shingo Sato and Junji Shikata. SO-CCA secure PKE in the quantum random oracle model or the quantum ideal cipher model. pages 317–341, 2019. (Cited on page 2, 3, 4, 5, 11, 15.)

[SS22] Shingo Sato and Junji Shikata. SO-CCA secure PKE in the quantum random oracle model or the quantum ideal cipher model. Cryptology ePrint Archive, Paper 2022/617, 2022. (Retrieved: 2022-07-21). (Cited on page 2, 5.)

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. pages 520–551, 2018. (Cited on page 2, 9, 10, 12, 16, 18.)

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. pages 192–216, 2016. (Cited on page 2.)

[Unr14a] Dominique Unruh. Quantum position verification in the random oracle model. pages 1–18, 2014. (Cited on page 3, 4, 9, 10, 23, 24, 25, 28.)

[Unr14b] Dominique Unruh. Revocable quantum timed-release encryption. pages 129–146, 2014. (Cited on page 3, 4, 25.)

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. pages 568–597, 2021. (Cited on page 5.)

[Zha12] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. pages 758–775, 2012. (Cited on page 11.)

Supporting Material

A Review of Adaptive One-way-to-hiding

Let $\mathcal{HF} := \{\{0,1\}^* \to \{0,1\}^n\}$ be a set containing all functions that have $\{0,1\}^*$ as domain and $\{0,1\}^n$ as codomain. Let $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ be an adversary that has quantum access to a QRO $\mathcal{H}$ and queries it at most $q_0 + q_1$ times. Unruh's adaptive OW2H lemma [Unr14a, Lemma 15] can be described as follows: let

$$
\begin{aligned}
P_0^{\mathcal{A}} &:= \Pr\left[b' = 1 : \mathcal{H} \xleftarrow{\$} \mathcal{HF}, m \leftarrow \mathcal{A}_0^{\mathcal{H}}(), x \xleftarrow{\$} \{0,1\}^l, b' \leftarrow \mathcal{A}_1^{\mathcal{H}}(x, \mathcal{H}(x||m))\right] \\
P_1^{\mathcal{A}} &:= \Pr\Big[b' = 1 : \mathcal{H} \xleftarrow{\$} \mathcal{HF}, m \leftarrow \mathcal{A}_0^{\mathcal{H}}(), x \xleftarrow{\$} \{0,1\}^l, \\
&\qquad\qquad B \xleftarrow{\$} \{0,1\}^n, b' \leftarrow \mathcal{A}_1^{\mathcal{H}}(x, B)\Big] \\
P_C &:= \Pr\Big[(x'||m') = (x||m) : \mathcal{H} \xleftarrow{\$} \mathcal{HF}, m \leftarrow \mathcal{A}_0^{\mathcal{H}}(), x \xleftarrow{\$} \{0,1\}^l, \\
&\qquad\qquad B \xleftarrow{\$} \{0,1\}^n, j \xleftarrow{\$} [q_0], x'||m' \xleftarrow{\$} C^{\mathcal{H}}(j, x, B)\Big]
\end{aligned}
$$

where q_0, q_1 are the numbers of time $\mathcal{A}_0, \mathcal{A}_1$ queries $\mathcal{H}$ respectively. C is an algorithm that has quantum access to $\mathcal{H}$ and on input (j, B, x), runs $\mathcal{A}_1^{\mathcal{H}}(x, B)$ until its j-th query, measures the QRO query in the computational basis, output the measurement outcome. Then

$$
\left|P_0^{\mathcal{A}} - P_1^{\mathcal{A}}\right| \leq 2q_1\sqrt{P_C} + q_0 2^{-l/2+2}
$$

The bound given in this adaptive OW2H lemma includes two parts: the first part is roughly the search bound of quantum adversaries to find a uniformly random x given $\mathcal{H}(x||m)$ (i.e., $q_0 2^{-l/2+2}$), and the second part is the advantage of $\mathcal{A}_1$ to distinguish two QROs: $\mathcal{H}_{(x||m)\to B}$ and $\mathcal{H}$, where $\mathcal{H}_{(x||m)\to B}$ is the same as $\mathcal{H}$ except that $\mathcal{H}_{(x||m)\to B}(x||m) = B$. Note that this advantage is described by the extracting algorithm C.

Unruh's adaptive OW2H lemma cannot be used to prove the bound of our reprogramming framework Figure 4 via hybrid arguments. This is because:

- The initial oracles of Ada and NonAda in our framework are not necessarily the same. In this case, our framework considers a stronger QROM adaptive reprogramming setting than the adaptive OW2H (and the adaptive reprogramming framework in [GHHM21]).
- Even if the initial oracles are the same, in our framework, sets S_i may not independent to each other, and thus each intermediate hybrid games in the hybrid argument may not independent. This makes it hard to modify the adaptive OW2H lemma to fit in our framework and use hybrid argument. More details will be given in Remark C.1.

B More Background about Quantum Computation

Trace Distance. Trace distance (TD) is used to measure how "close" two quantum states are, informally, the distance between the distributions of their measurement outcome. For pure states $|x_1\rangle, |x_2\rangle$, $\mathsf{TD}[|x_1\rangle, |x_2\rangle] = \mathsf{TD}[|x_1\rangle\langle x_1|, |x_2\rangle\langle x_2|]$. Following [NC16], if $\{\alpha_i\}_i$ (e.g., the distribution of some random variable α that $\Pr[\alpha = i] = \alpha_i$) and $\{\alpha'_i\}_i$ are two distributions with the same index set, then we write $\mathsf{TD}[\{\alpha_i\}_i, \{\alpha'{}_i\}_i]$ as the L_1-distance of the two distributions:

$$
\mathsf{TD}\left[\{\alpha_i\}_i, \{\alpha'{}_i\}_i\right] = \frac{1}{2}\sum_i |\alpha_i - \alpha'_i|
$$

Here introduces some lemmas used in the proof of Lemma 3.1. For pure states, Lemma B.1 shows that the euclidean distance of two pure states bound their trace distance.

Lemma B.1 (Lemmas 3 and 4 in [AHU19]). *If $|\phi_0\rangle$ and $|\phi_1\rangle$ are two pure quantum states, then* $\mathsf{TD}[|\phi_0\rangle, |\phi_1\rangle] \leq \||\phi_0\rangle - |\phi_1\rangle\|$

We are interested in the trace distance of two pure states that obtains from an algorithm $\mathcal{A}$ interacts with different QROs, where these QROs have the same output distribution except on some specific points. Lemma B.2 and Lemma B.3 give bounds of such trace distances.

Lemma B.2 *Let $|\varphi\rangle$ be a quantum state with registers X, Y (storing elements of sets $\mathcal{X}$ and $\mathcal{Y}$) and S be a subset of $\mathcal{X}$. Let f' and f be two functions with the same preimage set $\mathcal{X}$ and image set $\mathcal{Y}$ such that for all $x \in \mathcal{X}\backslash S$, $f(x) = f'(x)$. Let $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ and $U_{f'} : |x, y\rangle \to |x, y \oplus f'(x)\rangle$ be the unitary transformations corresponding to f and f', respectively. Then we have*

$$\mathsf{TD}\left[U_f|\varphi\rangle, U_{f'}|\varphi\rangle\right] \leq 2\|P_S|\varphi\rangle\|$$

where P_S is a projector that projects the the content of X-register of $|\varphi\rangle$ into the subspace spanned by S, i.e., $P_S = \sum_{t\in S}|t\rangle\langle t|$.

Lemma B.2. For any state $|x\rangle$ (in the same space as $|\varphi\rangle$), we have $U_f(I - P_S)|x\rangle = U_{f'}(I - P_S)|x\rangle$ by conditions of Lemma B.2. This is because $(I - P_S)$ projects the input state onto the orthogonal complement of the subspace spanned by S. Moreover, $(U_f - U_{f'})$ are norm-2 operator since both U_f and $U_{f'}$ are unitary. Therefore, by Lemma B.1,

$$\mathsf{TD}\left[U_f|\varphi\rangle, U_{f'}|\varphi\rangle\right] \leq \|U_f|\varphi\rangle - U_{f'}|\varphi\rangle\| = \|(U_f - U_{f'})P_S|\varphi\rangle\| \leq 2\||\varphi\rangle\|$$

□

Lemma B.3 *Let $|\varphi_0\rangle$ be a quantum state with registers X, Y (storing elements of sets $\mathcal{X}$ and $\mathcal{Y}$) and S be a subset of $\mathcal{X}$. Let f' and f be two functions with the same preimage set $\mathcal{X}$ and image set $\mathcal{Y}$ such that for all $x \in \mathcal{X}\backslash S$, $f(x) = f'(x)$. Let $U_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$ and $U_{f'} : |x, y\rangle \to |x, y \oplus f'(x)\rangle$ be the unitary transformations corresponding to f and f', respectively. We consider two quantum states:*

$$\begin{aligned}|\phi\rangle &:= T_q U_f T_{q-1} U_f ... T_1 U_f T_0 |\varphi_0\rangle \\ |\psi\rangle &:= T_q U_{f'} T_{q-1} U_{f'} ... T_1 U_{f'} T_0 |\varphi_0\rangle,\end{aligned}$$

where $T_q, ..., T_0$ are unitaries. Then we have

$$\mathsf{TD}\left[|\phi\rangle, |\psi\rangle\right] \leq \sum_{i=0}^{q-1} 2\|P_S T_i U_{f'} ... T_1 U_{f'} T_0 |\varphi_0\rangle\|$$

where P_S is a projector that projects the the content of X-register of $|\varphi_0\rangle$ into the subspace spanned by S, i.e., $P_S = \sum_{t\in S}|t\rangle\langle t|$.

Lemma B.3. This proof is similar to the proof of Unruh's OW2H lemma in [Unr14a]. Let $|\phi_i\rangle := T_i U_f ... T_1 U_f T_0 |\varphi_0\rangle$ and $|\psi_i\rangle := T_i U_{f'} ... T_1 U_{f'} T_0 |\varphi_0\rangle$ for $0 \leq i \leq q-1$, and let $|\psi_0\rangle = T_0|\varphi_0\rangle = |\phi_0\rangle$. We have $|\phi_i\rangle = T_i U_f |\phi_{i-1}\rangle$ and $|\psi_i\rangle := T_i U_{f'} |\psi_{i-1}\rangle$ for $1 \leq i \leq q$, and

$$\begin{aligned}\mathsf{TD}\left[|\phi_{i+1}\rangle, |\psi_{i+1}\rangle\right] &= \mathsf{TD}\left[T_i U_f |\phi_i\rangle, T_i U_{f'} |\psi_i\rangle\right] \\ &= \mathsf{TD}\left[U_f|\phi_i\rangle, U_{f'}|\psi_i\rangle\right] && \text{(Unitary preserves TD)} \\ &\leq_{(*)} \mathsf{TD}\left[U_f|\phi_i\rangle, U_f|\psi_i\rangle\right] + \mathsf{TD}\left[U_f|\psi_i\rangle, U_{f'}|\psi_i\rangle\right] \\ &\leq \mathsf{TD}\left[|\phi_i\rangle, |\psi_i\rangle\right] + 2\|P_S|\psi_i\rangle\|, && \text{(By Lemma B.2)}\end{aligned}$$

where $(*)$ is by triangle inequality. Therefore, we have

$$\begin{aligned}\mathsf{TD}\left[|\phi\rangle, |\psi\rangle\right] &= \mathsf{TD}\left[|\phi_q\rangle, |\psi_q\rangle\right] - \mathsf{TD}\left[|\phi_0\rangle, |\psi_0\rangle\right] (= 0) \\ &\leq \sum_{i=0}^{q-1} 2\|P_S|\psi_i\rangle\| = \sum_{i=0}^{q-1} 2\|P_S T_i U_{f'} ... T_1 U_{f'} T_0 |\varphi_0\rangle\|\end{aligned}$$

□

MIXED STATES AND DENSITY OPERATORS. Mixed quantum states will be described using density operators. If a quantum system is in state $|x_i\rangle$ with probability p_i, then the density operator of this system can be written as $\sum_i p_i|x_i\rangle\langle x_i|$. Let Ψ and Φ be two density operators, the trace distance between Ψ and Φ is written as $\mathsf{TD}[\Psi,\Phi]$. For an quantum adversary $\mathcal{A}$, it can be modeled as a sequence of unitary transformations $U, U_{\mathcal{O}}, ..., U, U_{\mathcal{O}}$ [BDF+11, Unr14b], where U is the transition unitary of $\mathcal{A}$ (we can also model it as $U_n, U_{\mathcal{O}}, ..., U_1, U_{\mathcal{O}}$, which does not influence our results in this paper). We directly write $\mathcal{O}$ to denote the unitary $U_{\mathcal{O}}$.

C Proof of Lemma 3.1

Before proving Lemma 3.1, we introduce some variables and notations. Following [Unr14a, Unr14b, KSS+20], we assume that $\mathcal{A}$ consists of three quantum registers A, X, and Y without loss of generality, where A is used to store $\mathcal{A}$'s internal state, and X and Y are used to store quantum random oracle queries.

- $\Gamma_i (i \in \{1, .., n\})$: $\Gamma_i = (\mathsf{out}_0, ..., \mathsf{out}_{i-1})$, where out_j is the output of $\mathcal{A}_j$ for $0 \leq j \leq i-1$. That is, Γ_i stores all outputs of $\mathcal{A}$'s first i-th stages (i.e., $(\mathcal{A}_0, ..., \mathcal{A}_{i-1})$). We define $\Gamma_{n+1} := \Gamma$ (the final output of games NONADA and ADA) and $\Gamma_0 := \emptyset$.
- U: the state transition unitary operation of $\mathcal{A}$.
- U_{in_i} (or simply U_i): the unitary transformation that models the operations of $\mathcal{A}$ when $\mathcal{A}$ receives in_i in its i-th stage. Without loss of generality, we can assume that $\mathsf{F}_{\mathbf{s}}$ is deterministic (since if F is not deterministic, we can include all randomness in $\mathbf{s}$). So, in_i is determined by out_i for fixed parameter $\mathbf{s}$. This means that, if we fixed $\mathcal{A}$'s output list Γ_i of its previous i stages (from 0 to $i-1$), then in_i and U_i are determined.
- $\mathcal{H}, \mathsf{H}, \mathsf{H}_i$: the unitary operations of quantum oracle access to QRO $\mathcal{H}$. Specifically, $\mathcal{H} : |a, x, y\rangle \to |a, x, y \oplus \mathcal{H}(x)\rangle$. In NONADA, $\mathcal{H} = \mathsf{H}$ and H is independent of Γ_i. While in ADA, $\mathcal{H} = \mathsf{H}_i$ in $\mathcal{A}$'s i-th stage (i.e., the QRO that $\mathcal{A}_i$ queries is H_i in ADA) and H_i is dependent to Γ_i. These unitary transformations do not influence registers A and X.
- $|\varphi_0\rangle, |\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle$ $(1 \leq i \leq n)$. Let $|\varphi_0\rangle$ be the initial state of $\mathcal{A}$ before receiving in_0. For fixed environment parameter $\mathbf{s}$, initial input in_0, and previous outputs Γ_i of $\mathcal{A}$, the final state of $\mathcal{A}_i$ ($\mathcal{A}$'s i-th stage) in NONADA is $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle$, and the final state of $\mathcal{A}_i$ in ADA is $|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle$. By the notations introduced above, $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle$ and $|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle$ can be written as:

$$|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle = (U\mathsf{H})^{q_i} U_i (U\mathsf{H})^{q_{i-1}} U_{i-1} ... (U\mathsf{H})^{q_0} U_0 |\varphi_0\rangle$$
$$|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_i}\rangle = (U\mathsf{H}_i)^{q_i} U_i (U\mathsf{H}_{i-1})^{q_{i-1}} U_{i-1} ... (U\mathsf{H}_0)^{q_0} U_0 |\varphi_0\rangle.$$

  Similarly, we define the final state of $\mathcal{A}_0$ in NONADA (resp., in ADA) as $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle := (U\mathsf{H})^{q_0} U_0 |\varphi_0\rangle$ (resp., $|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle := (U\mathsf{H}_0)^{q_0} U_0 |\varphi_0\rangle$). Without loss of generality, we define $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_{-1}}\rangle := |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{-1}}\rangle := |\varphi_0\rangle$ to deal with boundary cases.

  Here we do not write up the measurement operations of $\mathcal{A}$'s outputs in the state, this is because in our framework, we require that $\mathcal{A}$'s outputs are classical information.

We also define several probabilities that will be used in the proof:

- $\beta_{\mathbf{s},\mathsf{in}_0}$: We define $\beta_{\mathbf{s},\mathsf{in}_0} := \Pr[(\mathbf{s}, \mathsf{in}_0) = (\mathbf{s}', \mathsf{in}_0') : (\mathbf{s}', \mathsf{in}_0', \mathsf{H}, \mathsf{H}') \leftarrow \mathtt{INIT}]$ as the probability that $\mathtt{INIT}$ outputs a specific $(\mathbf{s}, \mathsf{in}_0)$.
- $\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_i)$ and $\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i)$ $(1 \leq i \leq n+1)$: $\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_i)$ is the probability that $\mathcal{A}$'s output list in NONADA is Γ_i right after its first i stages (i.e., $(\mathcal{A}_0, ..., \mathcal{A}_{i-1})$). Similarly, $\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i)$ is the probability that $\mathcal{A}$'s output list in ADA is Γ_i right after its first i stages. To deal with boundary cases, without loss of generality, we define $\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_0) = \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_0) = 1$.

- $\alpha_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i)$ and $\alpha'_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i)$ $(0 \le i \le n)$: for fixed $(\mathbf{s},\mathsf{in}_0)$ and Γ_i, $\alpha_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i)$ is the probability that $\mathcal{A}_i$ outputs out_i in NonAda conditioned on $(\mathbf{s},\mathsf{in}_0)$ and Γ_i. Similarly, $\alpha'_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i)$ is the probability that $\mathcal{A}_i$ outputs out_i in Ada conditioned on $(\mathbf{s},\mathsf{in}_0)$ and Γ_i. We also define $\alpha_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i,...,\mathsf{out}_j)$ (resp., $\alpha'_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i,...,\mathsf{out}_j)$) as the probability that, conditioned on $(\mathbf{s},\mathsf{in}_0)$ and Γ_i, $\mathcal{A}_i$ outputs out_i, $\mathcal{A}_{i+1}$ outputs out_{i+1} ..., and $\mathcal{A}_j$ outputs out_j in NonAda (resp., in Ada).

Lemma 3.1. Let Φ_n be the density operator (cf. Supp. Mat. B) of $\mathcal{A}$'s final state (which is also the final state of $\mathcal{A}_n$ so here we add the index n) in NonAda$^{\mathcal{A}}$ and Ψ_n be the density operator of $\mathcal{A}$'s final state in Ada$^{\mathcal{A}}$. By the notation introduced before, these operators can written as:

$$\Phi_n = \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\langle\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}| \tag{8}$$

$$\Psi_n = \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\langle\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}| \tag{9}$$

Similarly, we can define $\Phi_k (0 \le k \le n)$ be the density operator of $\mathcal{A}_k$'s final state in NonAda$^{\mathcal{A}}$ and Ψ_k be the density operator of $\mathcal{A}_k$'s final state in Ada$^{\mathcal{A}}$.

$$\Phi_k = \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0}\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_k)|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\langle\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}| \tag{10}$$

$$\Psi_k = \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k)|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\langle\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}| \tag{11}$$

Since the event Ev is a classical event, the probability that Ev happens in NonAda$^{\mathcal{A}}$ equals to the probability that a binary measurement on the final state of $\mathcal{A}$ outputs 1 (indicating the event occurs). Similarly, $\Pr\left[\mathsf{Ev} : \text{Ada}^{\mathcal{A}}\right]$ equals to the probability that such measurement outcome is 1. We still can use trace distance to bound the probability difference. By (8) and (9),

$$\begin{aligned}
&\left|\Pr\left[\mathsf{Ev} : \text{NonAda}^{\mathcal{A}}\right] - \Pr\left[\mathsf{Ev} : \text{Ada}^{\mathcal{A}}\right]\right| = \mathsf{TD}\left[\Phi_n, \Psi_n\right] \\
&= \mathsf{TD}\Big[\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\langle\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}|, \\
&\qquad\qquad \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\langle\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}|\Big] \\
&\le \mathsf{TD}\left[\{\beta_{\mathbf{s},\mathsf{in}_0}\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\}_{\mathbf{s},\mathsf{in}_0,\Gamma_n}, \{\beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\}_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\right] \\
&\qquad + \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\right],
\end{aligned} \tag{12}$$

where (12) comes from the strong convexity of trace distance [NC16, Theorem 9.3]. Here we give a brief explanation of Equation (12). The quantity on the left of "+" measures the distance between $\mathcal{A}$'s distributions of $\Gamma_n (= (\mathsf{out}_0,...,\mathsf{out}_{n-1}))$ in NonAda and Ada. The quantity on the right of "+" measures the probability difference between $\mathcal{A}$'s "behaviors" in NonAda and Ada if $\mathcal{A}$ has the same distribution of Γ_n in these two games.

The LHS trace distance bound the probability difference between $\mathcal{A}$'s output distributions of $(\mathsf{out}_0,...,\mathsf{out}_{n-1})$, and these outputs are determined when $\mathcal{A}$ completed its first n stages (i.e., $(\mathcal{A}_0,...,\mathcal{A}_{n-1})$). Therefore, this trace distance can be bounded by the trace distance between the final states of $\mathcal{A}_{n-1}$ in NonAda and Ada. That is,

$$\mathsf{TD}\left[\{\beta_{\mathbf{s},\mathsf{in}_0}\alpha_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\}_{\mathbf{s},\mathsf{in}_0,\Gamma_n}, \{\beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\}_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\right] \le \mathsf{TD}\left[\Phi_{n-1}, \Psi_{n-1}\right],$$

and so we have

$$\mathsf{TD}\left[\Phi_n, \Psi_n\right] \le \mathsf{TD}\left[\Phi_{n-1}, \Psi_{n-1}\right] + \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_n} \beta_{\mathbf{s},\mathsf{in}_0}\alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_n)\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_n}\rangle\right].$$

We can get similar inequalities for $1 \le k \le n$. For $k = 0$, we have $\mathsf{TD}\left[\Phi_0, \Psi_0\right] \le \sum_{\mathbf{s},\mathsf{in}_0} \beta_{\mathbf{s},\mathsf{in}_0} \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle\right]$. By a simple induction, we have

$$\mathsf{TD}\left[\Phi_n, \Psi_n\right] \le \sum_{k=0}^{n} \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right]. \tag{13}$$

To bound (13), we firstly fix k and focus on this quantity

$$\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right], \tag{14}$$

which can be bounded via applying Lemma B.3 $(k+1)$ times, since here $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle$ and $|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle$ have the same distribution of Γ_k. We firstly fix $\mathbf{s}, \mathsf{in}_0$ and Γ, and look at $\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right]$.

$$\begin{aligned}
&\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \\
&= \mathsf{TD}\left[(U\mathsf{H})^{q_k} U_k |\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle, (U\mathsf{H}_k)^{q_k} U_k |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle\right] \\
&\le \mathsf{TD}\left[(U\mathsf{H})^{q_k} U_k |\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle, (U\mathsf{H})^{q_k} U_k |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle\right] \\
&\quad + \mathsf{TD}\left[(U\mathsf{H})^{q_k} U_k |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle, (U\mathsf{H}_k)^{q_k} U_k |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle\right] \\
&\le \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle\right] + \sum_{j=0}^{q_k-1} 2\|P_{S_k}(U\mathsf{H}_k)^j U_k |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{k-1}}\rangle\|,
\end{aligned}$$

where the last inequality comes from Lemma B.3 (our framework assumed that $\mathsf{H}_k \backslash S_k = \mathsf{H} \backslash S_k$ for some set S_k). By induction, we have similar inequalities for $1 \le i \le k$, and thus

$$\begin{aligned}
\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \le\ & \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle\right] \\
& + \sum_{i=1}^{k} \sum_{j=0}^{q_k-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\|.
\end{aligned}$$

$\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle\right]$ can be also bounded by using Lemma B.3, since $|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle = (U\mathsf{H})^{q_0} U_0 |\varphi_0\rangle$ and $|\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_0}\rangle = (U\mathsf{H}_0)^{q_0} U_0 |\varphi_0\rangle$. Therefore,

$$\mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \le \sum_{i=0}^{k} \sum_{j=0}^{q_i-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\|. \tag{15}$$

Now we have

$$\begin{aligned}
&\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \\
&\le \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \sum_{i=0}^{k} \sum_{j=0}^{q_i-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\| \\
&= \sum_{i=0}^{k} \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \sum_{j=0}^{q_i-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\| \\
&= \sum_{i=0}^{k} \Big[\Big(\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_i} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i) \sum_{j=0}^{q_i-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\| \Big) \\
&\qquad \Big(\sum_{\mathsf{out}_i,\ldots,\mathsf{out}_{k-1}} \alpha'_{\mathbf{s},\mathsf{in}_0,\Gamma_i}(\mathsf{out}_i, \ldots, \mathsf{out}_{k-1}) \Big) (=1) \Big] \\
&= \sum_{i=0}^{k} \Big(\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_i} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i) \sum_{j=0}^{q_i-1} 2\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\| \Big) \\
&= \sum_{i=0}^{k} 2q_i \Big(\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_i} \sum_{j=0}^{q_i-1} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i) \frac{1}{q_i} \Big(\sqrt{\|P_{S_i}(U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle\|} \Big)^2 \Big)
\end{aligned}$$

$$\leq \sum_{i=0}^{k} 2q_i \sqrt{\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_i} \sum_{j=0}^{q_i-1} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i) \frac{1}{q_i} \| P_{S_i} (U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle \|^2}, \tag{16}$$

where (16) comes from Jensen's inequality for concave functions. Now consider the adversary $\mathcal{B}_i$ in Figure 5. $\mathcal{B}_i$ firstly chooses a uniformly random $t^* \leftarrow [q_i]$. Then it simulates ADA for $\mathcal{A}$ and performs projective measurement on $\mathcal{A}$'s t^* RO-query in $\mathcal{A}$'s i-th stage in computational basis. For fix $t^*, \mathbf{s}, \mathsf{in}_0$, the pure state of $\mathcal{A}$ right before the measurement is $(U\mathsf{H}_i)^{t^*-1} U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle$, and Γ_i, and the probability that the measurement outcome falls into S_i is $\| P_{S_i} (U\mathsf{H}_i)^{t^*-1} U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle \|^2$, so

$$\begin{aligned} &\sum_{\mathbf{s},\mathsf{in}_0,\Gamma_i} \sum_{j=0}^{q_i-1} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_i) \frac{1}{q_i} \| P_{S_i} (U\mathsf{H}_i)^j U_i |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_{i-1}}\rangle \|^2 \\ &= \Pr\left[x' \leftarrow \mathcal{B}_i^{\mathcal{H}} \text{ s.t. } x' \in S_i : \text{ADA}^{\mathcal{B}_i} \right]. \end{aligned} \tag{17}$$

Combining (13), (16), and (17), we get

$$\begin{aligned} \mathsf{TD}\left[\Phi_n, \Psi_n\right] &\leq \sum_{k=0}^{n} \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \\ &= \sum_{k=0}^{n} \sum_{\mathbf{s},\mathsf{in}_0,\Gamma_k} \beta_{\mathbf{s},\mathsf{in}_0} \alpha'_{\mathbf{s},\mathsf{in}_0}(\Gamma_k) \mathsf{TD}\left[|\phi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle, |\psi_{\mathbf{s},\mathsf{in}_0,\Gamma_k}\rangle\right] \\ &\leq \sum_{k=0}^{n} \sum_{i=0}^{k} 2q_i \Pr \sqrt{\left[x' \leftarrow \mathcal{B}_i^{\mathcal{H}} \text{ s.t. } x' \in S_i : \text{ADA}^{\mathcal{B}_i} \right]}. \end{aligned}$$

□

Remark C.1 We note that using Unruh's adaptive OW2H Lemma (cf. Supp. Mat. A and [Unr14a]) and a simple hybrid argument is not sufficient to prove our reprogramming framework. The main reason is that the intermediate hybrids cannot be simulated using Unruh's adaptive OW2H Lemma.

Consider a Hybrid $i \in \{0, ..., n\}$ where $(\mathcal{A}_0, ..., \mathcal{A}_i)$ are interacting with the same QRO as in NONADA (namely, H that has never been reprogrammed), and $(\mathcal{A}_{i+1}, ..., \mathcal{A}_n)$ are interacting with the same QRO as in ADA (namely, $\mathcal{A}_j$ interacts with H_j for $i+1 \leq j \leq n$ and H_j gets reprogrammed). To bound the difference between Hybrids i and $i+1$, we could consider using a "single-point" reprogramming framework, such as the adaptive OW2H Lemma [Unr14a]. However, it is unclear how the reduction can simulate $\mathsf{H}_{i+2}, ..., \mathsf{H}_n$ such that $\mathsf{H} \backslash S_j = \mathsf{H}_j \backslash S_j$ for $i+2 \leq j \leq n$, since the reduction may not know $S_{i+2}, ..., S_n$. The reduction does not always know these sets that can be arbitrary. For instance, it can be the case that $S_{i+2} \subset S_{i+1}$. Now, knowing S_{i+2}, the reduction already breaks the adaptive OW2H lemma without using $\mathcal{A}$.

The framework in [GHHM21] can be proven by hybrid argument because the proof of their lemma knows $S_0, ..., S_n$ (which are equivalent to $\mathfrak{L}_{\mathsf{O}}$ according to their notion).

D Proof of Theorem 5.3

Theorem 5.3. Let $h : \mathcal{M}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ and $g : \mathcal{M}' \to \mathcal{R}'$ be internal quantum-accessible ROs which are used to respond the queries to H and G, respectively. Similar to the proof of Theorem 4.1, to match the syntax of our framework, we combine G and H as one random oracle $G \times H$ where $G \times H(r', r, e) := (G(r'), H(r, e))$. $\mathcal{A}$ can query $G \times H$ at most $q = q_H + q_G$ times.

During the proof, we implicitly assume that $\mathcal{A}$ will not query DEC on (e, d, τ) with $(e, d) = (e_i, d_i)$ before seeing the challenge ciphertexts $\mathbf{c}$. Since r_i's are independent of $\mathcal{A}$'s view before it sees $\mathbf{c}$, the probability that $\mathcal{A}$ queries DEC on such ciphertexts is $\frac{\mu n_{\text{DEC}}}{|\mathcal{C}' - n_{\text{DEC}}|} + \frac{\mu q}{\sqrt{|\mathcal{M}'|}}$, where the second term is the bound to search $G(r_i)$ and $H(r_i)$ given quantum access to $G \times H$. Moreover, we also assume that there is no collision among outputs of r_i's, R_i's, K_i's, and $K_i^{\mathtt{mac}}$'s. This introduce collision bounds $\frac{\mu^2}{|\mathcal{M}'|} + \frac{\mu^2}{\mathcal{R}'} + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|}$. For simplicity, we just add these probability into our final bound, and do not consider it in the game sequences.

Games $\mathbf{G}_0$-$\mathbf{G}_5$

$(\mathsf{pk}_1, (\mathsf{sk}_1, k)) \leftarrow \mathsf{sKG}, \mathsf{pk} := \mathsf{pk}_1$
$\mathcal{M}_a \leftarrow \mathcal{A}^{\textsc{Dec}, |G \times H\rangle}(\mathsf{pk})$
for $i \in [\mu]$
  $\mathbf{m}[i] := m_i \leftarrow \mathcal{M}_a$
  $r_i \xleftarrow{\$} \mathcal{M}'$
  $R_i := G(r_i)$
  $e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i; R_i)$
  $(K_i, K_i^{\mathtt{mac}}) = H(r_i, e_i)$
  $d_i := K_i \oplus m_i$
  $\tau_i := \mathsf{Tag}(K_i^{\mathtt{mac}}, d_i)$
  $\mathbf{c}[i] := (e_i, d_i, \tau_i)$
$out \leftarrow \mathcal{A}^{\textsc{Open}, \textsc{Dec}, |G \times H\rangle}(\mathbf{c})$
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$H(r, e)$

if $e = \mathsf{Enc}_1(\mathsf{pk}, r; G(r))$ // $\mathbf{G}_3$-$\mathbf{G}_5$
  return $h_1(e)$ // $\mathbf{G}_3$-$\mathbf{G}_5$
return $h(r, e)$

$\textsc{Dec}(c = (e, d, \tau))$: for $c \notin \mathbf{c}$

$r' := \mathsf{Dec}_1(\mathsf{sk}, e)$ // $\mathbf{G}_0$-$\mathbf{G}_3$
if $r' = \bot$
  or $e \neq \mathsf{Enc}_1(\mathsf{pk}, r'; G(r'))$ // $\mathbf{G}_0$-$\mathbf{G}_3$
  $(K, K^{\mathtt{mac}}) := h(k, e)$ // $\mathbf{G}_0$
  $(K, K^{\mathtt{mac}}) := h'(e)$ // $\mathbf{G}_1$-$\mathbf{G}_3$
else $(K, K^{\mathtt{mac}}) := h(r', e)$ // $\mathbf{G}_0$-$\mathbf{G}_2$
else $(K, K^{\mathtt{mac}}) := h_1(e)$ // $\mathbf{G}_3$
$(K, K^{\mathtt{mac}}) := h_1(e)$ // $\mathbf{G}_4$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \bot$
return m

$\textsc{Open}(i)$

$I := I \cup \{i\}$
return (m_i, r_i)

$G(r)$

return $g(r)$ // $\mathbf{G}_0$-$\mathbf{G}_1$, $\mathbf{G}_5$
return $g'(r)$ // $\mathbf{G}_2$-$\mathbf{G}_4$

Figure 15: Games $\mathbf{G}_0$-$\mathbf{G}_5$ for the proof of Theorem 5.3.

In games $\mathbf{G}_0$-$\mathbf{G}_5$ (shown in Figure 15), we use the encrypt-than-hash technique so that the decryption oracle can be simulated without secret key. Since PKE_1 may not be perfectly-correct, we need games $\mathbf{G}_2$ and $\mathbf{G}_5$ to deal with the correctness of PKE when using the encrypt-than-hash technique.

In games $\mathbf{G}_6$-$\mathbf{G}_8$ (shown in Figure 16), we firstly switch the public key to a lossy key. Then, we use the framework in Lemma 3.1 to de-relate R_i from QROs G and H. Finally, by using multi-challenge lossiness of PKE_1, we bound the probability that $\mathcal{A}$ learns r_i before opening $\mathbf{c}[i]$.

Game $\mathbf{G}_0$: This game is equivalent to $\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}}$, so

$$\Pr[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}} \Rightarrow 1] = \Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1]$$

Game $\mathbf{G}_1$: The $\textsc{Dec}$ oracle computes $(K, K^{\mathtt{mac}}) = h'(e)$ rather than $h(k, e)$ if $r' = \bot$ or $e \neq \mathsf{Enc}_1(\mathsf{pk}_1, r'; g(r'))$, where $h' : \mathcal{M}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ is an internal quantum-accessible random oracles independent of h and g. By Lemma 2.7, we have

$$\left|\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]\right| \leq 2q_H/\sqrt{|\mathcal{M}'|}$$

Game $\mathbf{G}_2$: We restrict the range of G to be the "good" randomness space defined by $(\mathsf{pk}, \mathsf{sk})$. Namely, we define the set

$$\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1, \mathsf{sk}_1, r) := \{r' \in \mathcal{R}' \mid \mathsf{Dec}_1(\mathsf{sk}_1, \mathsf{Enc}_1(\mathsf{pk}_1, r; r')) = r\}$$

and let $g' : \mathcal{M}' \to \mathcal{R}'$ be a quantum-accessible random oracle such that $g'(r)$ is sampled uniformly from $\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1, \mathsf{sk}_1, r)$. If PKE is $(1-\delta)$-correct (see Definition 2.1), then

$$\left|\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]\right| \leq 8(\mu + n_{\textsc{Dec}} + q_G + q_H + 1)^2\delta.$$

The proof is given in Supp. Mat. D.1.

Game $\mathbf{G}_3$: We set $H(r, e) = h_1(e)$ if $e = \mathsf{Enc}_1(\mathsf{pk}_1, r; G(r))$, where $h_1 : \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathtt{mac}}$ is an internal quantum-accessible random oracle independent of h and h'. Since the randomness generated by G (i.e., g') is always a "good" randomness, $\mathsf{Enc}_1(\mathsf{pk}_1, \cdot; G(\cdot))$ is an injective function and thus $h_1(\mathsf{Enc}_1(\mathsf{pk}_1, \cdot; G(\cdot)))$ can be also viewed as an random oracle. Therefore, we have $\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1]$.

Game $\mathbf{G}_4$: We "merge" h_1 and h', namely, DEC always computes $(K, K^{\mathtt{mac}}) := h_1(e)$ regardless of whether $e = \mathsf{Enc}_1(\mathsf{pk}, r'; G(r'))$. Since h_1 and h' are internal QROs and cannot be queried by $\mathcal{A}$, the only way for $\mathcal{A}$ to learn information from h_1 is to query H or DEC on honestly-generated e, and the only way to learn information from h' is to query DEC on invalid e. e is invalid means that $\mathsf{Dec}_1(\mathsf{sk}_1, e) = r$ but $r = \perp$ or $\mathsf{Enc}_1(\mathsf{pk}_1, r; G(r)) \neq e$. However, since in this game the randomness in $\mathsf{Enc}_1(\mathsf{pk}_1, \cdot; \cdot)$ is generated by g' (always "good" randomnesses that will not lead to invalid ciphertexts), the internal queries to h and queries to h' are disjoint, and thus we can merge h_1 and h' as one oracle, and use the same oracle to respond invalid queries to DEC. In $\mathcal{A}$'s view, the responds of DEC and H in $\mathbf{G}_3$ and $\mathbf{G}_4$ still have the same distribution, and thus $\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1]$.

Game $\mathbf{G}_5$: The oracle G is simulated using g instead of g'. Similar to the difference between $\mathbf{G}_1$ and $\mathbf{G}_2$, we have

$$\left|\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1]\right| \leq 8(\mu + q_G + q_H + 1)^2\delta.$$

Note that in this game, DEC does not query G. Now the game simulator does not need sk_1 to simulate DEC.

Games $\mathbf{G}_5$-$\mathbf{G}_8$

01 $(\mathsf{pk}_1, (\mathsf{sk}_1, k)) \leftarrow \mathsf{sKG}, \mathsf{pk} := \mathsf{pk}_1$
02 $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1, \mathsf{pk} := \mathsf{lpk}_1$ // $\mathbf{G}_6$-$\mathbf{G}_8$
03 $\mathcal{M}_a \leftarrow \mathcal{A}^{\text{DEC}, |G \times H\rangle}(\mathsf{pk})$
04 **for** $i \in [\mu]$
05 $\quad \mathbf{m}[i] := m_i \leftarrow \mathcal{M}_a$
06 $\quad r_i \xleftarrow{\$} \mathcal{M}'$
07 $\quad R_i := G(r_i)$ // $\mathbf{G}_5$-$\mathbf{G}_7$
08 $\quad R_i \xleftarrow{\$} \mathcal{R}'$ // $\mathbf{G}_8$
09 $\quad e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i; R_i)$
10 $\quad (K_i, K_i^{\mathtt{mac}}) = H(r_i, e_i)$ // $\mathbf{G}_5$-$\mathbf{G}_7$
11 $\quad d_i := K_i \oplus m_i$ // $\mathbf{G}_5$-$\mathbf{G}_7$
12 $\quad d_i \xleftarrow{\$} \mathcal{M}$ // $\mathbf{G}_8$
13 $\quad K_i^{\mathtt{mac}} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$ // $\mathbf{G}_8$
14 $\quad \tau_i := \mathsf{Tag}(K_i^{\mathtt{mac}}, d_i)$
15 $\quad \mathbf{c}[i] := (e_i, d_i, \tau_i)$
16 $out \leftarrow \mathcal{A}^{\text{OPEN}, \text{DEC}, |G \times H\rangle}(\mathbf{c})$
17 **return** $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

OPEN(i)

18 $I := I \cup \{i\}$
19 **return** (m_i, r_i)

DEC($c = (e, d, \tau)$): for $c \notin \mathbf{c}$

20 $(K, K^{\mathtt{mac}}) := h_1(e)$
21 **if** $\exists i \in [\mu] \setminus I$ s.t. $e = e_i$ // $\mathbf{G}_7$-$\mathbf{G}_8$
22 $\quad$ **return** $\perp$ // $\mathbf{G}_7$-$\mathbf{G}_8$
23 **if** $\exists i \in I$ s.t. $e = e_i$ // $\mathbf{G}_8$
24 $\quad (K, K^{\mathtt{mac}}) := (K_i, K_i^{\mathtt{mac}})$ // $\mathbf{G}_8$
25 **if** $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
26 $\quad m := K \oplus d$
27 **else** $m := \perp$
28 **return** m

$H(r, e)$

29 **if** $\exists i \in I$ s.t. $(r, e) = (r_i, e_i)$ // $\mathbf{G}_8$
30 $\quad$ **return** $(d_i \oplus m_i, K_i^{\mathtt{mac}})$ // $\mathbf{G}_8$
31 **if** $e = \mathsf{Enc}_1(\mathsf{pk}, r; G(r))$ // $\mathbf{G}_3$-$\mathbf{G}_8$
32 $\quad$ **return** $h_1(e)$ // $\mathbf{G}_3$-$\mathbf{G}_8$
33 **return** $h(r, e)$

$G(r)$

34 **if** $\exists i \in I$ s.t. $r = r_i$ // $\mathbf{G}_8$
35 $\quad$ **return** R_i // $\mathbf{G}_8$
36 **return** $g(r)$

Figure 16: Games $\mathbf{G}_5$-$\mathbf{G}_8$ for the proof of Theorem 5.3.

Game $\mathbf{G}_6$: The public key is switched to lossy mode by $(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1$ (see Item 02). Note that this game can be simulated without using sk_1. By the key indistinguishability of PKE_1, we have

$$\left|\Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1]\right| \leq \mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{PKE}_1}(\mathcal{A})$$

Game $\mathbf{G}_7$: The decryption oracle always returns $\perp$ if the adversary queries a ciphertext (e, d, τ) that e is the PKE_1 part of some unopened challenge ciphertexts, i.e., $e = e_i$ for an $i \in [\mu] \setminus I$. This game is necessary for simulating the decryption oracle without secret key when constructing a reduction from lossiness of PKE_1.

Let Bad be the event that $\mathcal{A}$ queries DEC on a ciphertext (e, d, τ) that $e = e_i$ for an $i \in [\mu]$ and $\mathsf{Vrfy}(K_i^{\mathtt{mac}}, d, \tau) = 1$. That is, $\mathcal{A}$ forges valid MAC codes of some unopened ciphertext. Let $\mathsf{Bad}_j (:= \mathsf{Bad} : \mathbf{G}_j^{\mathcal{A}})$ be the event that Bad happens in $\mathbf{G}_j^{\mathcal{A}} (j \geq 7)$. If Bad_7 does not occur, then the winning probabilities of $\mathcal{A}$ in $\mathbf{G}_6$ and in $\mathbf{G}_7$ are the same. This is because if $e = e_i$ for some i but $\mathsf{Vrfy}(K_i^{\mathtt{mac}}, d, \tau) = 0$, then

by Item 26, the DEC oracle will still rejects the ciphertexts. Thus

$$\left|\Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathsf{Bad}_7].$$

Now we cannot bound $\Pr[\mathsf{Bad}_7]$ by constructing an MAC adversary, since in $\mathbf{G}_7$, unopened MAC keys $K_j^{\mathtt{mac}}$ are related to H. We introduce the next game $\mathbf{G}_8$ to bound it.

Game $\mathbf{G}_8$: the game simulator generates challenge ciphertexts independent of G and H (see Items 08, 12 and 13). To keep $\mathcal{A}$'s view consistent, when $\mathcal{A}$ issues OPEN queries, we reprogram G and H (cf. Items 34 to 35 and Items 29 to 30).

We use our framework (Lemma 3.1) to bound $\left|\Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_8^{\mathcal{A}} \Rightarrow 1]\right|$. Firstly, we view $\mathcal{A}$ as $(\mathcal{A}_0, \mathcal{A}_1) = (\mathcal{A}_0, (\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\mathrm{OP}}}))$, where n_{OP} is the number of OPEN queries. Essentially, $\mathcal{A}_1$ is divided into $(n_{\mathrm{OP}}+1)$ stages wrt OPEN queries:

- Before any OPEN query (i.e., at the 0-th stage), $\mathcal{A}_{1,0}$ takes $\mathsf{in}_0 := (st, \mathbf{c})$ as input and outputs the first opening index $\mathsf{out}_0 := (i_1)$.
- For $j \in \{1, ..., n_{\mathrm{OP}}-1\}$, $\mathcal{A}_{1,j}$ receives $\mathsf{in}_j = (m_{i_j}, r_{i_j})$ and ends the stage by outputting the $(j+1)$-th opening index $\mathsf{out}_j := i_{j+1}$.
- Finally, $\mathcal{A}_{1,n_{\mathrm{OP}}}$ receives $\mathsf{in}_{n_{\mathrm{OP}}} = (m_{i_{n_{\mathrm{OP}}}}, r_{i_{n_{\mathrm{OP}}}})$ and terminates by outputting $\mathsf{out}_{n_{\mathrm{OP}}} := out$ (i.e, the final output of $\mathcal{A}_1$ in Item 16).

For simplicity, we do not consider the $\mathcal{A}_0$ part, since the output distribution of $\mathcal{A}_0$ in $\mathbf{G}_7$ is the same as in $\mathbf{G}_8$. We only assume that $\mathcal{A}_1$ takes $\mathcal{A}_0$'s final state as its initial state.

In Figure 17, we define $\mathtt{INIT}, \mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ such that $\mathbf{G}_7$ is a NONADA game and $\mathbf{G}_8$ is a ADA. In $\mathbf{G}_8$, when $\mathcal{A}$ queries OPEN(i), the game simulator adds the index i into I. By the codes in Items 34 to 35 and Items 29 to 30, modifying I actually reprograms G and H. So, the OPEN oracle can be viewed as a combination of $\mathsf{F}_{\mathbf{s}}$ and $\mathsf{Repro}_{\mathbf{s}}$ in our framework in Figure 4. Therefore, $\mathbf{G}_7$ and $\mathbf{G}_8$ can be viewed as concrete cases of NONADA and ADA, respectively.

For $k \in \{0, ..., n_{\mathrm{OP}}\}$, let G_k, H_k be the QROs that interacts with $\mathcal{A}_{1,k}$ in $\mathbf{G}_8$, and let G', H' be the QROs that interacts with $\mathcal{A}_1$ in $\mathbf{G}_7$ (the QROs in $\mathbf{G}_7$ do not change). Let I_k be the list $(i_1, ..., i_k)$ which is the opening index list I when the game is interacting with $\mathcal{A}_{1,k}$. By the definition of $\mathbf{G}_7$ and $\mathbf{G}_8$ in Figure 16, we always have $G(r_i) = R_i$ (resp., $H(e_i, r_i) = (K_i, K_i^{\mathtt{mac}})$) for all $i \in [\mu]$ in $\mathbf{G}_7$. But in $\mathbf{G}_8$, we have $G(r_i) = R_i$ (resp., $H(e_i, r_i) = (K_i, K_i^{\mathtt{mac}})$) only if $i \in I_k$. That is, in $\mathbf{G}_8$, $G(r_i) \neq R_i$ and $H(e_i, r_i) \neq (K_i, K_i^{\mathtt{mac}})$ before $\mathcal{A}$ queries OPEN(i). Moreover, for all $r \notin \{r_i\}_{i\in[\mu]}$ and $e \notin \{e_i\}_{i\in[\mu]}$, $G'(r)$ has the same distribution with $G_k(r)$ and $H'(e, r)$ has the same distribution with $H_k(e_i, r_i)$ for all $k \in \{0, ..., n_{\mathrm{OP}}\}$. Therefore, answers of $G_k \times H_k$ differs with answers of $G' \times H'$ in the following set

$$\begin{aligned} S_k &:= \{r \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } r = r_i\} \times \{(r', e') \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } (r', e') = (r_i, e_i)\} \\ &= \{(r, (r', e')) \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } r = r_i \text{ or } (r', e') = (r_i, e_i)\}. \end{aligned} \tag{18}$$

Similar to the argument in the proof of Theorem 4.1, by using Lemma 3.1, there exist adversaries $\{\mathcal{B}_k\}_{k\in\{0,...,n_{\mathrm{OP}}\}}$ such that the probability difference between the event that $\mathbf{G}_6$ output 1 (i.e., $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out) = 1$ in $\mathbf{G}_6$) and similar event in $\mathbf{G}_7$ is bounded by

$$\begin{aligned} &\left|\Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7] - \Pr[\mathbf{G}_8^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_8]\right| \\ &\leq \sum_{k=0}^{n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[(r, (r', e')) \leftarrow \mathcal{B}_i^{|G\times H\rangle} \text{ s.t. } (r, (r', e')) \in S_i \ : \ \mathbf{G}_8^{\mathcal{B}_i}\right]} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}, \end{aligned}$$

where $\mathcal{B}_k$ plays $\mathbf{G}_8$ (and also simulates $\mathbf{G}_8$ for $\mathcal{A}$), randomly measures $\mathcal{A}_{1,k}$'s QRO queries, and outputs the measurement outcome. A detailed description of $\mathcal{B}_k$ will be given in Figure 22. Similarly, since Bad (defined in $\mathbf{G}_7$) is classical event, by Lemma 3.1 again, we also have

$$\begin{aligned} &\left|\Pr[\mathsf{Bad}_7] - \Pr[\mathsf{Bad}_8]\right| = \left|\Pr\left[\mathsf{Bad} : \mathbf{G}_7^{\mathcal{A}}\right] - \Pr\left[\mathsf{Bad} : \mathbf{G}_8^{\mathcal{A}}\right]\right| \\ &\leq \sum_{k=0}^{n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[(r, (r', e')) \leftarrow \mathcal{B}_i^{|G\times H\rangle} \text{ s.t. } (r, (r', e')) \in S_i \ : \ \mathbf{G}_8^{\mathcal{B}_i}\right]} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}. \end{aligned}$$

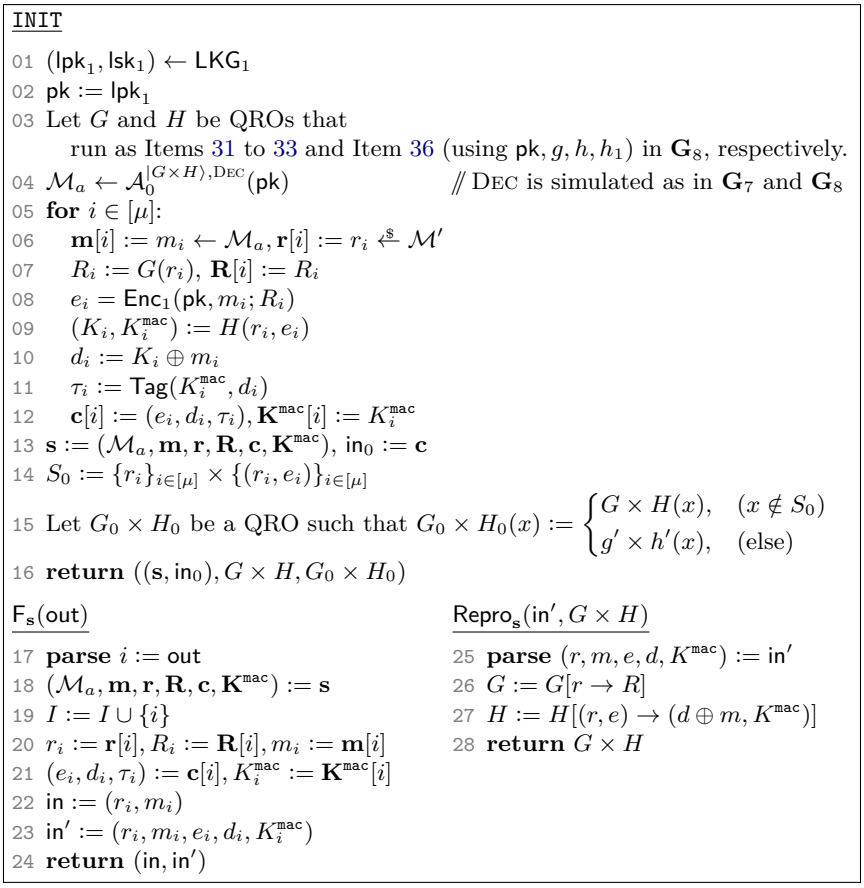

Figure 17: Construction of INIT, $\mathtt{INIT}, \mathsf{F}_\mathbf{s}$, and $\mathsf{Repro}_\mathbf{s}$ used in the proof of Theorem 5.3. g, g', h, h_1, h' are internal quantum-accessible random oracles. Here the adversary also has classical access to Dec. Since Dec will not make $G \times H$ reprogrammed, allowing the adversary to query Dec does not change the bound of Lemma 3.1.

These bounds include a term $\frac{2\mu q}{\sqrt{|\mathcal{R}'|}}$, since $\mathcal{A}_0$ also has quantum access to $|G \times H\rangle$, and this term is the probability that the first stage (i.e., $\mathcal{A}_{1,0}$) of $\mathcal{A}_1$ learns r_i before opening challenge ciphertexts. Such probability is only information-theoretic.

In $\mathbf{G}_8$, $K_i^{\mathtt{mac}}$ are independent of challenge ciphertexts $\mathbf{c}$ (before $\mathcal{A}$ queries $\textsc{Open}(i)$), so we can upper bound $\Pr[\mathsf{Bad}_8]$ by the otSUF-CMA security of $\mathsf{otSUF\text{-}CMA}$, as stated in Lemma D.1.

Lemma D.1 *With the notations and assumptions from the proof of Theorem 5.3, there exists an adversary $\mathcal{F}$ (cf. Figure 21) such that*

$$\Pr\left[\mathsf{Bad} : \mathbf{G}_8^{\mathcal{A}}\right] \leq \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})$$

We also bound the winning probability of $\mathcal{B}_k$ in Lemma D.2. This probability captures the "probability" that $\mathcal{A}$ learns r_i where $\mathbf{c}[i]$ is not opened. Intuitively, since the public key in $\mathbf{G}_8$ is lossy, by the lossiness of PKE_1, ciphertexts that encrypted by lossy key statistically hide the information of their plaintexts. The concrete bound is given in Lemma D.2.

Lemma D.2 *With the notations and assumptions from the proof of Theorem 5.3, for any $k \in \{0, ..., n_{\mathrm{OP}}\}$, we have*

$$\Pr\left[(r, (r', e')) \leftarrow \mathcal{B}_k^{|G\times H\rangle} \ s.t.\ (r, (r', e')) \in S_k \ : \ \mathbf{G}_8^{\mathcal{B}_k}\right] \leq \epsilon_{\mathsf{PKE}_1}^{\mathsf{m\text{-}ind\text{-}enc}} + \frac{\mu q}{|\mathcal{M}'|}.$$

For readability, we postpone the proofs of Lemma D.1 and Lemma D.2 to Supp. Mat. D.2 and continue the proof of Theorem 5.3. With Lemmata D.1 and D.2, we have

$$\left|\Pr\left[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_8^{\mathcal{A}} \Rightarrow 1\right]\right|$$

$$\leq 4(n_{\mathrm{OP}}+1)^2 q\sqrt{\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1}+\frac{\mu q}{|\mathcal{M}'|}}+\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})+\frac{4\mu q}{\sqrt{|\mathcal{R}'|}}$$

Now we can construct a simulator $\mathcal{S}$ that simulates $\mathbf{G}_8$ for $\mathcal{A}$ and interacts with the $\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}$ game. Its simulation process is given in Figure 18. If $\mathcal{A}$ outputs *out*, then $\mathcal{S}$ also outputs *out* except that Bad_8 happens. We have

$$\Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}}\Rightarrow 1]\leq \Pr[\mathbf{G}_8^{\mathcal{A}}\Rightarrow 1]+\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})$$

$\mathcal{S}^{\textsc{Open}'}$

01 Chooses QROs g, h, h_1 at random
02 $I=\emptyset$
03 $k \xleftarrow{\$} \mathcal{M}', (\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1$
04 $\mathsf{pk} := \mathsf{lpk}_1$
05 $\mathcal{M}_a \leftarrow \mathcal{A}^{\textsc{Dec},|G\times H\rangle}(\mathsf{pk})$
06 Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$
07 **for** $i \in [\mu]$
08 $\quad r_i \xleftarrow{\$} \mathcal{M}', R_i \xleftarrow{\$} \mathcal{R}'$
09 $\quad e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i; R_i)$
10 $\quad d_i \xleftarrow{\$} \mathcal{M}, K_i^{\mathtt{mac}} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$
11 $\quad \tau_i := \mathsf{Tag}(K_i^{\mathtt{mac}}, d_i)$
12 $\quad \mathbf{c}[i] := (e_i, d_i, \tau_i)$
13 $out \leftarrow \mathcal{A}^{\textsc{Open},\textsc{Dec},|G\times H\rangle}(\mathbf{c})$
14 **return** out

$G(r)$

15 **if** $\exists i \in I$ s.t. $r = r_i$
16 $\quad$ **return** R_i
17 **return** $g(r)$

$\textsc{Dec}(c=(e,d,\tau))$: for $c \notin \mathbf{c}$

18 $(K, K^{\mathtt{mac}}) := h_1(e)$
19 **if** $\exists i \in [\mu] \setminus I$ s.t. $e = e_i$
20 $\quad$ **return** $\perp$
21 **if** $\exists i \in I$ s.t. $e = e_i$
22 $\quad (K, K^{\mathtt{mac}}) := (K_i, K_i^{\mathtt{mac}})$
23 **if** $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
24 $\quad m := K \oplus d$
25 **else** $m := \perp$
26 **return** m

$\textsc{Open}(i)$

27 $I := I \cup \{i\}$
28 Queries $\textsc{Open}'(i)$ and gets m_i
29 **return** (m_i, r_i)

$H(r,e)$

30 **if** $\exists i \in I$ s.t. $(r,e) = (r_i, e_i)$
31 $\quad$ **return** $(d_i \oplus m_i, K_i^{\mathtt{mac}})$
32 **if** $e = \mathsf{Enc}_1(\mathsf{pk}, r; G(r))$
33 $\quad$ **return** $h_1(e)$
34 **return** $h(r,e)$

Figure 18: Simulator $\mathcal{S}$ in the proof of Theorem 4.2. $\mathcal{S}$ interacts with $\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}$ and has access to $\textsc{Open}'$.

By combining all the probability bounds, we have

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{SO\text{-}CCA}}_{\mathsf{sPKE}}(\mathcal{A},\mathcal{S},\mu,\mathsf{Rel})\\
&=\left|\Pr[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}}\Rightarrow 1]-\Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}}\Rightarrow 1]\right|\\
&\leq \mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{PKE}_1}(\mathcal{A})+3\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})\\
&\quad+6(n_{\mathrm{OP}}+1)^2 q\sqrt{\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1}+\frac{\mu q}{|\mathcal{M}'|}}+16(\mu+n_{\mathrm{DEC}}+q+1)^2\delta\\
&\quad+\frac{(2+\mu)q}{\sqrt{|\mathcal{M}'|}}+\frac{6\mu q}{\sqrt{|\mathcal{R}'|}}+\frac{\mu^2}{|\mathcal{M}'|}+\frac{\mu^2}{\mathcal{R}'}+\frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|}+\frac{\mu n_{\mathrm{DEC}}}{|\mathcal{C}'-n_{\mathrm{DEC}}|}+\frac{\mu^2}{|\mathcal{M}|}
\end{aligned}$$

□

D.1 Bounding $\mathbf{G}_1$ and $\mathbf{G}_2$ in Theorem 5.3

In the proof of Theorem 5.3, we defined a set

$$\begin{aligned}
\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1,\mathsf{sk}_1,r) &:= \{r'\in\mathcal{R}' \mid \mathsf{Dec}_1(\mathsf{sk}_1,\mathsf{Enc}_1(\mathsf{pk}_1,r;r'))=r\}\\
\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1,\mathsf{sk}_1,r) &:= \mathcal{R}'\setminus\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1,\mathsf{sk}_1,r)
\end{aligned}$$

which contains all "good" randomness with respect to the key pair $(\mathsf{pk}_1, \mathsf{sk}_1)$, namely, if a randomness $r' \in \mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1, \mathsf{sk}_1, r)$ is sampled in encrypting a message r, then the resulting ciphertext will be decrypted

correctly. Based on this, we define a set that contains "bad" randomness:

$$\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1, \mathsf{sk}_1, r) := \mathcal{R}' \backslash \mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1, \mathsf{sk}_1, r).$$

Based on these two sets, we further define

$$\delta(\mathsf{pk}_1, \mathsf{sk}_1) = \max_{r \in \mathcal{M}'} \{|\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1, \mathsf{sk}_1, r)|/|\mathcal{R}'|\} \text{ and } \delta = \mathop{\mathbb{E}}_{\mathsf{pk}_1, \mathsf{sk}_1} [\delta(\mathsf{pk}_1, \mathsf{sk}_1)],$$

where the former captures the maximal probability of decryption error with respect to a fixed key pair $(\mathsf{pk}_1, \mathsf{sk}_1)$, and the expectation of the latter is taken over $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}_1$. By Definition 2.1, δ is the error term in the correctness definition of PKE_1.

We use the following lemma from [HKSU20, Theorem 2] to bound the probability difference between $\mathbf{G}_1$ and $\mathbf{G}_2$ in Theorem 5.3.

Lemma D.3 (GDPB [HKSU20]). *Let $\mathcal{X}$ be a finite set, and let $\lambda \in [0, 1]$. Then, for any unbounded and quantum algorithm $\mathcal{A}$ issuing at most q quantum queries,*

$$\left|\Pr[\mathsf{GDPB}^{\mathcal{A}}_{\lambda,0} \Rightarrow 1] - \Pr[\mathsf{GDPB}^{\mathcal{A}}_{\lambda,1} \Rightarrow 1]\right| \leq 8(q+1)^2\lambda,$$

where games $\mathsf{GDPB}^{\mathcal{A}}_{\lambda,b}$ are defined in Figure 19.

Game $\mathsf{GDPB}^{\mathcal{A}}_{\lambda,b}$

$(\lambda_x)_{x\in\mathcal{X}} \leftarrow \mathcal{A}$
if $\exists x \in \mathcal{X}$ s.t. $\lambda_x > \lambda$: **return** 0
if $b = 0$
  Define $\mathsf{F} := 0$
else for $x \in \mathcal{X}$
  $\mathsf{F}(x) \leftarrow B_{\lambda_x}$
$b' \leftarrow \mathcal{A}^{\mathsf{F}}$
return b'

Figure 19: Game $\mathsf{GDPB}^{\mathcal{A}}_{\lambda,b}$ used in Lemma D.3.

The following proof is similar to the one in [PWZ23, Theorem 4.4]. We construct an unbounded adversary $\mathcal{B}$ in Figure 20 that plays $\mathsf{GDPB}_{\delta(\mathsf{pk}_1,\mathsf{sk}_1),b}$ where $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}_1$. Samp is a sampling process and f is a random function used to generate randomness Samp. $\mathcal{B}$ can construct such Samp and f since it is unbounded.

If $\mathcal{B}$ is playing $\mathsf{GDPB}_{\delta(\mathsf{pk}_1,\mathsf{sk}_1),0}$, then $\mathsf{F}(r)$ always outputs 0, and thus $G(r)$ always outputs "good" randomness. This corresponds to $\mathbf{G}_2$; Otherwise, $\mathsf{F}(r)$ outputs 1 with probability $|\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1, \mathsf{sk}_1, r)|/|\mathcal{R}'|$ and thus $G(r)$ outputs "bad" randomness with such probability. This means that the $G(r)$ outputs are uniformly distributed over $\mathcal{R}'$ and thus it behaves as in $\mathbf{G}_1$. Considering the expectation over $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KG}_1$, we have

$$\begin{aligned}
&\left|\Pr[\mathbf{G}^{\mathcal{A}}_1 \Rightarrow 1] - \Pr[\mathbf{G}^{\mathcal{A}}_2 \Rightarrow 1]\right| \\
&= \mathop{\mathbb{E}}_{(\mathsf{pk}_1,\mathsf{sk}_1)\leftarrow\mathsf{KG}_1} \left[\left|\Pr[\mathsf{GDPB}^{\mathcal{B}}_{\delta(\mathsf{pk}_1,\mathsf{sk}_1),1} \Rightarrow 1] - \Pr[\mathsf{GDPB}^{\mathcal{B}}_{\delta(\mathsf{pk}_1,\mathsf{sk}_1),0} \Rightarrow 1]\right|\right] \\
&= 8(q+1)^2 \mathop{\mathbb{E}}_{(\mathsf{pk}_1,\mathsf{sk}_1)\leftarrow\mathsf{KG}_1} [\delta(\mathsf{pk}_1, \mathsf{sk}_1)] \\
&= 8(\mu + n_{\mathrm{DEC}} + q_G + q_H + 1)^2\delta,
\end{aligned}$$

since $\mathcal{B}$ issues at most $\mu + n_{\mathrm{DEC}} + q_G + q_H$ quantum queries to F.

D.2 Proofs of Lemmata D.1 and D.2

Lemma D.1. In Figure 21, we construct a forger $\mathcal{F}$ that simulates $\mathbf{G}_8$ for $\mathcal{A}$ and forges a valid message-tag of otSUF-CMA. By Definition 2.4, $\mathcal{F}$ has access to oracles Tag (at most one query) and Vrfy. $\mathcal{F}$

$\underline{\mathcal{B}^{|\mathsf{F}\rangle}}$

Picks a random function f
$(\mathsf{pk}_1, (\mathsf{sk}_1, k)) \leftarrow \mathsf{sKG}$
for $\forall r \in \mathcal{M}'$
  $\lambda_r := |\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1, \mathsf{sk}_1, r)|/|\mathcal{R}'|$
    // We have $\lambda_r \leq \delta(\mathsf{pk}_1, \mathsf{sk}_1)$
Output $(\lambda_r)_{r\in\mathcal{M}'}$ to the GDPB game.
Simulates $\mathbf{G}_1$ (in Figure 15) for $\mathcal{A}$...
return $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out)$

$\underline{G(r)}$

if $\mathsf{F}(r) = 0$
  return $\mathsf{Samp}(\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_1, \mathsf{sk}_1, r); f(r))$
else
  return $\mathsf{Samp}(\mathcal{R}'_{\mathsf{bad}}(\mathsf{pk}_1, \mathsf{sk}_1, r); f(r))$

Figure 20: Adversary $\mathcal{B}$ in bounding $\mathbf{G}_1$ and $\mathbf{G}_2$ in Theorem 5.3. $\mathcal{B}$ plays $\mathsf{GDPB}_{\delta,b}$, so it has quantum access to an oracle F (defined in Figure 19). $\mathcal{B}$ simulates G using Samp and f. Other oracles are the same as in $\mathbf{G}_1$ of Figure 15.

$\underline{\text{Forger } \mathcal{F}^{\text{Tag},\text{Vrfy}}}$

Chooses QROs g, h, h_1 at random
$I = \emptyset$
$\mathsf{Bad} := \mathbf{false}, i^* \xleftarrow{\$} [\mu]$
$m^* := \bot, \tau^* := \bot$
$(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1, \mathsf{pk} := \mathsf{lpk}_1$
$\mathcal{M}_a \leftarrow \mathcal{A}^{|G\times H\rangle, \text{Dec}}(\mathsf{pk})$
for $i \in [\mu]$
  $\mathbf{m}[i] := m_i \xleftarrow{\$} \mathcal{M}_a$
  $r_i \xleftarrow{\$} \mathcal{M}', R_i \xleftarrow{\$} \mathcal{R}'$
  $e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i; R_i)$
  $d_i \xleftarrow{\$} \mathcal{M}, K_i^{\mathtt{mac}} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$
  if $i = i^*$: $\tau_{i^*} = \text{Tag}(e_{i^*}, d_{i^*})$
  else $\tau_i := \mathsf{Tag}(K_i^{\mathtt{mac}}, e_i, d_i)$
  $\mathbf{c}[i] := (e_i, d_i, \tau_i)$
$out \leftarrow \mathcal{A}^{\text{Open},\text{Dec},|G\times H\rangle}(\mathbf{c})$
return (m^*, τ^*)

$\underline{\text{Dec}(c = (e, d, \tau))\text{: for } c \notin \mathbf{c}}$

$(K, K^{\mathtt{mac}}) := h_1(e)$
if $\exists i \in [\mu] \setminus I$ s.t. $e = e_i$
  if $\text{Vrfy}((e, d), \tau) = 1$
    $\mathsf{Bad} := \mathbf{true}$
    $m^* := (e, d), \tau^* := \tau$
  return $\bot$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \bot$
return m

$\underline{\text{Open}(i)}$

if $i = i^*$: **abort**
$I := I \cup \{i\}$
return (m_i, r_i)

Figure 21: The forger $\mathcal{F}$ in the proof of Lemma D.1. It has access to the oracles $\{\text{Tag}, \text{Vrfy}\}$ provided by game $\mathsf{otSUF\text{-}CMA}_{\mathsf{MAC}}$. Oracles G and H are the same as in $\mathbf{G}_8$ in the proof of Theorem 5.3.

chooses $i^* \xleftarrow{\$} [\mu]$ and generates τ_{i^*} by querying Tag oracle on d_{i^*}. Note that now $K_{i^*}^{\mathtt{mac}}$ is not the actual key of τ_{i^*}. $\mathcal{F}$ aborts the game if $\mathcal{A}$ opens c_{i^*}. If $\mathcal{A}$ triggers event Bad, then $\mathcal{F}$ records the message-tag pair (see Items 19 to 21). When $\mathcal{A}$ terminates, $\mathcal{F}$ outputs the recorded message-tag pair.

The probability that $\mathcal{A}$ does not open c_{i^*} is $(\mu - n_{\text{OP}})/\mu$. If the event Bad_8 occurs, then the probability that the ciphertext (e, d, τ) (that raises this event) satisfies $e = e_{i^*}$ is $1/(\mu - n_{\text{OP}})$. So, we have $\Pr[\mathsf{Bad}_8] = \mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})$

□

Lemma D.2. In $\mathbf{G}_8$ of the proof of Theorem 5.3, the adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ is divided into $(\mathcal{A}_0, (\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\text{OP}}}))$ with respect to Open queries. We ignore $\mathcal{A}_0$, since its queries do not require reprogramming of the QROs. We assume that $\mathcal{A}_1$'s initial state is $\mathcal{A}_0$'s final state. By our framework in Lemma 3.1, $\mathcal{B}_k (k \in \{0, ..., n_{\text{OP}}\})$ is an adversary that runs $\mathcal{A}_{1,0}, ...\mathcal{A}_{1,k}$ and randomly measures $\mathcal{A}_{1,k}$'s QRO query. The construction of $\mathcal{B}_k$ in this proof is shown in Figure 22.

By our framework in Lemma 3.1, $\mathcal{B}_k$ interacts with an Ada game. As shown in the proof of Theorem 5.3, $\mathbf{G}_8$ can be viewed as an Ada game, and the Open oracle is the $\mathsf{F}_{\mathbf{s}}$ function, since queries to Open will make QROs reprogrammed.

Since $\mathcal{B}_k$ finally output the measurement outcome of one of $\mathcal{A}_{1,k}$'s QRO query, we slightly modify $\mathbf{G}_8$ to fit into $\mathcal{B}_k$. Figure 23 shows the modified game $\mathbf{G}'_8$. Game $\mathbf{G}'_8$ is the same as $\mathbf{G}_8$ except that $\mathbf{G}'_8$ runs $\mathcal{B}_k$ and outputs $\mathcal{B}_k$'s output. That is,

$$\Pr\left[(r, (r', e')) \leftarrow \mathcal{B}_k^{|G\times H\rangle} \text{ s.t. } (r, (r', e')) \in S_k \ : \ \mathbf{G}_8^{\mathcal{B}_k}\right]$$

$\underline{\mathcal{B}_k^{\text{OPEN},\text{DEC},|G\times H\rangle}(\mathbf{c}), k \in [0,...,n_{\text{OP}}]}$

$t^* \xleftarrow{\$} [q_k]$
$\mathsf{in}_0 := \mathbf{c}$
for $j = 0$ **to** $k-1$:
  $\mathsf{out}_j := i \leftarrow \mathcal{A}_{1,j}^{|G\times H\rangle,\text{DEC}}(\mathsf{in}_j)$ // $\mathcal{A}_{1,j}$ wants to open cipertext $\mathbf{c}[i]$
  $(m_i, r_i) := \text{OPEN}(\mathsf{out}_j)$ // Queries $\text{OPEN}(i)$ and ...
  $\mathsf{in}_{j+1} := (m_i, r_i)$ // ... gets the message and randomness of $\mathbf{c}_i$
Runs $\mathcal{A}_{1,k}^{|G\times H\rangle,\text{DEC}}(\mathsf{in}_k)$ until it issues t^*-th quantum query to $G \times H$
Let $|\varphi\rangle$ be the t^*-th quantum query to $G \times H$
$(r, (r', e')) \leftarrow \mathsf{Measure}(|\varphi\rangle)$
return $(r, (r', e'))$

Figure 22: The constructions of $\mathcal{B}_k (0 \le k \le n_{\text{OP}})$ in the proof of Lemma D.2. If $\mathcal{A}_{1,j}$ queries $G \times H$ or DEC, $\mathcal{B}_k$ just forwards these queries to its game simulator and and then forwards the response to $\mathcal{A}_{1,j}$. Since $(\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\text{OP}}})$ is obtained by dividing $\mathcal{A}_1$ into $(n_{\text{OP}} + 1)$ stages with respect to $\mathcal{A}_1$'s queries to OPEN, $\mathcal{A}_{1,j}$ terminates with outputting the j-th opening index i_j.

$$= \Pr\left[\mathbf{G}_8'^{\mathcal{B}_k} \Rightarrow (r, (r', e')) \text{ s.t. } (r, (r', e')) \in S_k\right],$$

where S_k is defined in Equation (18). Recall that $(r, (r', e')) \in S_k$ means that $r = r_i$ or $(r', e') = (r_i, e_i)$ where $i \in [\mu] \setminus I_k$ (i.e., $\mathcal{B}_k$ does not open $\mathbf{c}[i]$). To bound the probability that $\mathcal{B}_k$ outputs $(r, (r', e')) \in S_k$, we consider the games $\mathbf{G}_8'$ and $\mathbf{G}_9'$ in Figure 23.

Games $\mathbf{G}_8'$ and $\mathbf{G}_9'$ for $\mathcal{B}_k (k \in [0,...,n_{\text{OP}}])$

$(\mathsf{pk}_1, (\mathsf{sk}_1, k)) \leftarrow \mathsf{sKG}, \mathsf{pk} := \mathsf{pk}_1$
$(\mathsf{lpk}_1, \mathsf{lsk}_1) \leftarrow \mathsf{LKG}_1, \mathsf{pk} := \mathsf{lpk}_1$
$\mathcal{M}_a \leftarrow \mathcal{A}_0^{|G\times H\rangle,\text{DEC}}(\mathsf{pk})$
for $i \in [\mu]$
  $\mathbf{m}[i] := m_i \leftarrow \mathcal{M}_a$
  $r_i \xleftarrow{\$} \mathcal{M}'$
  $R_i \xleftarrow{\$} \mathcal{R}'$ // $\mathbf{G}_8'$
  $e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i; R_i)$ // $\mathbf{G}_8'$
  $r_i' \xleftarrow{\$} \mathcal{M}', R_i' \xleftarrow{\$} \mathcal{R}'$ // $\mathbf{G}_9'$
  $e_i := \mathsf{Enc}_1(\mathsf{pk}, r_i'; R_i')$ // $\mathbf{G}_9'$
  $R_i \leftarrow \mathsf{open}_1(\mathsf{lsk}_1, \mathsf{lpk}_1, e_i, R_i', r_i)$ // $\mathbf{G}_9'$
  $d_i \xleftarrow{\$} \mathcal{M}, K_i^{\mathtt{mac}} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$
  $\tau_i := \mathsf{Tag}(K_i^{\mathtt{mac}}, e_i, d_i)$
  $\mathbf{c}[i] := (e_i, d_i, \tau_i)$
$(r, (r', e')) \leftarrow \mathcal{B}_k^{\text{OPEN},\text{DEC},|G\times H\rangle}(st, \mathbf{c})$
return $(r, (r', e'))$

$\underline{\text{OPEN}(i)}$

$I := I \cup \{i\}$
return (m_i, r_i)

$\underline{\text{DEC}(c = (e, d, \tau)): \text{ for } c \notin \mathbf{c}}$

$(K, K^{\mathtt{mac}}) := h_1(e)$
if $\exists i \in [\mu] \setminus I$ s.t. $e = e_i$
  return $\perp$
if $\exists i \in I$ s.t. $e = e_i$
  $(K, K^{\mathtt{mac}}) := (K_i, K_i^{\mathtt{mac}})$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \perp$
return m

$\underline{H(r, e)}$

if $\exists i \in I$ s.t. $(r, e) = (r_i, e_i)$
  return $(d_i \oplus m_i, K_i^{\mathtt{mac}})$
if $e = \mathsf{Enc}_1(\mathsf{pk}, r; G(r))$
  return $h_1(e)$
return $h(r, e)$

$\underline{G(r)}$

if $\exists i \in I$ s.t. $r = r_i$
  return R_i
return $g(r)$

Figure 23: Games $\mathbf{G}_8'$-$\mathbf{G}_9'$ and constructions of $\mathcal{B}_k (k \in \{0,...,n_{\text{OP}}\})$ for the proof of Lemma D.2.

Game $\mathbf{G}_9'$: We change the generation of challenge ciphertexts. To generate e_i, we independently sample a PKE_1 message r_i' and randomness R_i', encrypt r_i' using R_i', and get e_i'. Then we use the opening algorithm open_1 to claim the ciphertext e_i to the PKE_1 message r_i with randomness R_i. Similar to $\mathbf{G}_8'$, we still use r_i and R_i to reprogram G and H. r_i' and R_i' are just used to generate e_i. By the property of open_1, R_i has the same distribution with R_i'. Since the public key in $\mathbf{G}_8'$ and $\mathbf{G}_9'$ is lossy, by the

multi-challenge lossiness of PKE_1 (Definition 5.2), we have

$$\Big|\Pr\left[\mathbf{G}_8'^{\mathcal{B}_k} \Rightarrow (r,(r',e')) \text{ s.t. } (r,(r',e')) \in S_k\right] - \Pr\left[\mathbf{G}_9'^{\mathcal{B}_k} \Rightarrow (r,(r',e')) \text{ s.t. } (r,(r',e')) \in S_k\right]\Big| \leq \epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1}$$

Moreover, in $\mathbf{G}_9'$, r_i and R_i are uniformly random and independent of $\mathcal{B}_k$'s view before $\mathcal{B}_k$ opens $\mathbf{c}[i]$. By using a union bound over all QRO queries, we get

$$\Pr\left[\mathbf{G}_9'^{\mathcal{B}_k} \Rightarrow (r,(r',e')) \text{ s.t. } (r,(r',e')) \in S_k\right] \leq \frac{q\mu}{|\mathcal{M}'|}.$$

Therefore, we have

$$\Pr\left[(r,(r',e')) \leftarrow \mathcal{B}_k^{|G\times H\rangle} \text{ s.t. } (r,(r',e')) \in S_k \ : \ \mathbf{G}_8^{\mathcal{B}_k}\right] \leq \epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{PKE}_1} + \frac{q\mu}{|\mathcal{M}'|},$$

as stated in Lemma D.2. □

E Lossy Encryption from LWE

We construct a lossy encryption scheme from the (Decisional) Learning With Errors (LWE) assumption. Essentially, our lossy encryption is the same as the Regev scheme [Reg05] except that our encryption algorithm uses short Gaussian errors instead of binary. The same has been done in [GPV08]. The purpose of doing so is to achieve weak openability (cf. Definition 5.1) as required by our tight generic construction in Section 5.

Before giving the construction, we first recall the Learning With Errors (LWE) assumption and some relevant lemmas.

Definition E.1 (LWE Assumption). Let n be a positive integer, $q := q(n)$ be a modulus, χ be a discrete distribution over $\mathbb{Z}_q$. We say that the $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, if for every PPT algorithm $\mathcal{B}$, the following advantage is negligible in n:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{LWE}_{n,m,q,\chi}}(\mathcal{B}) := |\Pr[\mathcal{B}(\mathbf{A},\mathbf{b}) = 1 \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{b} \xleftarrow{\$} \mathbb{Z}_q^m] \\ - \Pr[\mathcal{B}(\mathbf{A},\mathbf{A}^T\mathbf{s}+\mathbf{e}) = 1 \mid \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m]|.\end{aligned}$$

Lemma E.2 (Theorem 5.1 in [MP12]). *There is an efficient randomized algorithm* $(\mathbf{B},\mathbf{R}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$ *that, given any integers* $n \geq 1, q \geq 2$, *and sufficiently large* $m = O(n \log q)$, *outputs a parity-check matrix* $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ *and a trapdoor* $\mathbf{R}$ *such that the distribution of* $\mathbf{B}$ *is* $\mathsf{negl}(n)$*-far from uniform.*

Moreover, for any $\mathbf{y} \in \mathbb{Z}_q^n$ *and large enough* $s = O(\sqrt{n \log q})$, *the randomized algorithm* $\mathsf{SampleD}(\mathbf{R},\mathbf{B},\mathbf{y},s)$ *samples from a distribution within* $\mathsf{negl}(n)$ *statistical distance of* $D_{\Lambda_{\mathbf{y}}^{\perp}(\mathbf{B}),s\cdot\omega(\sqrt{\log n})}$, *where* $\Lambda_{\mathbf{y}}^{\perp}(\mathbf{B})$ *is defined as the set* $\{\mathbf{z} \in \mathbb{Z}^m | \mathbf{B}\mathbf{z} = \mathbf{y}\}$.

For integer q and real number $\alpha \in \mathbb{R}$, we define Ψ_α be the distribution on $\mathbb{R}/\mathbb{Z}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$. Let $\chi : \mathbb{Z}_q \to \mathbb{R}^+$ be the discrete distribution over $\mathbb{Z}_q$ of the random variable $\lfloor q \cdot X_{\Psi_\alpha} \rceil \mod q$, where random variable X_{Ψ_α} has distribution Ψ_α and $\lfloor\cdot\rceil$ rounds a real number to its nearest integer.

Let $b \in \{0,1\}$ and $v \in \mathbb{Z}_q$. We define $\mathsf{Encode}_q(b) := \lfloor \frac{q}{2} \cdot b \rceil$, where $\lfloor\cdot\rceil$ rounds a real number to its nearest integer, and define $\mathsf{Decode}_q(v)$ such that if v is closer to 0 than to $(\lfloor \frac{q}{2} \rfloor \mod q)$, then $\mathsf{Decode}_q(v)$ outputs 0, and otherwise output 1. For a vector $\mathbf{v}$, $\mathsf{Encode}_q(\mathbf{v})$ means that applying Encode_q to $\mathbf{v}$ coordinate-wise, and the same for Decode_q.

Let integer $m \geq 2n \cdot \log q$ and $D_{\mathbb{Z}^m,r}$ be the discrete Gaussian distribution over $\mathbb{Z}^m$ with a parameter r. Our lossy encryption LWEPKE_1 with message space $\mathcal{M}' := \{0,1\}^\ell$ is described in Figure 24. It has randomness space $\mathcal{R}' := D_{\mathbb{Z}^m,r}$ and ciphertext space $\mathcal{C}' := \mathbb{Z}_q^n \times \mathbb{Z}_q^\ell$. In its lossy mode, it requires the $\mathbf{G}$-trapdoor technique [MP12] to achieve weak openability, and we recall the useful lemma as follow.

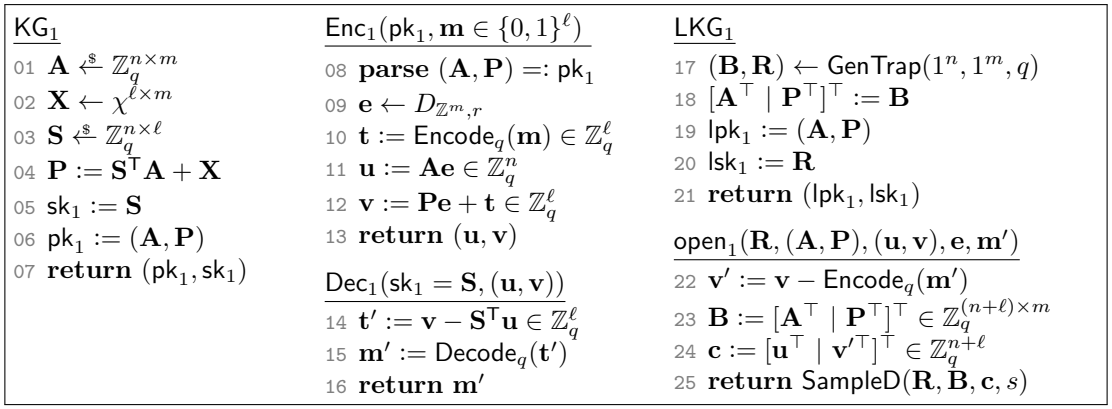

Figure 24: A LWE-based lossy encryption scheme $\mathsf{LWEPKE}_1 = (\mathsf{KG}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$.

Theorem E.3 *Let λ be a security parameter. If we use the following parameter setting $n = \mathsf{poly}(\lambda), \ell = O(n)$, prime $q \in [\frac{n^4}{2}, n^4]$, $m = O(n\log(q))$, $r = O(\sqrt{m}\log(n))$, $\alpha = \frac{1}{O(m\log^2(n))}$, then the PKE scheme LWEPKE_1 in Figure 24 is a lossy encryption. Specifically, LWEPKE_1 is $(1-\mathsf{negl}(n))$-correct, $\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{LWEPKE}_1} = \mathsf{negl}(n)$, and for any adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{LWEPKE}_1}(\mathcal{A}) \le \ell \cdot \mathsf{Adv}^{\mathsf{LWE}_{n,m,q,\chi}}(\mathcal{B}) + \mathsf{negl}(n)$$

Proof. Correctness. This is very similar to [GPV08]. By the parameters in Theorem E.3 and the following lemma, the scheme PKE_1 is $(1-\mathsf{negl}(n))$-correct.

Lemma E.4 *([GPV08, Lemma 8.2]) If $q > 5rm$ and $\alpha \le 1/(r\sqrt{m}\cdot\omega(\sqrt{\log n}))$, then Dec_1 in Figure 24 decrypts correctly with overwhelming probability.*

Key Indistinguishability. We use the $\mathsf{LWE}_{n,m,q,\chi}$ assumption ℓ times to show that $\mathbf{P}$ generated by KG_1 is indistinguishable from a random matrix in $\mathbb{Z}_q^{\ell\times m}$. Hence, our real key pk_1 is pseudorandom. Moreover, by Lemma E.2, our lossy key lpk_1 generated using $\mathsf{GenTrap}$ is $\mathsf{negl}(n)$-far from uniform. Hence, we have

$$\mathsf{Adv}^{\mathsf{ind\text{-}key}}_{\mathsf{LWEPKE}_1}(\mathcal{A}) \le \ell \cdot \mathsf{Adv}^{\mathsf{LWE}_{n,m,q,\chi}}(\mathcal{B}) + \mathsf{negl}(n),$$

Lossiness. Let $(\mathbf{A}, \mathbf{P})$ be generated by LKG_1. Again, by Lemma E.2, $\mathbf{B} = \begin{pmatrix}\mathbf{A}\\ \mathbf{P}\end{pmatrix} \in \mathbb{Z}_q^{(n+\ell)\times m}$ is $\mathsf{negl}(n)$-close to uniform. If $\mathbf{B}$ is uniform, by the leftover hash lemma in [AP08, Section 2.2.1], the distribution of $\mathbf{B}\mathbf{e}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n+\ell}$). Hence, $\epsilon^{\mathsf{m\text{-}ind\text{-}enc}}_{\mathsf{LWEPKE}_1} = \mathsf{negl}(n)$.

(Weak) Openability. The openability of our scheme LWEPKE_1 is stronger than the weak openability required as in Definition 5.1, namely, our opening algorithm does not need to use the original encryption randomness $\mathbf{e}$. By the lossiness, any ciphertext is a valid ciphertext of $\mathbf{m}'$. According to open_1 in Figure 24, $\mathbf{e}' \xleftarrow{\$} \mathsf{SampleD}(\mathbf{R}, \mathbf{B}, \mathbf{c}, s)$ will satisfy $\mathbf{B}\mathbf{e}' + \begin{pmatrix}\mathbf{0}\\ \mathsf{Encode}_q(\mathbf{m}')\end{pmatrix} = \begin{pmatrix}\mathbf{u}\\ \mathbf{v}\end{pmatrix}$ and $\mathbf{e}' \in D_{\mathbb{Z}^m, r}$. □

F Proof of Theorem 4.2

The proof of Theorem 4.2 is similar to Theorem 5.3. The main difference is that in Theorem 4.2 we need to construct a reduction from OW-CPA security. This reduction is also similar to the OW-CPA reduction in the proof of Theorem 4.1.

Theorem 4.2. Let $h : \mathcal{M}' \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}^{\mathsf{mac}}$ and $g : \mathcal{M}' \to \mathcal{R}'$ be internal quantum-accessible ROs which are used to respond the queries to H and G, respectively. Similar to the proof of Theorem 4.1, to fit into the syntax of our framework, we combine G and H as one random oracle $G \times H$ such that $G \times H(r', r, e) := (G(r'), H(r, e))$. $\mathcal{A}$ can query $G \times H$ at most $q = q_H + q_G$ times.

During the proof, we implicitly assume that $\mathcal{A}_0$ will not query $\textsc{Dec}$ on (e,d,τ) with $(e,d)=(e_i,d_i)$ before seeing the challenge ciphertexts $\mathbf{c}$. Since r_i's are independent of $\mathcal{A}$'s view before it sees $\mathbf{c}$, the probability that $\mathcal{A}$ queries $\textsc{Dec}$ on such ciphertexts is $\frac{\mu n_{\textsc{Dec}}}{|\mathcal{C}'-n_{\textsc{Dec}}|}+\frac{\mu q}{\sqrt{|\mathcal{M}'|}}$, where the second term is the bound to search $G(r_i)$ and $H(r_i)$ given quantum access to $G\times H$. Moreover, we also assume that there is no collision among outputs of r_i's, R_i's, K_i's, and $K_i^{\mathtt{mac}}$'s. This introduce collision bounds $\frac{\mu^2}{|\mathcal{M}'|}+\frac{\mu^2}{\mathcal{R}'}+\frac{\mu^2}{|\mathcal{M}|}+\frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|}$. For simplicity, we just add these probability into our final bound, and do not consider it in the game sequences.

Game $\mathbf{G}_0$-$\mathbf{G}_7$

$(\mathsf{pk},(\mathsf{sk},k))\leftarrow\mathsf{sKG}$
$\mathcal{M}_a\leftarrow\mathcal{A}^{\textsc{Dec},|G\times H\rangle}(\mathsf{pk})$
for $i\in[\mu]$
  $\mathbf{m}[i]:=m_i\leftarrow\mathcal{M}_a$
  $r_i\xleftarrow{\$}\mathcal{M}'$
  $R_i:=G(r_i)$ // $\mathbf{G}_0$-$\mathbf{G}_6$
  $R_i\xleftarrow{\$}\mathcal{R}'$ // $\mathbf{G}_7$
  $e_i:=\mathsf{Enc}_0(\mathsf{pk},r_i;R_i)$
  $(K_i,K_i^{\mathtt{mac}})=H(r_i,e_i)$ // $\mathbf{G}_0$-$\mathbf{G}_6$
  $d_i:=K_i\oplus m_i$ // $\mathbf{G}_0$-$\mathbf{G}_6$
  $d_i\xleftarrow{\$}\mathcal{M}$ // $\mathbf{G}_7$
  $K_i^{\mathtt{mac}}\xleftarrow{\$}\mathcal{K}^{\mathtt{mac}}$ // $\mathbf{G}_7$
  $\tau_i:=\mathsf{Tag}(K_i^{\mathtt{mac}},d_i)$
  $\mathbf{c}[i]:=(e_i,d_i,\tau_i)$
$out\leftarrow\mathcal{A}^{\textsc{Open},\textsc{Dec},|G\times H\rangle}(\mathbf{c})$
return $\mathsf{Rel}(\mathcal{M}_a,\mathbf{m},I,out)$

Oracle $H(r,e)$

if $\exists i\in I$ s.t. $(r,e)=(r_i,e_i)$ // $\mathbf{G}_7$
  return $(d_i\oplus m_i,K_i^{\mathtt{mac}})$ // $\mathbf{G}_7$
if $e=\mathsf{Enc}_0(\mathsf{pk},r;G(r))$ // $\mathbf{G}_3$-$\mathbf{G}_6$
  return $h_1(e)$ // $\mathbf{G}_3$-$\mathbf{G}_6$
return $h(r,e)$

$\textsc{Dec}(c=(e,d,\tau))$: for $c\notin\mathbf{c}$

$r':=\mathsf{Dec}_0(\mathsf{sk},e)$ // $\mathbf{G}_0$-$\mathbf{G}_3$
if $r'=\bot$
  or $e\neq\mathsf{Enc}_0(\mathsf{pk},r';G(r'))$ // $\mathbf{G}_0$-$\mathbf{G}_3$
  $(K,K^{\mathtt{mac}}):=h(k,e)$ // $\mathbf{G}_0$
  $(K,K^{\mathtt{mac}}):=h'(e)$ // $\mathbf{G}_1$-$\mathbf{G}_3$
else $(K,K^{\mathtt{mac}}):=h(r',e)$ // $\mathbf{G}_0$-$\mathbf{G}_2$
else $(K,K^{\mathtt{mac}}):=h_1(e)$ // $\mathbf{G}_3$
$(K,K^{\mathtt{mac}}):=h_1(e)$ // $\mathbf{G}_4$-$\mathbf{G}_7$
if $\exists i\in[\mu]\backslash I$ s.t. $e=e_i$ // $\mathbf{G}_6$-$\mathbf{G}_7$
  return $\bot$ // $\mathbf{G}_6$-$\mathbf{G}_7$
if $\exists i\in I$ s.t. $e=e_i$ // $\mathbf{G}_7$
  $(K,K^{\mathtt{mac}}):=(K_i,K_i^{\mathtt{mac}})$ // $\mathbf{G}_7$
if $\mathsf{Vrfy}(K^{\mathtt{mac}},\tau)=1$
  $m:=K\oplus d$
else $m:=\bot$
return m

$\textsc{Open}(i)$

$I:=I\cup\{i\}$
return (m_i,r_i)

Oracle $G(r)$

if $\exists i\in I$ s.t. $r=r_i$ // $\mathbf{G}_7$
  return R_i // $\mathbf{G}_7$
return $g(r)$ // $\mathbf{G}_0$-$\mathbf{G}_1$, $\mathbf{G}_5$-$\mathbf{G}_7$
return $g'(r)$ // $\mathbf{G}_2$-$\mathbf{G}_4$

Figure 25: Games $\mathbf{G}_0$-$\mathbf{G}_7$ for the proof of Theorem 4.2.

Game $\mathbf{G}_0$ is equivalent to $\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}}$, so

$$\Pr[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}}\Rightarrow 1]=\Pr[\mathbf{G}_0^{\mathcal{A}}\Rightarrow 1]$$

Game $\mathbf{G}_1$: The $\textsc{Dec}$ oracle computes $(K,K^{\mathtt{mac}})=h'(e)$ rather than $h(k,e)$ if $r'=\bot$ or $e\neq\mathsf{Enc}_0(\mathsf{pk},r';g(r'))$. By Lemma 2.7, we have

$$\left|\Pr[\mathbf{G}_0^{\mathcal{A}}\Rightarrow 1]-\Pr[\mathbf{G}_1^{\mathcal{A}}\Rightarrow 1]\right|\leq 2q_H/\sqrt{|\mathcal{M}'|}$$

Game $\mathbf{G}_2$: We restrict the range of G to be the "good" randomness space wrt $(\mathsf{pk},\mathsf{sk})$. Specifically, let $\mathcal{R}'_{\mathsf{good}}(\mathsf{pk},\mathsf{sk},r)$ be the set $\{r'\in\mathcal{R}'\mid\mathsf{Dec}_0(\mathsf{sk},\mathsf{Enc}_0(\mathsf{pk},r;r'))=r\}$ and $g':\mathcal{M}'\to\mathcal{R}'$ be a quantum-accessible random oracle such that $g'(r)$ is sampled uniformly from $\mathcal{R}'_{\mathsf{good}}(\mathsf{pk},\mathsf{sk},r)$. If PKE is $(1-\delta)$-correct (see Definition 2.1), then by a similar argument in Supp. Mat. D.1, we have

$$\left|\Pr[\mathbf{G}_1^{\mathcal{A}}\Rightarrow 1]-\Pr[\mathbf{G}_2^{\mathcal{A}}\Rightarrow 1]\right|\leq 8(\mu+n_{\textsc{Dec}}+q_G+q_H+1)^2\delta.$$

Game $\mathbf{G}_3$: We set $H(r,e)=h_1(e)$ if $e=\mathsf{Enc}_0(\mathsf{pk},r;G(r))$, where $h_1:\mathcal{C}\to\mathcal{M}\times\mathcal{K}^{\mathtt{mac}}$ is an internal quantum-accessible random oracle. Since the randomness generated by G (i.e., g') is always a "good" randomness, $\mathsf{Enc}_0(\mathsf{pk},\cdot;G(\cdot))$ is an injective function and thus $h_1(\mathsf{Enc}_0(\mathsf{pk},\cdot;G(\cdot)))$ can be also viewed as an random oracle. Therefore, we have $\Pr[\mathbf{G}_2^{\mathcal{A}}\Rightarrow 1]=\Pr[\mathbf{G}_3^{\mathcal{A}}\Rightarrow 1]$.

Game $\mathbf{G}_4$: We "merge" h_1 and h', namely, DEC always computes $(K, K^{\mathtt{mac}}) := h_1(e)$ regardless of the validity of e. Similar to the argument in the proof of Theorem 5.3, we have $\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1]$.

Game $\mathbf{G}_5$: The oracle G is simulated using g instead of g'. Similar to the difference between $\mathbf{G}_1$ and $\mathbf{G}_2$, we have

$$\left|\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1]\right| \leq 8(\mu + n_{\mathrm{DEC}} + q_G + q_H + 1)^2\delta.$$

Game $\mathbf{G}_6$: The decryption oracle always returns $\perp$ if the adversary queries a ciphertext (e, d, τ) that e is the PKE part of some unopened challenge ciphertexts, i.e., $\exists i \in [\mu]\backslash I, e = e_i$. This game is necessary for simulating the decryption oracle without secret key when constructing OW-CPA reduction.

Let Bad_j be the event that $\mathcal{A}$ queries DEC on a ciphertext (e, d, τ) that $\exists i$ s.t. $e = e_i$ and $\mathsf{Vrfy}(K_i^{\mathtt{mac}}, d, \tau) = 1$ in $\mathbf{G}_j (j \geq 6)$. That is, $\mathcal{A}$ forges valid MAC codes of some unopened ciphertext. If Bad_6 does not occur, then the winning probabilities of $\mathcal{A}$ in $\mathbf{G}_5$ and in $\mathbf{G}_6$ are the same. This is because if $\exists i$ s.t. $e = e_i$ but $\mathsf{Vrfy}(K_i^{\mathtt{mac}}, d, \tau) = 0$, then by lines 19 and 31, the DEC oracle will still rejects the ciphertexts. Thus $\left|\Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathsf{Bad}_6]$.

Note that now we cannot bound $\Pr[\mathsf{Bad}_6]$ by constructing an MAC adversary, since in $\mathbf{G}_6$, unopened MAC keys $K_j^{\mathtt{mac}}$ are related to H. In the next game $\mathbf{G}_7$, the randomness R_i, the MAC keys, and symmetric keys will be generated independent of G and H at first, and if $\mathcal{A}$ opens a challenge ciphertext, we reprogram G and H to make the simulation consistent.

Game $\mathbf{G}_7$: the game simulator generates challenge ciphertexts independent of G, H, and then adaptively reprograms G and H to make the simulation consistent when the adversary issues OPEN queries.

The OPEN queries in $\mathbf{G}_7$ will make the QRO reprogrammed, while the QRO in $\mathbf{G}_6$ always remains the same. So, we can view $\mathbf{G}_6$ and $\mathbf{G}_6$ as concrete cases of NONADA and ADA, respectively. We focus on $\mathcal{A}_1$ since $\mathcal{A}_0$ will not make the QRO reprogrammed. Similar to the proof of Theorem 5.3 in Section 5, we view $\mathcal{A}$ as $(\mathcal{A}_0, \mathcal{A}_1) = (\mathcal{A}_0, (\mathcal{A}_{1,0}, ..., \mathcal{A}_{1,n_{\mathrm{OP}}}))$ and $\mathcal{A}_1$ is divided into $(n_{\mathrm{OP}} + 1)$ stages wrt OPEN queries:

- Before any OPEN query (i.e., at the 0-th stage), $\mathcal{A}_{1,0}$ takes $\mathsf{in}_0 := (st, \mathbf{c})$ as input and outputs the first opening index $\mathsf{out}_0 := (i_1)$.
- For $j \in \{1, ..., n_{\mathrm{OP}} - 1\}$, $\mathcal{A}_{1,j}$ receives $\mathsf{in}_j = (m_{i_j}, r_{i_j})$ and ends the stage by outputting the $(j+1)$-th opening index $\mathsf{out}_j := i_{j+1}$.
- Finally, $\mathcal{A}_{1,n_{\mathrm{OP}}}$ receives $\mathsf{in}_{n_{\mathrm{OP}}} = (m_{i_{n_{\mathrm{OP}}}}, r_{i_{n_{\mathrm{OP}}}})$ and terminates by outputting $\mathsf{out}_{n_{\mathrm{OP}}} := out$.

We do not consider the $\mathcal{A}_0$ part and only assume that $\mathcal{A}_1$ takes $\mathcal{A}_0$'s final state as its initial state.

In Figure 26, we define $\mathtt{INIT}, \mathsf{F_s}$, and $\mathsf{Repro_s}$. If we instantiate NONADA using $\mathtt{INIT}$, $\mathsf{F_s}$, and $\mathsf{Repro_s}$, then the instantiated game is equivalent to the part of $\mathbf{G}_6$ that interacts with $\mathcal{A}_1$. Similarly, it we instantiate ADA using $\mathtt{INIT}$, $\mathsf{F_s}$, and $\mathsf{Repro_s}$, then the instantiated game is equivalent to the part of $\mathbf{G}_7$ that interacts with $\mathcal{A}_1$. Moreover, at $\mathcal{A}$'s k-th stage, our $\mathsf{F_s}$ defines a set

$$S_k := \{(r, (r', e')) \mid \exists i \in [\mu]\backslash I_k \text{ s.t. } r = r_i \text{ or } (r', e') = (r_i, e_i)\} \tag{19}$$

where $I_k := \{i_1, ..., i_k\}$ is the opening index set I in $\mathcal{A}_1$'s k-th stage (i.e., $\mathcal{A}_{1,k}$). Answers of $G \times H$ in $\mathbf{G}_6$ (i.e., NONADA) and $\mathbf{G}_7$ (i.e., ADA) are only different on S_k. For $k = 0$, S_0 is defined at Item 13 and $I_0 = \emptyset$. Similar to the argument in the proof of Theorem 4.1, by using Lemma 3.1, the probability difference between the event that $\mathbf{G}_6$ output 1 (i.e., $\mathsf{Rel}(\mathcal{M}_a, \mathbf{m}, I, out) = 1$ in $\mathbf{G}_6$) and similar event in $\mathbf{G}_7$ is bounded by

$$\begin{aligned}
&\left|\Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_6] - \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7]\right| \\
&\leq \sum_{k=0}^{n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[(r, (r', e')) \leftarrow \mathcal{B}_i^{|G \times H\rangle} \text{ s.t. } (r, (r', e')) \in S_i \ : \ \mathbf{G}_7^{\mathcal{B}_i}\right]} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}},
\end{aligned}$$

where Ev here is the event that $\mathcal{A}_1$ outputs a particular opening index set I and final output out, and $\mathtt{INIT}$, $\mathsf{F_s}$, and $\mathsf{Repro_s}$ are defined in Figure 26. It includes a term $\frac{2\mu q}{\sqrt{|\mathcal{R}'|}}$, since $\mathcal{A}_0$ also has quantum access to $|G \times H\rangle$, and this term is the probability that the first stage (i.e., $\mathcal{A}_{1,0}$) of $\mathcal{A}_1$ learns r_i before

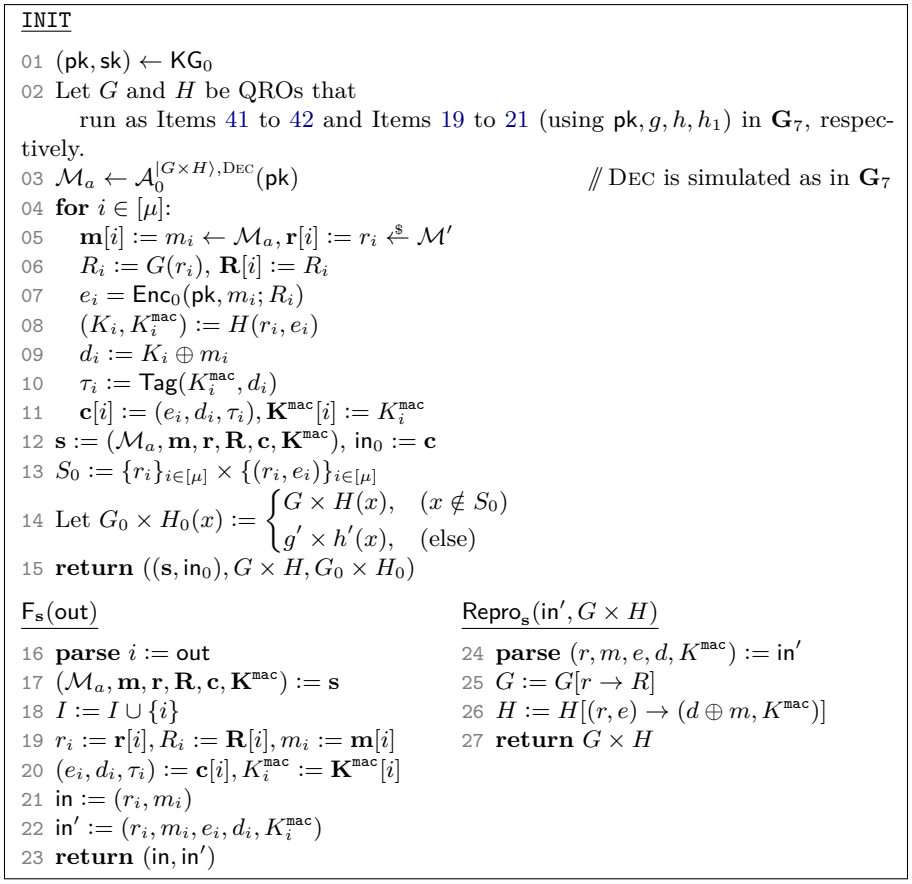

Figure 26: Construction of INIT, $\mathtt{INIT}, \mathsf{F}_\mathbf{s}$, and $\mathsf{Repro}_\mathbf{s}$ used in the proof of Theorem 4.2. g, g', h, h_1, h' are internal quantum-accessible random oracles. Here the adversary also has classical access to DEC. Since DEC will not make $G \times H$ reprogrammed and will not leak information about S_k, allowing the adversary to query DEC does not change the bound of Lemma 3.1.

opening challenge ciphertexts. Moreover, similar to the argument in bounding $\mathbf{G}_7$ and $\mathbf{G}_8$ in the proof of Theorem 5.3, we have

$$\begin{aligned}
&|\Pr[\mathsf{Bad}_6] - \Pr[\mathsf{Bad}_7]| = \left|\Pr\left[\mathsf{Bad} : \mathbf{G}_6^{\mathcal{A}}\right] - \Pr\left[\mathsf{Bad} : \mathbf{G}_7^{\mathcal{A}}\right]\right| \\
&\leq \sum_{k=0}^{n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[(r,(r',e')) \leftarrow \mathcal{B}_i^{|G\times H\rangle} \text{ s.t. } (r,(r',e')) \in S_i \ : \ \mathbf{G}_7^{\mathcal{B}_i}\right]} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}.
\end{aligned}$$

Based on $\mathcal{B}_k (1 \leq k \leq n_{\mathrm{OP}})$, we can construct an OW-CPA adversary $\mathcal{B}_k^{ow}$ (in Figure 27) against PKE that has the same structure with the adversaries in Figure 9, except that $\mathcal{B}_k^{ow}$ needs to simulate the decryption oracle. Since in $\mathbf{G}_7$, the decryption oracle can be simulated without secret key, $\mathcal{B}_k^{ow}$ can simulates $\mathbf{G}_7$ for $\mathcal{B}_k$ perfectly if Bad_7 does not occur. Therefore, similar to the argument in Theorem 4.1, there exists an OW-CPA adversary $\mathcal{B}^{ow}$ such that

$$\begin{aligned}
&\left|\Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_6] - \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7]\right| \\
&\leq 2(n_{\mathrm{OP}}+1)^2 q \sqrt{2\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{\mathrm{ow}}) + \Pr[\mathsf{Bad}_7]} + \frac{2\mu q}{\sqrt{|\mathcal{R}'|}}
\end{aligned} \tag{20}$$

$\mathbf{G}_7$ also enable us to construct an otSUF-CMA reduction to bound $\Pr[\mathsf{Bad}_7]$. Let $\mathcal{F}$ be an otSUF-CMA adversary that simulates $\mathbf{G}_7$ for $\mathcal{A}$. $\mathcal{F}$ here has similar structure with the one in Figure 21. It firstly chooses $t^* \xleftarrow{\$} [\mu]$ uniformly and sets $K_{t^*}^{\mathtt{mac}} := \bot$. To generates τ_{t^*}, $\mathcal{F}$ queries its TAG oracle on d_{t^*} and sets the responding tag as τ_{t^*}. $\mathcal{F}$ aborts the game if $\mathcal{A}$ opens c_{t^*}. When $\mathcal{A}_1$ queries the DEC oracle on

$\underline{\mathcal{B}_k^{\mathrm{ow}}(\mathsf{pk}^*, e^*)}$ // (pk^*, e^*) is a OW-CPA challenge of PKE

01 $((\mathbf{s}, \mathsf{in}_0), (G \times H), (G_0 \times H_0)) \leftarrow \mathtt{INIT}$ // INIT is defined in Figure 26, // where $\mathsf{pk} := \mathsf{pk}^*$

02 **parse** $(\mathcal{M}_a, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}, \mathbf{K}^{\mathtt{mac}}) := \mathbf{s}, \mathbf{c} := \mathsf{in}_0$

03 $t^* \xleftarrow{\$} [\mu], (e_{t^*}, d_{t^*}, \tau_{t^*}) := \mathbf{c}[t^*]$

04 $\mathbf{c}[t^*] := (e^*, d_{t^*}, \tau_{t^*})$, $\mathsf{in}_0 := \mathbf{c}$ // embed the challenge

05 **if** $k = 0$: **goto** line 15

06 $\mathsf{out}_0 \leftarrow \mathcal{B}_k^{|G_0 \times H_0\rangle, \mathrm{DEC}}(\mathsf{in}_0)$ // DEC is simulated as in $\mathbf{G}_8$ in Figure 25

07 **if** $\mathsf{out}_0 = t^*$: **abort**

08 $(\mathsf{in}_1, \mathsf{in}_1') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_0, (G_0 \times H_0))$ // $\mathsf{F}_{\mathbf{s}}$ is defined in Figure 26

09 $(G_1 \times H_1) := \mathsf{Repro}_{\mathbf{s}}(\mathsf{in}_1', (G_0 \times H_0))$ // $\mathsf{Repro}_{\mathbf{s}}$ is defined in Figure 26

10 **for** $j = 1$ **to** $k - 1$:

11 $\quad \mathsf{out}_j \leftarrow \mathcal{B}_k^{|G_j \times H_j\rangle, \mathrm{DEC}}(\mathsf{in}_j)$

12 $\quad$ **if** $\mathsf{out}_j = t^*$: **abort**

13 $\quad (\mathsf{in}_{j+1}, \mathsf{in}_{j+1}') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_j, (G_j \times H_j))$

14 $\quad (G_{j+1} \times H_{j+1}) := \mathsf{Repro}_{\mathbf{s}}(\mathsf{in}_{j+1}', (G_j \times H_j))$

15 $(r_0', (r_1', e')) \leftarrow \mathcal{B}_k^{|G_k \times H_k\rangle, \mathrm{DEC}}(\mathsf{in}_k)$ // perform measurement

16 $b \xleftarrow{\$} \{0, 1\}, r^* := r_b'$ // randomly choose a solution

17 **return** r^*

Figure 27: The constructions of OW-CPA adversaries $\mathcal{B}_k^{\mathrm{ow}}$ in the proof of Theorem 4.2.

input (e, d, τ) that $e = e_{t^*}$ and $\mathrm{VRFY}(d, \tau) = 1$, $\mathcal{F}$ outputs (d, τ). If $\mathcal{A}_1$ finally outputs *out* but the event Bad_7 does not occur, then $\mathcal{F}$ aborts.

If the event Bad_7 occurs, then the probability that the ciphertext (e, d, τ) (that raises this event) satisfies $e = e_{t^*}$ is $\frac{1}{\mu - n}$. The probability that $\mathcal{A}_1$ does not open c_{t^*} is $\frac{\mu - n}{\mu}$. so we have,

$$\Pr[\mathsf{Bad}_7] = \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})$$

and thus we have

$$\begin{aligned}
&\left| \Pr[\mathbf{G}_6^{\mathcal{A}} = 1] - \Pr[\mathbf{G}_7^{\mathcal{A}} = 1] \right| \\
&\leq 4(n_{\mathrm{OP}} + 1)^2 q \sqrt{2\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}^{\mathrm{ow}}) + \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})} + \frac{4\mu q}{\sqrt{|\mathcal{R}'|}} + \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})
\end{aligned}$$

Now we can construct a simulator $\mathcal{S}$ that is simulating $\mathbf{G}_7$ for $\mathcal{A}$ and interacts with the $\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}^{\mathcal{S}}$ game. Its simulation process is given in Figure 28, which is similar to the one in Figure 10, and the simulation of DEC is the same as $\mathbf{G}_7$ (without using sk), so we have

$$\left| \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}^{\mathcal{S}} \Rightarrow 1] \right| \leq \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})$$

In conclusion, for any SO-CCA adversary $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ such that

$$\begin{aligned}
&\mathsf{Adv}_{\mathsf{sPKE}}^{\mathsf{SO\text{-}CCA}}(\mathcal{A}, \mathcal{S}, \mu, \mathsf{Rel}) \\
&= \left| \Pr[\mathsf{REAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}^{\mathcal{A}}(\mathcal{A}) \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}^{\mathcal{S}}(\mathcal{S}) \Rightarrow 1] \right| \\
&\leq 3\mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F}) + 6(n_{\mathrm{OP}} + 1)^2 q \sqrt{2\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}^{\mathrm{ow}}) + \mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})} \\
&\quad + \frac{2q_H}{\sqrt{2^k}} + 16(\mu + n_{\mathrm{DEC}} + q_G + q_H + 1)^2 \delta + \frac{\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + \frac{6\mu q}{\sqrt{|\mathcal{R}'|}} + \frac{\mu n_{\mathrm{DEC}}}{|\mathcal{C}'| - n_{\mathrm{DEC}}} + \frac{(2 + \mu)q}{\sqrt{|\mathcal{M}'|}}
\end{aligned}$$

□

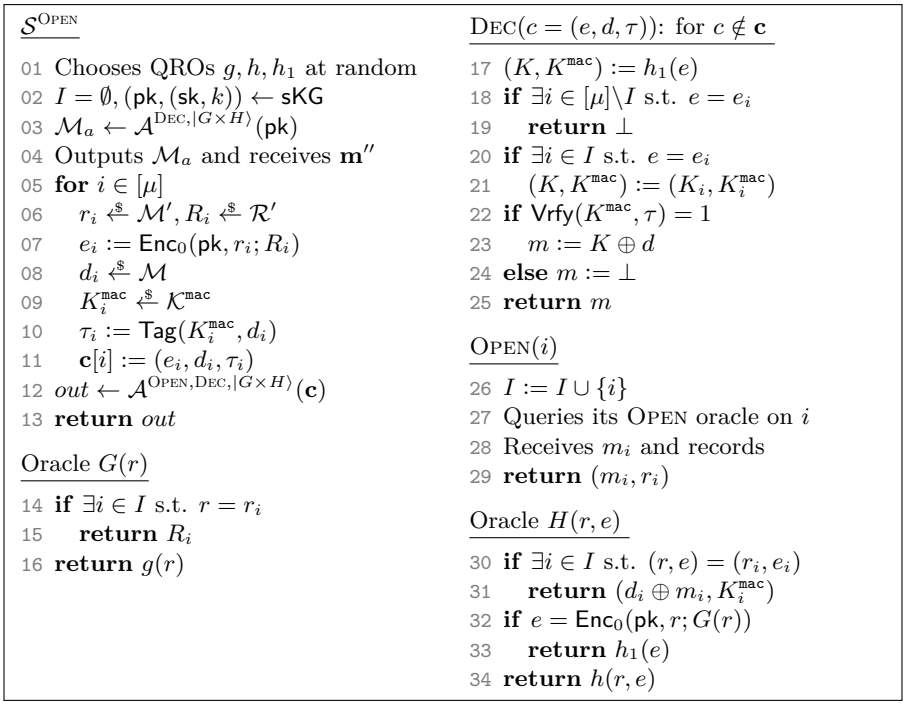

Figure 28: Simulator $\mathcal{S}$ in the proof of Theorem 4.2.

G Bi-SO Security Proof of $\mathsf{FO}^{\not\perp}$

G.1 Proof of Theorem 6.1

Similar to the proof of Theorem 4.2, we combine G and H as one random oracle $G \times H$ such that $G \times H(\mathsf{pk}', \mathsf{pk}, r', r, e) := (G(\mathsf{pk}', r'), H(\mathsf{pk}, r, e))$. $\mathcal{A}$ can query $G \times H$ at most $q = q_H + q_G$ times.

Let $g, g_{\mathsf{pk}_1}, ..., g_{\mathsf{pk}_p}$ $h, h_{\mathsf{pk}_1}, ..., h_{\mathsf{pk}_p}, h', h'_{\mathsf{pk}_1}, ..., h'_{\mathsf{pk}_p}, \hat{h}'_{\mathsf{pk}_1}, ..., \hat{h}'_{\mathsf{pk}_p}$ be internal quantum random oracles. The subscripts pk_j are just notations to distinguish these QROs. These internal QROs are used to respond G, H, H'.

In this games transition, we also consider the case that $\mathcal{A}_0$ queries DEC on (j, e, d, τ) such that $(e, d) = (e_{j,i}, d_{j,i})$ before seeing the challenge ciphertexts $\mathbf{c}$. The probability that $\mathcal{A}_0$ queries DEC on such ciphertexts is $\frac{p\mu n_{\text{DEC}}}{|\mathcal{C}'|-n_{\text{DEC}}} + \frac{p\mu q}{\sqrt{|\mathcal{M}'|}}$. For simplicity, we just add this probability into our final bound. Moreover, we also assume that there is no collision among outputs of $k_{j,i}$, $r_{j,i}$'s, $R_{j,i}$'s, $K_{j,i}$'s, and $K_{j,i}^{\mathtt{mac}}$'s. This introduce collision bounds $\frac{p^2\mu^2+p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{\mathcal{R}'} + \frac{p^2\mu^2}{|\mathcal{M}|} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + p\eta_{\mathsf{KG}_0}$. For simplicity, we just add these probability into our final bound, and do not consider it in the game sequences.

Similar to the proofs of Theorem 4.1 and Theorem 5.3, we use the encrypt-than-hash technique so that the decryption oracle can be simulated without secret key. However, in the Bi-SO setting, we cannot use this technique directly since the adversary can learn user's secret key by querying CORRUPT and then use the implicit rejection key k to determine if the game simulator uses the same internal QRO to simulate DEC for both valid and invalid ciphertext. To deal with it, we need to use our QROM reprogramming framework in Lemma 3.1.

The games sequence is given in Figure 29. We have

$$\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathsf{REAL\text{-}Bi\text{-}SO\text{-}CCA}_{\mathsf{sPKE}}^{\mathcal{A}} \Rightarrow 1]$$

Game $\mathbf{G}_1$: If $\mathcal{A}$ queries DEC$(j, (e, d, \tau))$ where e is invalid or cannot pass the re-encryption check, then the oracle computes $(K, K^{\mathtt{mac}})$ as $h'_{\mathsf{pk}_j}(e)$ instead of $H'(\mathsf{pk}_j, k_j, e)$ (see Item 48). Moreover, to make the simulation consistent, if $\mathcal{A}$ queries $H'(\mathsf{pk}_j, k_j, e)$ where pk_j is corrupted and e is an invalid ciphertext or cannot pass the re-encryption check, then $H'(\mathsf{pk}, k, e)$ returns $h'_{\mathsf{pk}}(e)$ instead of $h'(\mathsf{pk}, k, e)$ (see Items 26 to 30). The latter modification can be seen as we reprogram $H'[(\mathsf{pk}_j, k_j, e) \to \hat{h}'_{\mathsf{pk}_j}(e)]$ for

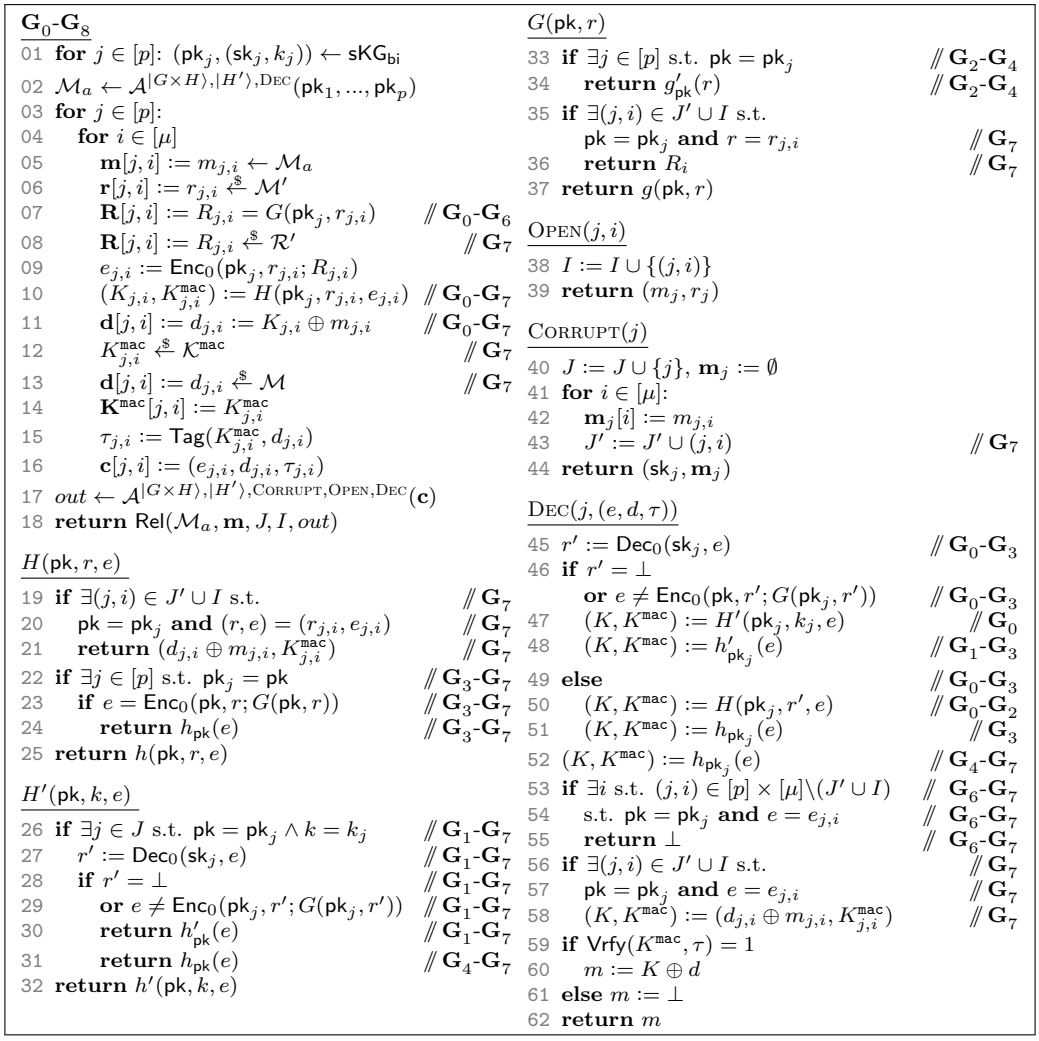

Figure 29: Games $\mathbf{G}_0$-$\mathbf{G}_8$ in the proof of Theorem 6.1

some e if $\mathcal{A}$ corrupts the user j.

We can use Lemma 2.7 to bound the difference caused by the first modification. Since there are p users in the security games, so we need to apply Lemma 2.7 here p times.

The probability difference due to the second modification can be bounded by using our QROM reprogramming framework. To use Lemma 3.1, we can split $\mathcal{A}$ into $n_{\mathrm{Co}} + 1$ stages with respect to its CORRUPT queries. In $\mathbf{G}_0$, the QRO H' does not change during the game, while in $\mathbf{G}_1$, corrupting user will make H' reprogrammed.

We can define algorithms $\mathtt{INIT}, \mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ in a natural way such that $\mathbf{G}_0$ is a NONADA game and $\mathbf{G}_1$ is a ADA game and construct an adversary $\mathcal{B}_k$ to bound the probability that $\mathcal{A}$ "queries" k_j where user j is uncorrupted. Since for all j, k_j are chosen at independently uniformly random, thus the probability that $\mathcal{B}_k$ find k_j for any $j \notin J$ is $\frac{pq_{H'}}{|\mathcal{M}'|}$, where $q_{H'}$ is the number of time that $\mathcal{A}$ queries H'. For simplicity, we ignore the details. By Lemma 2.7 and Lemma 3.1, we have

$$\left|\Pr[\mathbf{G}_0^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1]\right| \leq \frac{2(n_{\mathrm{Co}}+1)^2\sqrt{pq_{H'}} + 2pq_{H'}}{\sqrt{|\mathcal{M}'|}}$$

Game $\mathbf{G}_2$: We restrict the range of $G(\mathsf{pk}_j, \cdot)$ to be the "good" randomness space $\mathcal{R}'_{\mathsf{good}} \subset \mathcal{R}'$ respect to $(\mathsf{pk}_j, \mathsf{sk}_j)$. Specifically, let $\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_j, \mathsf{sk}_j, r)$ be the set $\{r' \in \mathcal{R}' \mid \mathsf{Dec}_0(\mathsf{sk}_j, \mathsf{Enc}_0(\mathsf{pk}_j, r; r')) = r\}$ and $\mathcal{R}'_{\mathsf{Bad}}(\mathsf{pk}_j, \mathsf{sk}_j, r)$ be the set $\mathcal{R}' \backslash \mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_j, \mathsf{sk}_j, r)$. Let $g'_{\mathsf{pk}_j} : \mathcal{M}' \to \mathcal{R}'$ be a quantum-accessible random oracle such that $g'_{\mathsf{pk}_j}(r)$ is sampled uniformly from $\mathcal{R}'_{\mathsf{good}}(\mathsf{pk}_j, \mathsf{sk}_j, r)$. We can use a similar idea of [PWZ23,

Theorem 4.4] to extend the proof in Supp. Mat. D.1 to the multi-user setting by hybrid argument. For simplicity, here we only present the probability bound. If PKE is δ-correct (see Definition 2.1), then

$$\left|\Pr[\mathbf{G}_1^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1]\right| \leq 8p(\mu + n_{\mathrm{DEC}} + q_G + q_H + q_{H'} + 1)^2\delta.$$

Game $\mathbf{G}_3$: We set $H(\mathsf{pk}_j, r, e) = h_{\mathsf{pk}_j}(e)$ if $e = \mathsf{Enc}_0(\mathsf{pk}_j, r; G(\mathsf{pk}_j, r))$ (see Items 22 to 24). Since the randomness generated by G (i.e., g'_{pk_j}) is always a "good" randomness, the map $\mathsf{Enc}_0(\mathsf{pk}_j, \cdot; G(\mathsf{pk}_j, \cdot))$ is injective and thus the map $h_{\mathsf{pk}_j}(\mathsf{Enc}_0(\mathsf{pk}_j, \cdot; G(\mathsf{pk}_j, \cdot)))$ can be also viewed as an random oracle. Therefore, we have

$$\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_4$: We "merge" h_{pk_j} and h'_{pk_j}, namely, in this game we use h_{pk_j} to replace h'_{pk_j}. Similar to the argument in the proof of Theorem 4.2, the indirect queries from $\mathcal{A}$ to h_{pk_j} and to h'_{pk_j} in $\mathbf{G}_3$ (regardless of whether user j is corrupted) are disjoint, and so we can use the same internal QRO to respond these queries. We have

$$\Pr\left[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1\right] = \Pr\left[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1\right].$$

Game $\mathbf{G}_5$: In this game, the oracle $G(\mathsf{pk}, \cdot)$ is simulated using g_{pk} instead of g'_{pk}. Similar to the difference between $\mathbf{G}_1$ and $\mathbf{G}_2$, we have

$$\left|\Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1]\right| \leq 8p(\mu + n_{\mathrm{DEC}} + q_G + q_H + q_{H'} + 1)^2\delta.$$

Game $\mathbf{G}_6$: In this game, the decryption oracle always returns $\perp$ if the adversary queries a ciphertext (e, d, τ) that e is the PKE part of some unopened ciphertexts, i.e., $\exists(j, i) \in [p] \times [\mu] \backslash (J' \cup I)$ such that $e = e_{j,i}$.

Let Bad_k be the event that $\mathcal{A}$ queries DEC on a ciphertext (e, d, τ) that $\exists(j, i) \in [p] \times [\mu] \backslash (J' \cup I)$ s.t. $e = e_{j,i}$ and $\mathsf{Vrfy}(K^{\mathtt{mac}}_{j,i}, d, \tau) = 1$ in $\mathbf{G}_k (k \geq 6)$. That is, $\mathcal{A}$ forges valid MAC codes of some unopened ciphertext. Similar to the arguments in the proof of Theorem 4.2, if Bad_k does not occur, then the winning probabilities of $\mathcal{A}$ in $\mathbf{G}_5$ and in $\mathbf{G}_6$ are the same. Thus $\left|\Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathsf{Bad}_6]$.

Game $\mathbf{G}_7$: The game simulator generates challenge ciphertexts independent of G, H. If $\mathcal{A}$ corrupted the user j (which means that all $c_{j,i}$ for $i \in [\mu]$ are opened), or opened $c_{j,i}$, then G and H will be reprogrammed such that $G(\mathsf{pk}_j, r_{j,i}) = R_{j,i}$ and $H(\mathsf{pk}_j, r_{j,i}, e_{j,i}) = d_{j,i} \oplus m_{j,i}$, and the responds of DEC are modified to make the simulation consistent. See Items 19 to 21 and Items 56 to 58.

In $\mathbf{G}_7$, there are two types of query (OPEN and CORRUPT) from $\mathcal{A}$ will make G and H be reprogrammed. In Figure 30 we define $\mathtt{INIT}$, $\mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ such that $\mathbf{G}_6$ is a NONADA game and $\mathbf{G}_7$ is an ADA game in Figure 4. The queries to H' and DEC do not influence the distribution of $G \times H$.

We firstly view $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$, where $\mathcal{A}_0$ is $\mathcal{A}$ in the stage cannot issues OPEN or CORRUPT queries and $\mathcal{A}_1$ is $\mathcal{A}$ in the stage that can issues OPEN and CORRUPT queries. Suppose that $\mathcal{A}$ corrupts at most n_{Co} users and opens at most n_{Op} challenge ciphertexts. $\mathcal{A}_1$ can be further divided into $(n_{\mathrm{Op}} + n_{\mathrm{Co}} + 1)$ stages wrt OPEN queries or CORRUPT queries: Before any OPEN or CORRUPT query (i.e., at the 0-th stage), $\mathcal{A}_1$ takes $\mathsf{in}_0 := \mathbf{c}$ as input and outputs out_0, where out_0 can be an opening index (j_1, i_1) (corresponding to issue OPEN query) or a corrupting index j_1 (corresponding to issue OPEN query); and after that, at k-th stage $(1 \leq k \leq n_{\mathrm{Op}} + n_{\mathrm{Co}} - 1)$, $\mathcal{A}_1$ receives in_k and ends the stage by outputting out_k, where in_k is the secret key of a user or the message-randomness pair of a opened ciphertext, and out_k is an opening index or a corrupting index. Finally, at the $(n_{\mathrm{Co}} + n_{\mathrm{Op}})$-th stage, $\mathcal{A}_1$ receives in_n and terminates by outputting *out*.

Moreover, at $\mathcal{A}_1$'s k-th stage, our $\mathsf{Repro}_{\mathbf{s}}$ defines a set

$$S_k := \Big\{((\mathsf{pk}_j, r_{j,i}), (\mathsf{pk}_{j'}, r_{j',i'}, e_{j',i'}))\big|(j, i) \in [p] \times [\mu] \backslash (J'_k \cup I_k) \\ \text{or } (j', i') \in [p] \times [\mu] \backslash (J'_k \cup I_k)\Big\}, \tag{21}$$

where J'_k, I_k are the lists J', I respectively just after $\mathcal{A}_1$ issues the k-th query to OPEN or CORRUPT.

Let q_k be the number of query to $G \times H$ issued by $\mathcal{A}_1$ at its k-th stage. By Lemma 3.1, there exists $\mathcal{B}_k$ for $k \in \{0, ..., n_{\mathrm{Op}} + n_{\mathrm{Co}}\}$ such that:

$$\left|\Pr\left[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_6\right] - \Pr\left[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7\right]\right|$$

$\underline{\texttt{INIT}}$

$I := \emptyset, J := \emptyset, J' := \emptyset$
for $j \in [p]$: $(\mathsf{pk}_j, (\mathsf{sk}_j, k_j)) \leftarrow \mathsf{sKG}_{\mathsf{bi}}$, $\mathbf{pk}[j] := \mathsf{pk}_j$
Chooses QROs $g, h, h', h_{\mathsf{pk}_1}, ..., h_{\mathsf{pk}_p}, g_{\mathsf{pk}_1}, ..., g_{\mathsf{pk}_p}, g'', h''$
Simulates G, H, and H' as $\mathbf{G}_6$ and Dec as $\mathbf{G}_7$ in Figure 29
$\mathcal{M}_a \leftarrow \mathcal{A}_0^{|G\times H\rangle,|H'\rangle}(\mathsf{pk}_1, ..., \mathsf{pk}_p)$
for $j \in [p]$:
  for $i \in [\mu]$
    $\mathbf{m}[j,i] := m_{j,i} \leftarrow \mathcal{M}_a$, $\mathbf{r}[j,i] := r_{j,i} \xleftarrow{\$} \mathcal{M}'$, $\mathbf{R}[j,i] := R_{j,i} = G(\mathsf{pk}_j, r_{j,i})$
    $e_{j,i} = \mathsf{Enc}_0(\mathsf{pk}_j, m_{j,i}; R_{j,i})$, $(K_{j,i}, K^{\mathtt{mac}}_{j,i}) = H(r_{j,i}, e_{j,i})$
    $d_{j,i} = K_{j,i} \oplus m_{j,i}, \mathbf{K}^{\mathtt{mac}}[j,i] := K^{\mathtt{mac}}_{j,i}$
    $\mathbf{c}[j,i] := (e_{j,i}, d_{j,i}, \tau_{j,i})$
$\mathbf{s} := (\mathcal{M}_a, \mathbf{pk}, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}, \mathbf{K}^{\mathtt{mac}})$, $\mathsf{in}_0 := \mathbf{c}$
$S_0^G := \{(\mathsf{pk}_j, r_{j,i}) | (j,i) \in [p] \times [\mu]\}$
$S_0^H := \{(\mathsf{pk}_{j'}, r_{j',i'}, e_{j',i'}) | (j',i') \in [p] \times [\mu]\}$
$S_0 := S_0^G \times S_0^H$
Let $G_0 \times H_0(x) := \begin{cases} G \times H(x), & (x \notin S_0) \\ g'' \times h''(x), & (\text{else}) \end{cases}$
return $((\mathbf{s}, \mathsf{in}_0), G \times H, G_0 \times H_0)$

$\underline{\mathsf{F}_{\mathbf{s}}(\mathsf{out})}$

$(\mathcal{M}_a, \mathbf{pk}, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}, \mathbf{K}^{\mathtt{mac}}) := \mathbf{s}$
if out is an integer (denoted as j): // Corrupt query
  $\mathsf{pk}_j := \mathbf{pk}[j]$, $\mathbf{m}_j := \emptyset$
  for $i \in [\mu] : \mathbf{m}_j[i] := m_{j,i} := \mathbf{m}[j,i]$
  $\mathsf{in} := (\mathsf{sk}_j, \mathbf{m}_j), \mathsf{in}' := j$
else // Open query
  $(j,i) := \mathsf{out}, r_{j,i} := \mathbf{r}[j,i], m_{j,i} := \mathbf{m}[j,i]$
  $\mathsf{in} := (r_{j,i}, m_{j,i}), \mathsf{in}' := (j,i)$
return $(\mathsf{in}, \mathsf{in}')$

$\underline{\mathsf{Repro}_{\mathbf{s}}(G \times H, \mathsf{in}')}$

$(\mathcal{M}_a, \mathbf{pk}, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}, \mathbf{K}^{\mathtt{mac}}) := \mathbf{s}$
if in' is an integer (denoted as j): // Corrupt query
  $J := J \cup \{j\}$, $\mathsf{pk}_j := \mathbf{pk}[j]$,
  for $i \in [\mu]$
    $J' := J' \cup (j,i), r_{j,i} := \mathbf{r}[j,i], R_{j,i} := \mathbf{R}[j,i], G := G[(\mathsf{pk}_j, r_{j,i}) \to R_{j,i}]$
    $(e_{j,i}, d_{j,i}, \tau_{j,i}) := \mathbf{c}[j,i], m_{j,i} := \mathbf{m}[j,i], K^{\mathtt{mac}}_{j,i} := \mathbf{K}^{\mathtt{mac}}[j,i]$
    $H := H[(\mathsf{pk}_j, r_{j,i}, e_{j,i}) \to (d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})]$
else // Open query
  $(j,i) := \mathsf{in}'$
  $I := I \cup \{(j,i)\}$, $\mathsf{pk}_j := \mathbf{pk}[j], (e_{j,i}, d_{j,i}, \tau_{j,i}) := \mathbf{c}[j,i]$
  $r_{j,i} := \mathbf{r}[j,i], R_{j,i} := \mathbf{R}[j,i], m_{j,i} := \mathbf{m}[j,i], K^{\mathtt{mac}}_{j,i} := \mathbf{K}^{\mathtt{mac}}[j,i]$
  $G := G[(\mathsf{pk}_j, r_{j,i}) \to R_{j,i}], H := H[(\mathsf{pk}_j, r_{j,i}, e_{j,i}) \to (d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})]$
return $G \times H$

Figure 30: Constructions of $\texttt{INIT}$, $\mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ in the proof of Theorem 6.1. We assume that all pk_j's , k_j's, $d_{j,i}$'s, $K^{\mathtt{mac}}_{j,i}$'s are different. I, J, J' are publicly accessible.

$$\leq \sum_{k=0}^{n_{\mathrm{Co}}+n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[x' \leftarrow \mathcal{B}_i^{|G\times H\rangle,|H'\rangle,\mathrm{DEC}} \text{ s.t. } x' \in S_i : \mathbf{G}_7^{\mathcal{B}_i}\right]} + \frac{2p\mu q}{\sqrt{|\mathcal{R}'|}},$$

where sets S_k is defined as Equation (21). Based on $\mathcal{B}_k$, we can construct adversaries $\mathcal{B}_k^{\mathrm{ow}}$ against the OW-CPA security of PKE. See Figure 31. By the construction of $\mathcal{B}_k^{\mathrm{ow}}$, as long as $\mathcal{A}$ does not corrupt j^*, $\mathcal{B}_k^{\mathrm{ow}}$ can always simulate DEC and $\mathbf{G}_8$

$\underline{\mathcal{B}_k^{\mathrm{ow}}(\mathsf{pk}^*, e^*)}$ // (pk^*, e^*) is a OW-challenge of PKE. $\mathcal{B}_k^{\mathrm{ow}}$ simulates $\mathbf{G}_8$ for $\mathcal{B}_k$

$j^* \xleftarrow{\$} [p], i^* \xleftarrow{\$} [\mu]$
$((\mathbf{s}, \mathsf{in}_0), (G \times H), (G_0 \times H_0)) \leftarrow \mathtt{INIT}$ // using $\mathsf{pk}_{j^*} := \mathsf{pk}^*$ in Figure 30
parse $(\mathcal{M}_a, \mathbf{pk}, \mathbf{m}, \mathbf{r}, \mathbf{R}, \mathbf{c}, \mathbf{K}^{\mathtt{mac}}) := \mathbf{s}, \mathbf{c} := \mathsf{in}_0$
$(e_{j^*,i^*}, d_{j^*,i^*}, \tau_{j^*,i^*}) := \mathbf{c}[j^*, i^*]$
$\mathbf{c}[j^*, i^*] := (e^*, d_{j^*,i^*}, \tau_{j^*,i^*})$, $\mathsf{in}_0 := \mathbf{c}$ // embed the challenge
if $k = 0$: **goto** line 15
$\mathsf{out}_0 \leftarrow \mathcal{B}_k^{|G_0\times H_0\rangle,|H'\rangle,\mathrm{DEC}}(\mathsf{in}_0)$ // out_t has the form j or (j, i)
if $\mathsf{out}_0 = j^*$ **or** $\mathsf{out}_0 = (j^*, i^*)$: **abort**
$(\mathsf{in}_1, \mathsf{in}_1') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_0)$, $(G_1 \times H_1) := \mathsf{Repro}_{\mathbf{s}}(G_0 \times H_0, \mathsf{in}_1')$
for $t = 1$ **to** $k - 1$:
  $\mathsf{out}_t \leftarrow \mathcal{B}_k^{|G_t\times H_t\rangle,|H'\rangle,\mathrm{DEC}}(\mathsf{in}_t)$
  if $\mathsf{out}_t = j^*$ **or** $\mathsf{out}_t = (j^*, i^*)$: **abort**
  $(\mathsf{in}_{t+1}, \mathsf{in}_{t+1}') := \mathsf{F}_{\mathbf{s}}(\mathsf{out}_t)$
  $(G_{t+1} \times H_{t+1}) := \mathsf{Repro}_{\mathbf{s}}(G_t \times H_t, \mathsf{in}_{t+1}')$
$(r_0', (r_1', e')) \leftarrow \mathcal{B}_k^{|G_k\times H_k\rangle,|H'\rangle,\mathrm{DEC}}(\mathsf{in}_k)$
$b \xleftarrow{\$} \{0, 1\}, r^* := r_b'$ // randomly choose a solution
return r^*

Figure 31: The detailed constructions of OW-CPA adversaries $\mathcal{B}_k^{\mathrm{ow}}, 0 \leq k \leq n_{\mathrm{Co}} + n_{\mathrm{OP}}$.

By the construction of $\mathcal{B}_k^{\mathrm{ow}}$, if $\mathcal{A}_1$ does not corrupt user j^* and does not open c_{j^*,i^*}, and r or r' equals the solutions of e^*, then $\mathcal{B}_k^{\mathrm{ow}}$ wins. So the winning probability for $\mathcal{B}_j^{\mathrm{ow}}$ to breaks the OW-CPA challenge is:

$$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_k^{\mathrm{ow}}) = \frac{1}{2p\mu} \Pr\left[x \leftarrow \mathcal{B}_k^{|G\times H\rangle,|H'\rangle,\mathrm{DEC}} \text{ s.t. } x \in S_k : \mathbf{G}_7^{\mathcal{B}_k}\right],$$

and thus we have

$$\begin{aligned}
&\left|\Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_6] - \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7]\right| \\
&\leq \sum_{k=0}^{n_{\mathrm{Co}}+n_{\mathrm{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{2p\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}_i^{\mathrm{ow}}) + \Pr[\mathsf{Bad}_7]} + \frac{2p\mu q}{\sqrt{|\mathcal{R}'|}}
\end{aligned}$$

Let $\mathcal{B}^{\mathrm{ow}}$ be the adversary that has highest advantage against PKE among $\{\mathcal{B}_k^{\mathrm{ow}}\}_{k\in\{0,\ldots,n_{\mathrm{Co}}+n_{\mathrm{OP}}\}}$.

$$\begin{aligned}
&\left|\Pr\left[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_6\right] - \Pr\left[\mathbf{G}_7^{\mathcal{A}} \Rightarrow 1 | \neg\mathsf{Bad}_7\right]\right| \\
&\leq 2(n_{\mathrm{Co}} + n_{\mathrm{OP}} + 1)^2 q \sqrt{2p\mu \mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{OW\text{-}CPA}}(\mathcal{B}^{\mathrm{ow}}) + \Pr[\mathsf{Bad}_7]} + \frac{2p\mu q}{\sqrt{|\mathcal{R}'|}} \qquad (22)
\end{aligned}$$

Now we can construct an otSUF-CMA reduction to bound $\Pr[\mathsf{Bad}_7]$. Let $\mathcal{F}$ be an otSUF-CMA adversary that simulates $\mathbf{G}_7$ for $\mathcal{A}$. It firstly chooses $j^* \xleftarrow{\$} [p], i^* \xleftarrow{\$} [\mu]$ uniformly and sets $K_{j^*,i^*}^{\mathtt{mac}} := \perp$. To generates τ_{j^*,i^*}, $\mathcal{F}$ queries its TAG oracle on d_{j^*,i^*} and sets the responding tag as τ_{j^*,i^*}. $\mathcal{F}$ aborts the game if $\mathcal{A}$ opens c_{j^*,i^*} or corrupts j^*. When $\mathcal{A}_1$ queries the DEC oracle on input $(\mathsf{pk}_j, e, d, \tau)$ that $e = e_{j^*,i^*}$ and $\mathrm{VRFY}(d, \tau) = 1$, $\mathcal{F}$ outputs (d, τ). If $\mathcal{A}_1$ finally outputs out but the event Bad_7 does not occur, then $\mathcal{F}$ aborts. Similar to the argument in the proof of Theorem 6.1, we have

$$\Pr[\mathsf{Bad}_7] = p\mu \mathsf{Adv}_{\mathsf{MAC}}^{\mathsf{otSUF\text{-}CMA}}(\mathcal{F})$$

Since the events Bad_6 and Bad_7 are well defined classical event, it can be detected by the game simulator when responding the DEC queries. By Lemma 3.1 and the bound of $|\Pr[\mathbf{G}_6^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_7^{\mathcal{A}} \Rightarrow$

$\mathcal{S}^{\text{Open}',\text{Corrupt}'}$

for $j \in [p]$: $(\mathsf{pk}_j, (\mathsf{sk}_j, k_j)) \leftarrow \mathsf{sKG}_{\mathsf{bi}}$
$\mathcal{M}_a \leftarrow \mathcal{A}^{\text{Dec},|G\times H\rangle,|H'\rangle}(\mathsf{pk}_1, ..., \mathsf{pk}_p)$
Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$
for $j \in [p]$:
  for $i \in [\mu]$
    $\mathbf{r}[j,i] := r_{j,i} \xleftarrow{\$} \mathcal{M}'$
    $\mathbf{R}[j,i] := R_{j,i} \xleftarrow{\$} \mathcal{R}'$
    $e_{j,i} = \mathsf{Enc}_0(\mathsf{pk}_j, r_{j,i}; R_{j,i})$
    $\mathbf{d}[j,i] := d_{j,i} \xleftarrow{\$} \mathcal{M}$
    $\mathbf{K}^{\mathtt{mac}}[j,i] := K^{\mathtt{mac}}_{j,i} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$
    $\tau_{j,i} := \mathsf{Tag}(K^{\mathtt{mac}}_{j,i}, d_{j,i})$
    $\mathbf{c}[j,i] := (e_{j,i}, d_{j,i}, \tau_{j,i})$
$out \leftarrow \mathcal{A}^{\text{Open},\text{Corrupt},\text{Dec},|G\times H\rangle,|H'\rangle}(\mathbf{c})$
return out

$H(\mathsf{pk}, r, e)$

if $\exists (j,i) \in J' \cup I$ s.t.
  $\mathsf{pk} = \mathsf{pk}_j$ **and** $(r,e) = (r_{j,i}, e_{j,i})$
  return $(d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})$
if $\exists j \in [p]$ s.t. $\mathsf{pk}_j = \mathsf{pk}$
  if $e = \mathsf{Enc}_0(\mathsf{pk}, r; G(\mathsf{pk}, r))$
    return $h_{\mathsf{pk}}(e)$
return $h(\mathsf{pk}, r, e)$

$H'(\mathsf{pk}, k, e)$

if $\exists j \in J$ s.t. $\mathsf{pk} = \mathsf{pk}_j \wedge k = k_j$
  $r' := \mathsf{Dec}_0(\mathsf{sk}_j, e)$
  if $r' = \perp$ **or** $e \neq \mathsf{Enc}_0(\mathsf{pk}_j, r'; G(\mathsf{pk}_j, r'))$
    return $h_{\mathsf{pk}}(e)$
return $h'(\mathsf{pk}, k, e)$

$G(\mathsf{pk}, r)$

if $\exists (j,i) \in J' \cup I$ s.t.
  $\mathsf{pk} = \mathsf{pk}_j$ **and** $r = r_{j,i}$
  return R_i
return $g(\mathsf{pk}, r)$

$\text{Open}(j,i)$

$I := I \cup \{(j,i)\}$
Queries $\text{Open}'(j,i)$ and gets $m_{j,i}$
return $(m_{j,i}, r_{j,i})$

$\text{Corrupt}(j)$

$J := J \cup \{j\}$
Queries $\text{Corrupt}'(j)$ and gets $\mathbf{m}_j$
for $i \in [\mu]$
  $J' := J' \cup (j,i)$, $m_{j,i} := \mathbf{m}_j[i]$
return $(\mathsf{sk}_j, \mathbf{m}_j)$

$\text{Dec}(j, (e, d, \tau))$

$(K, K^{\mathtt{mac}}) := h_{\mathsf{pk}_j}(e)$
if $\exists i$ s.t. $(j,i) \in [p] \times [\mu] \backslash (J' \cup I)$
  s.t. $\mathsf{pk} = \mathsf{pk}_j$ **and** $e = e_{j,i}$
  return $\perp$
if $\exists (j,i) \in J' \cup I$ s.t.
  $\mathsf{pk} = \mathsf{pk}_j$ **and** $e = e_{j,i}$
  $(K, K^{\mathtt{mac}}) := (d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \perp$
return m

Figure 32: The simulator of the proof of Theorem 6.1

1]|, there exists adversaries $\mathcal{B}^{\text{ow}}$ and $\mathcal{F}$ such that

$$\begin{aligned}&\left|\Pr[\mathsf{Bad}_6] - \Pr[\mathsf{Bad}_7]\right| \\ &\leq 2(n_{\text{Co}} + n_{\text{Op}} + 1)^2 q\sqrt{2p\mu\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{\text{ow}}) + p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} + \frac{2p\mu q}{\sqrt{|\mathcal{R}'|}},\end{aligned}$$

and thus we have

$$\begin{aligned}&\left|\Pr\left[\mathbf{G}^{\mathcal{A}}_6 \Rightarrow 1\right] - \Pr\left[\mathbf{G}^{\mathcal{A}}_7 \Rightarrow 1\right]\right| \\ &\leq 4(n_{\text{Co}} + n_{\text{Op}} + 1)^2 q\sqrt{2p\mu\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{\text{ow}}) + p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} \\ &\quad + p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + \frac{4p\mu q}{\sqrt{|\mathcal{R}'|}}\end{aligned}$$

Now we can construct a simulator $\mathcal{S}$ that interacts with the $\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}_{\mathsf{bi}}}$ game and simulates $\mathbf{G}_7$ for $\mathcal{A}$. $\mathcal{S}$'s construction is shown in Figure 32, and it perfectly simulates $\mathbf{G}_7$ for $\mathcal{A}$ except that Bad_7 happens, and thus

$$|\Pr[\mathbf{G}^{\mathcal{A}}_7 \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}_{\mathsf{bi}}} \Rightarrow 1]| \leq p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}).$$

Therefore,

$$\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}CCA}}_{\mathsf{sPKE}_{\mathsf{bi}}}(\mathcal{A}, \mathcal{S}, p, \mu, \mathsf{Rel})$$

$$
\begin{aligned}
&\leq |\Pr[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}_{\mathsf{bi}}} \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}_{\mathsf{bi}}} \Rightarrow 1]| \\
&\leq 6(n_{\mathrm{Co}} + n_{\mathrm{OP}} + 1)^2 q\sqrt{2p\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{B}^{\mathrm{ow}}) + p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})} + p\eta_{\mathsf{KG}_0} \\
&\quad + 3p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + \frac{p\mu n_{\mathrm{DEC}}}{|\mathcal{C}'| - n_{\mathrm{DEC}}} + \frac{p^2\mu^2 + p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{\mathcal{R}'} + \frac{p^2\mu^2}{|\mathcal{M}|} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} \\
&\quad + \frac{6p\mu q}{\sqrt{|\mathcal{R}'|}} + 16p(\mu + n_{\mathrm{DEC}} + q + q_{H'} + 1)^2\delta + \frac{2(n_{\mathrm{Co}} + 1)^2\sqrt{pq_{H'}} + 2pq_{H'} + p\mu q}{\sqrt{|\mathcal{M}'|}}
\end{aligned}
$$

H Bi-SO Security Proof of $\mathsf{U}^{\not\perp}_{\mathsf{m}}$

In NTRU KEM [CDH+20], its randomness space is the message space of the underlying deterministic PKE, and its encapsulation samples randomness according some specific distribution $\mathcal{D}_{\mathcal{M}'}$. Hence, we consider $\mathcal{D}_{\mathcal{M}'}$-OW-CPA security, namely, OW-CPA security with challenge messages chosen following $\mathcal{D}_{\mathcal{M}'}$. We require that $\mathcal{D}_{\mathcal{M}'}$ itself has high minimum entropy. Otherwise, $\mathcal{D}_{\mathcal{M}'}$-OW-CPA security can be broken trivially.

Definition H.1 ($\mathcal{D}_{\mathcal{M}'}$-OW-CPA). Let PKE be a deterministic PKE with message space $\mathcal{M}$, and let $\mathcal{D}_{\mathcal{M}'}$ be some distribution over $\mathcal{M}$. For an adversary $\mathcal{A}$, its advantage against $\mathcal{D}_{\mathcal{M}'}$-OW-CPA security of PKE is defined as

$$
\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}'}}(\mathcal{A}) := \Pr\Big[m' = m^* : (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}, m^* \leftarrow \mathcal{D}_{\mathcal{M}'}, \\
c^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m^*), m' \leftarrow \mathcal{A}(\mathsf{pk}, c^*)\Big].
$$

PKE is $\mathcal{D}_{\mathcal{M}'}$-OW-CPA secure if for all adversaries $\mathcal{A}$ $\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE}}(\mathcal{A}) = \mathsf{negl}(\lambda)$.

Theorem 6.2. The proof idea is the same as the idea of Theorem 6.1 and the proof has almost the same structure with the proof of Theorem 6.1, except that now PKE is deterministic and has perfect correctness and rigidity [BP18].

Similar to the proof of Theorem 6.1, in this proof, we let $h, h_{\mathsf{pk}_1}, ..., h_{\mathsf{pk}_p}, h', h'_{\mathsf{pk}_1}, ..., h'_{\mathsf{pk}_p}$ be internal QROs. The subscripts pk_j are just notations to distinguish these QROs. These internal QROs are used to respond H and H'.

In this games transition, we consider the case that $\mathcal{A}_0$ queries DEC on (j, e, d, τ) such that $(e, d) = (e_{j,i}, d_{j,i})$ before seeing the challenge ciphertexts $\mathbf{c}$. The probability that $\mathcal{A}_0$ queries DEC on such ciphertexts is $\frac{p\mu n_{\mathrm{DEC}}}{|\mathcal{C}'| - n_{\mathrm{DEC}}} + \frac{p\mu q}{\sqrt{|\mathcal{M}'|}}$. For simplicity, we just add this probability into our final bound. Moreover, we also assume that there is no collision among outputs of k_i, r_i's, K_i's, and $K^{\mathtt{mac}}_i$'s. This introduce collision bounds $\frac{p^2\mu^2+p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{|\mathcal{M}|} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + p\eta_{\mathsf{KG}_0}$. For simplicity, we just add these probability into our final bound. The games sequence is given in Figure 33. We have

$$
\Pr[\mathbf{G}^{\mathcal{A}}_0 \Rightarrow 1] = \Pr[\mathsf{REAL\text{-}Bi\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}} \Rightarrow 1]
$$

Game $\mathbf{G}_1$: If $\mathcal{A}$ queries $\mathrm{DEC}(j, (e, d, \tau))$ where e is invalid or cannot pass the re-encryption check, then the oracle computes $(K, K^{\mathtt{mac}})$ as $h'_{\mathsf{pk}_j}(e)$ instead of $H'(\mathsf{pk}_j, k_j, e)$. Moreover, to make the simulation consistent, if $\mathcal{A}$ queries $H'(\mathsf{pk}_j, k_j, e)$ where pk_j is corrupted and e is an invalid ciphertext, then $H'(\mathsf{pk}, k, e)$ returns $h'_{\mathsf{pk}}(e)$ instead of $h'(\mathsf{pk}, k, e)$. The latter modification can be seen as we reprogram $H'[(\mathsf{pk}_j, k_j, e) \to \hat{h}'_{\mathsf{pk}_j}(e)]$ for some e if $\mathcal{A}$ corrupts the user j. Similar to the argument in the proof of Theorem 6.1 in Supp. Mat. G.1, by using Lemma 2.7 and Lemma 3.1, we have

$$
\big|\Pr[\mathbf{G}^{\mathcal{A}}_0 \Rightarrow 1] - \Pr[\mathbf{G}^{\mathcal{A}}_1 \Rightarrow 1]\big| \leq \frac{2(n_{\mathrm{Co}} + 1)^2\sqrt{pq_{H'}} + 2pq_{H'}}{\sqrt{|\mathcal{M}'|}}
$$

Game $\mathbf{G}_2$: We set $H(\mathsf{pk}_j, r) = h_{\mathsf{pk}_j}(\mathsf{Enc}_0(\mathsf{pk}_j, r))$. Since PKE is rigid correct, the map $\mathsf{Enc}_0(\mathsf{pk}_j, \cdot)$ is injective and thus $h_{\mathsf{pk}_j}(\mathsf{Enc}_0(\mathsf{pk}_j, \cdot))$ can be also viewed as an random oracle. Therefore, we have

$$
\Pr[\mathbf{G}^{\mathcal{A}}_1 \Rightarrow 1] = \Pr[\mathbf{G}^{\mathcal{A}}_2 \Rightarrow 1].
$$

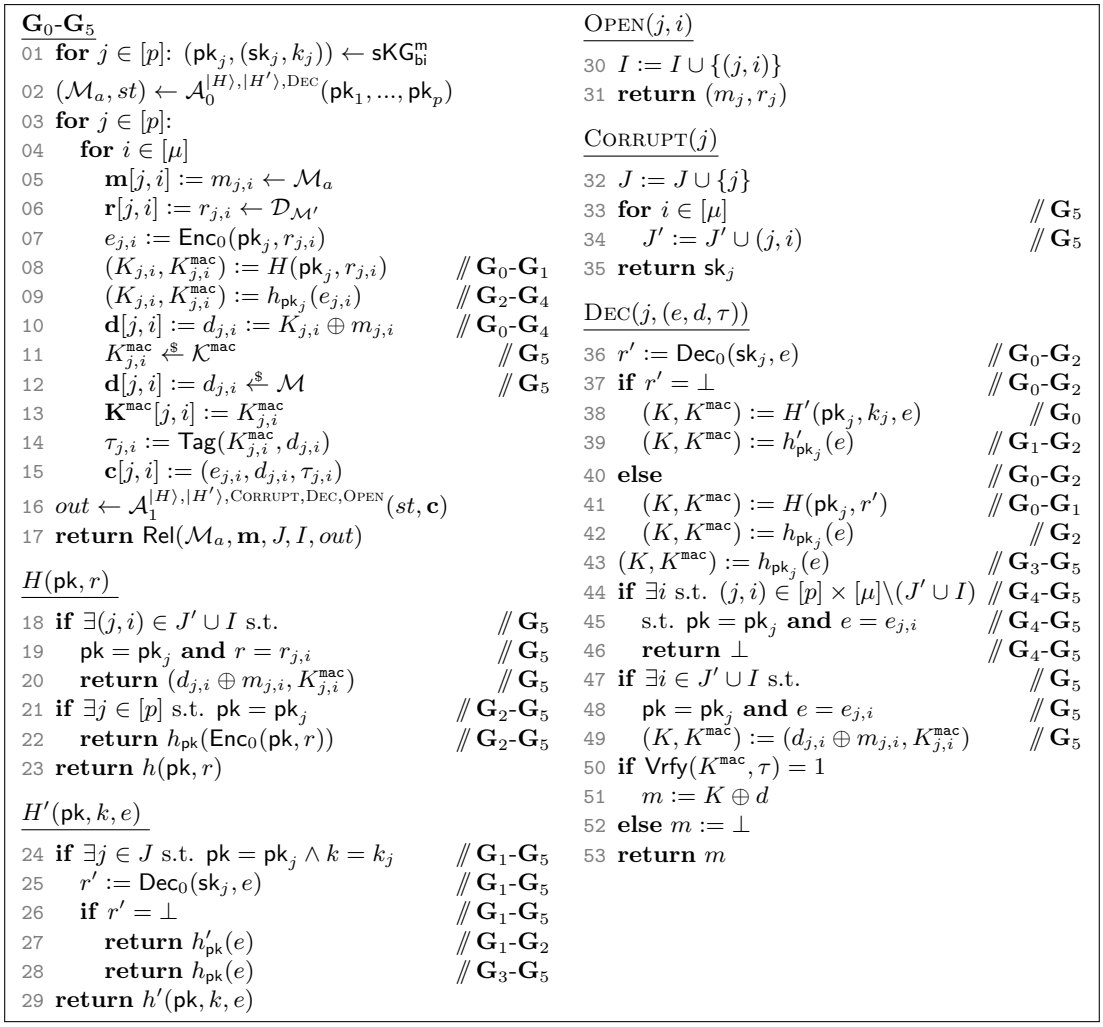

Figure 33: Games $\mathbf{G}_0$-$\mathbf{G}_6$ in the proof of Theorem 6.2

Game $\mathbf{G}_3$: We "merge" h_{pk_j} and h'_{pk_j} , namely, we use h_{pk_j} to replace h'_{pk_j}. This modification does not change $\mathcal{A}$'s view since all indirect queries from $\mathcal{A}$ to h_{pk_j} and to h'_{pk_j} in $\mathbf{G}_2$ (regardless of whether user j is corrupted) are disjoint. We have

$$\Pr[\mathbf{G}_2^{\mathcal{A}} \Rightarrow 1] = \Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1].$$

Game $\mathbf{G}_4$: In this game, the decryption oracle always returns $\perp$ if the adversary queries a ciphertext (e, d, τ) that e is the PKE part of some unopened ciphertexts, i.e., $\exists (j,i) \in [p] \times [\mu] \backslash (J' \cup I), e = e_{j,i}$.

Let Bad_k be the event that $\mathcal{A}$ queries $\textsc{Dec}$ on a ciphertext (e, d, τ) that $\exists (j,i) \in [p] \times [\mu] \backslash (J' \cup I)$ s.t. $e = e_{j,i}$ and $\mathsf{Vrfy}(K^{\mathtt{mac}}_{j,i}, d, \tau) = 1$ in $\mathbf{G}_k (k \geq 4)$. Similar to the arguments in the proof of Theorem 4.2, if Bad_k does not occur, then the winning probabilities of $\mathcal{A}$ in $\mathbf{G}_3$ and in $\mathbf{G}_4$ are the same. Thus $\left|\Pr[\mathbf{G}_3^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1]\right| \leq \Pr[\mathsf{Bad}_4]$.

Game $\mathbf{G}_5$: the game simulator generates challenge ciphertexts independent of H. If $\mathcal{A}$ corrupted the user j (which means that all $c_{j,i}$ for $i \in [\mu]$ are opened), or opened $c_{j,i}$, then H will be reprogrammed such that $H(\mathsf{pk}_j, r_{j,i}) = (d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})$, and the responds of $\textsc{Dec}$ are modified to make the simulation consistent.

Similar to the proof of Theorem 6.1, there are two types of query ($\textsc{Open}$ and $\textsc{Corrupt}$) from $\mathcal{A}$ will make H be reprogrammed. Lemma 3.1 still can be used here. One can define algorithms $\mathtt{INIT}$, $\mathsf{F}_{\mathbf{s}}$, and $\mathsf{Repro}_{\mathbf{s}}$ such that $\mathbf{G}_4$ is a $\textsc{NonAda}$ game and $\mathbf{G}_5$ is an $\textsc{Ada}$ game. Since such algorithms have almost the same structure with those in the proof of Theorem 6.1 (the main difference is that here we do not

$\underline{\mathcal{S}^{\text{Open}',\text{Corrupt}'}}$

for $j \in [p]$: $(\mathsf{pk}_j, (\mathsf{sk}_j, k_j)) \leftarrow \mathsf{sKG}^{\mathsf{m}}_{\mathsf{bi}}$
$\mathcal{M}_a \leftarrow \mathcal{A}^{\text{Dec},|H\rangle,|H'\rangle}(\mathsf{pk}_1, ..., \mathsf{pk}_p)$
Outputs $\mathcal{M}_a$ and receives $\mathbf{m}''$
for $j \in [p]$:
  for $i \in [\mu]$
    $\mathbf{r}[j,i] := r_{j,i} \leftarrow \mathcal{D}_{\mathcal{M}'}$
    $e_{j,i} := \mathsf{Enc}_0(\mathsf{pk}_j, r_{j,i})$
    $\mathbf{d}[j,i] := d_{j,i} \xleftarrow{\$} \mathcal{M}$
    $\mathbf{K}^{\mathtt{mac}}[j,i] := K^{\mathtt{mac}}_{j,i} \xleftarrow{\$} \mathcal{K}^{\mathtt{mac}}$
    $\tau_{j,i} := \mathsf{Tag}(K^{\mathtt{mac}}_{j,i}, d_{j,i})$
    $\mathbf{c}[j,i] := (e_{j,i}, d_{j,i}, \tau_{j,i})$
$out \leftarrow \mathcal{A}^{\text{Open},\text{Corrupt},\text{Dec},|H\rangle,|H'\rangle}(\mathbf{c})$
return out

$\underline{H(\mathsf{pk}, r)}$

if $\exists (j,i) \in J' \cup I$ s.t.
  $\mathsf{pk} = \mathsf{pk}_j$ **and** $r = r_{j,i}$
  return $(d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})$
if $\exists j \in [p]$ s.t. $\mathsf{pk} = \mathsf{pk}_j$
  return $h_{\mathsf{pk}}(\mathsf{Enc}_0(\mathsf{pk}, r))$
return $h(\mathsf{pk}, r)$

$\underline{H'(\mathsf{pk}, k, e)}$

if $\exists j \in J$ s.t. $\mathsf{pk} = \mathsf{pk}_j \wedge k = k_j$
  $r' := \mathsf{Dec}_0(\mathsf{sk}_j, e)$
  if $r' = \perp$
    return $h_{\mathsf{pk}}(e)$
return $h'(\mathsf{pk}, k, e)$

$\underline{\text{Open}(j,i)}$

$I := I \cup \{(j,i)\}$
Queries $\text{Open}'(j,i)$ and gets $m_{j,i}$
return (m_j, r_j)

$\underline{\text{Corrupt}(j)}$

$J := J \cup \{j\}$
Queries $\text{Corrupt}'(j)$ and gets $\mathbf{m}_j$
for $i \in [\mu]$
  $J' := J' \cup (j,i)$, $m_{j,i} := \mathbf{m}_j[i]$
return $(\mathsf{sk}_j, \mathbf{m}_j)$

$\underline{\text{Dec}(j, (e, d, \tau))}$

$(K, K^{\mathtt{mac}}) := h_{\mathsf{pk}_j}(e)$
if $\exists i$ s.t. $(j,i) \in [p] \times [\mu] \backslash (J' \cup I)$
  s.t. $\mathsf{pk} = \mathsf{pk}_j \wedge e = e_{j,i}$
  return $\perp$
if $\exists i \in J' \cup I$ s.t.
  $\mathsf{pk} = \mathsf{pk}_j \wedge e = e_{j,i}$
  $(K, K^{\mathtt{mac}}) := (d_{j,i} \oplus m_{j,i}, K^{\mathtt{mac}}_{j,i})$
if $\mathsf{Vrfy}(K^{\mathtt{mac}}, \tau) = 1$
  $m := K \oplus d$
else $m := \perp$
return m

Figure 34: The simulator of the proof of Theorem 6.2

need the QRO G), for sake of simplicity, we ignore the details here, and only give a final bound.

$$\begin{aligned}
&\left|\Pr\left[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1 | \neg \mathsf{Bad}_4\right] - \Pr\left[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1 | \neg \mathsf{Bad}_5\right]\right| \\
&\leq \sum_{k=0}^{n_{\text{Co}}+n_{\text{OP}}} \sum_{i=0}^{k} 2q_i \sqrt{\Pr\left[x' \leftarrow \mathcal{B}_i^{|G\times H\rangle,|H'\rangle,\text{Dec}} \text{ s.t. } x' \in S_i : \mathbf{G}_5^{\mathcal{B}_i}\right]} + \frac{2p\mu q}{2^{\epsilon_{\mathcal{D}_{\mathcal{M}'}}}}.
\end{aligned}$$

Here $\epsilon_{\mathcal{D}_{\mathcal{M}'}}$ is the minimal entropy of the distribution $\mathcal{D}_{\mathcal{M}'}$. As we did in the proof of Theorem 6.1, here we can also construct $\mathcal{B}^{\text{ow}}$ and $\mathcal{F}$ such that

$$\begin{aligned}
&\left|\Pr\left[\mathbf{G}_4^{\mathcal{A}} \Rightarrow 1\right] - \Pr\left[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1\right]\right| \\
&\leq p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F}) + \frac{4p\mu q}{2^{\epsilon_{\mathcal{D}_{\mathcal{M}'}}}} \\
&\quad + 4(n_{\text{Co}} + n_{\text{OP}} + 1)^2 q \sqrt{2p\mu \mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}'}}(\mathcal{B}^{\text{ow}}) + p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})}
\end{aligned}$$

Now we can construct a simulator $\mathcal{S}$ that interacts with the $\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}}$ game and simulates $\mathbf{G}_5$ for $\mathcal{A}$. $\mathcal{S}$'s construction is shown in Figure 32, and it perfectly simulates $\mathbf{G}_5$ for $\mathcal{A}$, and thus $|\Pr[\mathbf{G}_5^{\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}} \Rightarrow 1]| \leq p\mu \mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})$. Therefore,

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{Bi\text{-}SO\text{-}CCA}}_{\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}}(\mathcal{A}, \mathcal{S}, p, \mu, \lambda) \\
&\leq |\Pr[\mathsf{REAL\text{-}SO\text{-}CCA}^{\mathcal{A}}_{\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}} \Rightarrow 1] - \Pr[\mathsf{IDEAL\text{-}SO\text{-}CCA}^{\mathcal{S}}_{\mathsf{sPKE}^{\mathsf{m}}_{\mathsf{bi}}} \Rightarrow 1]| \\
&\leq \frac{p\mu n_{\text{Dec}}}{|\mathcal{C}'| - n_{\text{Dec}}} + \frac{p^2\mu^2 + p^2}{|\mathcal{M}'|} + \frac{p^2\mu^2}{|\mathcal{M}|} + \frac{p^2\mu^2}{|\mathcal{K}^{\mathtt{mac}}|} + p\eta_{\mathsf{KG}_0} + \frac{6p\mu q}{2^{\epsilon_{\mathcal{D}_{\mathcal{M}'}}}}
\end{aligned}$$

$$+ \frac{2(n_{\mathrm{Co}}+1)^2\sqrt{pq_{H'}} + 2pq_{H'} + p\mu q}{\sqrt{|\mathcal{M}'|}} + 3p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})$$
$$+ 6(n_{\mathrm{Co}}+n_{\mathrm{OP}}+1)^2 q\sqrt{2p\mu\mathsf{Adv}^{\mathsf{OW\text{-}CPA}}_{\mathsf{PKE},\mathcal{D}_{\mathcal{M}'}}(\mathcal{B}^{\mathrm{ow}}) + p\mu\mathsf{Adv}^{\mathsf{otSUF\text{-}CMA}}_{\mathsf{MAC}}(\mathcal{F})}$$

□