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 $\text{FO}^{\perp}$ (Hofheinz, Hövelmanns, and Kiltz, TCC 2017) and $\text{U}^{\perp}_m$ (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 $\text{FO}^{\perp}$ 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 $\text{FO}^{\perp}$ and $\text{U}^{\perp}_m$ 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 indistinguishability against chosen-ciphertext attacks (IND-CCA), which states that confidentiality holds even if an adversary $A$ 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, BDYW12, 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 Fujisaki-Okamoto (FO) transformation [FO99a, FO99b, FO13, HK17]. 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, $\text{FO}^\oplus$ [JZC+18] and $\text{U}_m^\otimes$ [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 $\text{FO}^\oplus$ (or $\text{U}_m^\otimes$) with one-time pad and a message authentication code (MAC). The one with $\text{FO}^\oplus$ is the same scheme considered in [SS19], but ours is the first correct proof in the QROM. Since the proofs for $\text{FO}^\oplus$ and $\text{U}_m^\otimes$ are similar, we leave the one for $\text{U}_m^\otimes$ in Supp. Mat. H, and there we only prove the Bi-SO-CCA for $\text{U}_m^\otimes$, 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 [BHYO9, 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, $\text{FO}^\oplus$ and $\text{U}_m^\otimes$, 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 versions of transformation $\text{FO}^\oplus$, and NTRU uses $\text{U}_m^\otimes$. 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 $\text{FO}^\oplus$ 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 $\text{FO}^\oplus$ transformation, $(e, d)$, is defined as follow:

\[
e := \text{Enc}_0(\text{pk}, r, G(r)) \quad \text{for} \quad r \in \mathcal{R}, M' \quad \\
d := H(r, e) \oplus m
\]  

(1)
where $\text{Enc}_0$ is the randomized encryption algorithm of a OW-CPA secure PKE with message space $\mathcal{M}'$, $G(r)$ is the explicit randomness used in $\text{Enc}_0$, and $G,H$ are two hash functions with suitable domains and ranges. Public and secret keys of $\text{FO}^k$ 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 $(\text{Enc}_0(\text{pk},r^*_i; R^*_i), d^*_i)_{1 \leq i \leq \mu}$ to the SO-CPA adversary $A$. For these challenge ciphertexts, the reduction $\mathcal{R}$ can open a ciphertext $(\text{Enc}_0(\text{pk},r^*_i; R^*_i), d^*_i)$ 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 to one of the unopened ciphertexts, $G(r^*_i)$ 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 $S \subset X$. Let $H,H' : X \to Y$ be two different ROs such that, for all $x \in X \setminus S$, $H(x) = H'(x)$ (denoted by $H \setminus S = H' \setminus S$). The semi-classical O2H lemma states that a quantum adversary $A$ cannot tell the difference between $H$ and $H'$ by giving only quantum access to them, unless $A$ 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$_3$ 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 $A$. 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 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 $A$ 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 $A$. 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\footnote{https://eprint.iacr.org/archive/2022/617/20230108:160413} 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 $\Sigma$ 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 Hybrid$^{(1)}$ 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 yield 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 $\{1, \ldots, n\}$. Let $\mathcal{X}$ and $\mathcal{Y}$ be two finite sets and $f : \mathcal{X} \to \mathcal{Y}$ be a function. $f(\mathcal{X}) := \{f(x) | x \in \mathcal{X}\}$. $x \in之道 \mathcal{X}$ denotes sampling a uniform element $x$ from $\mathcal{X}$ at random. If $S$ is a subset of $\mathcal{X}$, then $\mathcal{X} \setminus S$ denotes the set $\{x \in \mathcal{X} | x \not\in S\}$. Let $A$ be an algorithm. If $A$ is probabilistic, then $y \leftarrow A(x)$ means that the variable $y$ is assigned to the output of $A$ on input $x$. If $A$ is deterministic, then we write $y := A(x)$. We write $A^O$ to indicate that $A$ has classical access to oracle $O$. We write $T_A(A_0) \approx T_A(A_1)$ if the running times of $A_0$ and $A_1$ 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 $\epsilon$. $Pr[G^A \Rightarrow 1]$ denotes the probability that the final output
GA of game G running an adversary A is 1. Let Ev be an (classical and well-defined) event. We write Pr[Ev : G] to denote the probability that Ev occurs during the game G.

2.1 Public-Key Encryption

A Public Key Encryption (PKE) scheme PKE consists of three algorithms (KG, Enc, Dec) and a message space 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 (pk, sk). pk also defines a finite randomness space R := R(pk) and a ciphertext space C := C(pk). For sake of simplicity, in this paper, we ignore the input λ and simply write the process as (pk, sk) \leftarrow KG.
- The encryption algorithm Enc, on input pk and a message m \in M, outputs a ciphertext c \in C. We also write c := Enc(pk, m; r) to indicate the randomness r \in R explicitly.
- The (deterministic) decryption algorithm Dec, on input sk and a ciphertext c, outputs a message m' \in M or a rejection symbol \perp.

Definition 2.1 (PKE Correctness). A PKE scheme PKE := (KG, Enc, Dec) with message space M is (1 − δ)-correct if

\[ \mathbb{E}_{m \in M}[\max \Pr[\text{Dec}(sk, c) \neq m : c \leftarrow \text{Enc}(pk, m)]] \leq \delta, \]

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

Definition 2.2 (Collision Probability of Key Generation). Let

\[ \eta_{\text{PKE}} := \max [\Pr[pk_0 = pk_1 : (pk_0, sk_0) \leftarrow KG, (pk_1, sk_1) \leftarrow KG]] \]

be the collision probability of KG of PKE. The maximum is taken over all pk_0, pk_1. In this paper, we assume that for any OW-CPA-secure PKE, \eta_{\text{PKE}} = \text{negl}(\lambda).

We focus on two security notions for PKE: one-wayness under chosen-plaintext attacks (OW-CPA) and selective-opening security under chosen-ciphertext-attacks (SO-CCA). Let PKE := (KG, Enc, Dec) be a PKE scheme with message space M and ciphertext space C.

Definition 2.3 (OW-CPA). For an adversary A, its advantage against OW-CPA security of PKE is defined as

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

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

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

Definition 2.4 (otSUF-CMA). For a forger F, its advantage against otSUF-CMA security of MAC is defined as

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

MAC is otSUF-CMA secure if for all F, \text{Adv}_{\text{PKE}}^{\text{otSUF-CMA}}(F) = \text{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
We note that a non-adaptive variant of SO security has been used in \cite{LYHW18}, 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 \( \text{OPEN} \) 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.

\textbf{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 \( \lambda \), \( \mu := \mu(\lambda) > 0 \) 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\textsubscript{PKE} and a SO simulator \( \mathcal{S} \) in ideal-SO-ATK\textsubscript{PKE}. \( \mathcal{M}_a \) is a distribution over \( \mathcal{M} \) chosen by \( \mathcal{A} \), and \( \mathcal{A} \) is not allowed to issue \( \text{OPEN} \) queries before it outputs \( \mathcal{M}_a \) and receives challenge ciphertexts \( \mathfrak{c} \). Messages sampled from \( \mathcal{M}_a \) may be dependent on each other. \( \text{Dec} \) is not available in SO-CPA security.

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

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

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

\textbf{(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 corrupt 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 \( \text{OPEN} \) queries and \( \text{CORRUPT} \) queries in any order. The enhanced definition is also simulation-based. If \( \mathcal{A} \) corrupts a
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 $\lambda$, let $\mu := \mu(\lambda)$ and $p := p(\lambda)$ that are both polynomially bounded. Let $\text{Rel}$ be a relation. We consider two games defined in Figure 3, where $\mathcal{A}$ is run in REAL-Bi-SO-ATK$_{\text{PKE}}$ and a Bi-SO simulator $S$ in IDEAL-Bi-SO-ATK$_{\text{PKE}}$. $\mathcal{M}_a$ 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 $c$. Messages sampled from $\mathcal{M}_a$ may be dependent on each other. $\text{Dec}$ is not available in Bi-SO-CPA security.

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

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

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

**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 $\lambda$-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 $\lambda$-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, $x' \leftarrow \text{Measure}(|x\rangle)$ 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 $\mathcal{O} : \mathcal{X} \rightarrow \mathcal{Y}$ 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 $U_\mathcal{O} : |x\rangle |y\rangle \rightarrow |x\rangle |y \oplus \mathcal{O}(x)\rangle$, and the adversary can query random oracles on quantum states. For an quantum adversary $\mathcal{A}$, the notation $\mathcal{A}^{(\mathcal{O})}$ 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 $h : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Y}$ and $h' : \mathcal{X} \rightarrow \mathcal{Y}$ be two independent random oracles. If an unbounded time quantum adversary $\mathcal{A}$ that queries $h$ at most $q_h$ times, then we have

$$\Pr[1 \leftarrow \mathcal{A}^{(\mathcal{O})}(h(x, \cdot))(s \in \mathcal{X})] - \Pr[1 \leftarrow \mathcal{A}^{(\mathcal{O})}(h'(\cdot))(1)] \leq 2q_h/\sqrt{|\mathcal{X}|}$$

### 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 $\mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y}$ and takes $i_{n_0}$ as input and terminates by outputting $\text{out}_{n_1}$. During its execution, $\mathcal{A}$ outputs some $\text{out}_1$, and then takes $i_{n+1}$ as input ($0 \leq i \leq n - 1$). We view $\mathcal{A}$ as a $(n + 1)$-stage adversary, $(A_0, A_1, ..., A_n)$, where $A_i$ takes $i_{n_0}$ as input and outputs $\text{out}_i$. Here $i_{n_0}, \text{out}_0, i_{n_1}, ..., i_{n_n},$ and $\text{out}_n$ can be arbitrary classical information. In this paper, we consider post-quantum setting where adversaries have quantum access to hash functions. The classical information $i_{n_0}, \text{out}_0, i_{n_1}, ..., i_{n_n}, \text{out}_n$ 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 $\mathcal{A} = (A_0, ..., A_n)$ to divide $\mathcal{A}$ into $n + 1$ stages for better analysis. By writing $\text{out}_i \leftarrow A_i(i_{n_i})$ we mean that at stage $i$, $\mathcal{A}$ receives input $i_{n_i}$ and outputs $\text{out}_i$ at the end of the stage. The index indicates the stage number of $\mathcal{A}$. So, all $A_i$ 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 $H_1$. In NONADA, $H$ will never get reprogrammed, while in ADA different stages of $\mathcal{A}$ will have access to different $\mathcal{H}_i$. That is, $A_i$ queries $H_i$, and according to $A_i$’s output $\text{out}_i$, $H_i$ will be reprogrammed and become $H_{i+1}$ (cf. Items 07, 17 and 18). To formalize this, we define three algorithms INIT, $F_s$, and $\text{Repro}_s$ in Figure 4 as:

- The initialization algorithm INIT outputs $((s, i_{n_0}), H, H_0)$ (cf. Items 01 and 11), where $s$ is some parameter that used in a security reduction, $i_{n_0}$ is the initial input to $A$, and $H$ and $H_0$ are two random oracles. Here the tuple $((s, i_{n_0}), H, H_0)$ may have an arbitrary joint distribution.

- $F_s$ takes $\text{out}_i$ as input and computes $(i_{n+1}, i_{n+1}')$, where $i_{n+1}$ is the input to $A_{i+1}$ and $i_{n+1}'$ is the information for reprogramming $H_i$. Here $i_{n+1}'$ is used to capture the fact that $H$ can be reprogrammed according to $A_i$’s behavior, and the algorithm $\text{Repro}_s$ (described below) will take it as input. To make our lemma general and useful for a wider class of applications, we only require that $F_s$ does not have access to random oracles.

- $\text{Repro}_s$ is defined to reprogram $H$ in ADA (cf. Item 17). $\text{Repro}_s$ takes $i_{n}'$ 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 $\text{Repro}_s$ depends on the concrete use of our framework. Here we only require $\text{Repro}_s$ to be deterministic.

Let $S_i$ be a set such that $H \setminus S_i = H_i \setminus S_i$ (namely, for all $x \in \mathcal{X}$, if $x \in S_i$, then $H(x) \neq H_i(x)$). $\mathcal{A}$ 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
described earlier), there exists adversaries 
access to random oracle 
$H$ and $Pr$

most
raised by $A$

Figure 4: Games $\text{NonADA}^A$ and $\text{Ada}^A$ used in Lemma 3.1. The main difference between two games is highlighted with gray box. In both games, $A$ is divided into $n + 1$ stages, namely, $(A_0, ..., A_n)$. The input and output of $A$ in each stage are classical information because we consider post-quantum settings. The list $\Gamma$ stores $A$'s outputs in each stage. $F_s$ is a deterministic algorithm that provides inputs for each stage of $A$. $\text{Repro}_s$ is a deterministic algorithm that reprograms QROs. For a concise presentation, we assume that $A_i$ takes $A_{i-1}$'s final state as its initial state. In our framework, $H_0$ can be different to $H$.

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

**Lemma 3.1** Let $A$ be an adversary that can be divided into $(n+1)$ stages as in Figure 4 and has quantum access to random oracle $H$ (\(\equiv H \) in $\text{NonADA}$ or $H_i$ in $\text{Ada}$). Let $Ev$ be a classical event that may be raised by $A$ in $\text{NonADA}$ or $\text{Ada}$. Suppose that $A$ queries $H$ 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 $\text{INIT}$, $F_s$, and $\text{Repro}_s$ (as described earlier), there exists adversaries $B_i$ for $i \in \{0, ..., n\}$ (shown in Figure 5) such that

$$
\left| \Pr[Ev : \text{NonADA}^A] - \Pr[Ev : \text{Ada}^A] \right| \\
\leq \sum_{i=0}^{n} \sum_{k=0}^{q_i} 2q_i \Pr[\exists x' \in S_i : \text{Ada}^{B_i}] 
$$

where $S_i$ is a set such that $H_0 \setminus S_i = H_i \setminus S_i$. Such an $S_i$ is defined by the operations in $\text{Repro}_s$. $\Pr[Ev : \text{NonADA}^A]$ and $\Pr[Ev : \text{Ada}^A]$ are the probabilities that $A$ triggers $Ev$ in $\text{NonADA}$ and in $\text{Ada}$, respectively.

**Discussions on Lemma 3.1.** In $\text{Ada}$, reprogramming the RO is captured by algorithm $\text{Repro}_s$. 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 $\text{NonADA}$ and $\text{Ada}$ is that between $H$ and $H_i$ caused by $\text{Repro}_s$. 

Figure 5: Algorithm $B_i$ (used in Lemma 3.1) plays Game $\text{Ada}$ (where $i \in [n]$). $B_i$ proceeds identically with $(A_1, ..., A_n)$, except that $B_i$ measures the $t^*$-th QRO query issued by $A_i$ and then outputs the measurement outcome.
Concretely, in $i$-th stage, $\text{Repro}_s$ will define a set $S_i$ such that $H \setminus S_i = H_i \setminus S_i$. For any $k \in \{0, \ldots, n\}$, if $A$ queries $H$ with an $x \in \cup_{i \leq k} S_k$ before the end of its $k$-th stage, then $A$ can distinguish $\text{NonAda}$ and $\text{Ada}$. To bound this in the quantum setting, our approach is to randomly measure $A$’s queries to $H$, which is captured by $B_i$ (in Figure 5). The advantage of $A$ distinguishing $\text{NonAda}$ and $\text{Ada}$ is bounded by the probability that $B_i$’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 $F_s$ and $\text{Repro}_s$. However, when we use this framework to construct (efficient) reduction, $F_s$ and $\text{Repro}_s$ 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 $F_s$ and $\text{Repro}_s$, we can describe Grilo et al.’s framework using our framework. In Grilo et al.’s framework [GHHM21], the $i$-th output of $A$ is a distribution $\text{out}_i := p_i$. $F_s$ 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 $(\text{in}_{i+1} := (x_i, x'_i), \text{in}'_{i+1} := (x_i, x'_i, y_i))$.

$\text{Repro}_s$ can be defined as, on input $\text{in}_{i+1} := (x_i, x'_i, y_i)$, it reprograms the QRO $H := H[(x_i, x'_i) \rightarrow y_i]$ and returns the reprogrammed QRO. Their framework implicitly requires that the probability bound for $A$ 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 $A$ is a PPT adversary, our framework provides efficient extractors $B_i$’s to bound the difference of $A$ in $\text{NonAda}$ and $\text{Ada}$. In our proofs, we need to instantiate $\text{INIT}, F_s$, and $\text{Repro}_s$ efficiently. This $B_i$ 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 $\text{NonAda}$ and $\text{Ada}$ 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 $B_i$ (i.e., $B_i$ 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 $\text{PKE} := (\text{KG}, \text{Enc}, \text{Dec})$ be a $(1 - \delta)$-correct PKE scheme with message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, and randomness space $\mathcal{R}$. Let $G : \mathcal{M} \rightarrow \mathcal{R}$ be a hash function.

### 4.1 Selective Opening Security against Chosen-Plaintext Attacks

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

---

3The randomness for sampling can be included in $s$, since it is captured by the game simulator.
and $G$ to finish the proof. To fit into the syntax of our framework, we combine $g$s are used to respond queries to $\text{Adv}_q$. $\text{Adv}_{\text{PKE}}(\lambda)$, respectively. $\text{Adv}_{\text{PKE}}(\lambda)$, respectively.

<table>
<thead>
<tr>
<th>Game $G_0,G_3$</th>
<th>OPEN(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 (pk,sk) \leftarrow KG_0$</td>
<td>$17 I := I \cup {i}$</td>
</tr>
<tr>
<td>$2 M_a \leftarrow A^{(G \times H)}(pk)$</td>
<td>$18 \text{return } (m_1,r_i)$</td>
</tr>
<tr>
<td>$3$ for $i \in [\lambda]$</td>
<td>$19$ if $\exists i \in I \text{ s.t. } (r, e) = (r_i,e_i)$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$4 m_i := m_i \leftarrow M_a, r_i \leftarrow \mathcal{R}$</td>
<td>$20$ return $K_i$ // $G_2$</td>
</tr>
<tr>
<td>$5 R_i := G(r_i)$</td>
<td>$21$ return $d_i \oplus m_i$ // $G_3$</td>
</tr>
<tr>
<td>$6 R_i \leftarrow \mathcal{R}'$ // $G_2$-G_3</td>
<td>$22$ return $h(r,e)$</td>
</tr>
<tr>
<td>$7$ $e_i := \text{Enc}_0(pk,r_i;R_i)$</td>
<td>$23$ if $\exists i \in I \text{ s.t. } r = r_i$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$8 K_i := H(r_i,e_i)$</td>
<td>$24$ return $R_i$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$9 K_i \leftarrow \mathcal{M}$ // $G_2$</td>
<td>$25$ return $g(r)$</td>
</tr>
<tr>
<td>$10 d_i := K_i \oplus m_i$ // $G_1$-G_2</td>
<td>$26$ abort</td>
</tr>
<tr>
<td>$11 c[i] := (e_i,d_i)$</td>
<td>$27 \text{return } I_{\text{out}}$</td>
</tr>
<tr>
<td>$12$ if $\exists i \neq j \text{ s.t. } K_i = K_j$</td>
<td>$28 \text{return } I_{\text{out}}$</td>
</tr>
<tr>
<td>$13$ abort // $G_1$-G_2</td>
<td>$29$ abort</td>
</tr>
<tr>
<td>$14$ out $\leftarrow A^{\text{OPEN},(G \times H)}(c)$</td>
<td>$30$ abort</td>
</tr>
<tr>
<td>$15$ return $\text{Rel}(M_a,m,I_{\text{out}}$</td>
<td>$31$ abort</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Game $G_0$</th>
<th>OPEN(i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1 (pk,sk) \leftarrow KG_0$</td>
<td>$17 I := I \cup {i}$</td>
</tr>
<tr>
<td>$2 M_a \leftarrow A^{(G \times H)}(pk)$</td>
<td>$18 \text{return } (m_1,r_i)$</td>
</tr>
<tr>
<td>$3$ for $i \in [\lambda]$</td>
<td>$19$ if $\exists i \in I \text{ s.t. } (r, e) = (r_i,e_i)$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$4 m_i := m_i \leftarrow M_a, r_i \leftarrow \mathcal{R}$</td>
<td>$20$ return $K_i$ // $G_2$</td>
</tr>
<tr>
<td>$5 R_i := G(r_i)$</td>
<td>$21$ return $d_i \oplus m_i$ // $G_3$</td>
</tr>
<tr>
<td>$6 R_i \leftarrow \mathcal{R}'$ // $G_2$-G_3</td>
<td>$22$ return $h(r,e)$</td>
</tr>
<tr>
<td>$7$ $e_i := \text{Enc}_0(pk,r_i;R_i)$</td>
<td>$23$ if $\exists i \in I \text{ s.t. } r = r_i$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$8 K_i := H(r_i,e_i)$</td>
<td>$24$ return $R_i$ // $G_2$-G_3</td>
</tr>
<tr>
<td>$9$ $K_i \leftarrow \mathcal{M}$ // $G_2$</td>
<td>$25$ return $g(r)$</td>
</tr>
<tr>
<td>$10$ $d_i := K_i \oplus m_i$ // $G_1$-G_2</td>
<td>$26$ abort</td>
</tr>
<tr>
<td>$11$ $c[i] := (e_i,d_i)$</td>
<td>$27$ abort</td>
</tr>
<tr>
<td>$12$ if $\exists i \neq j \text{ s.t. } K_i = K_j$</td>
<td>$28$ abort</td>
</tr>
<tr>
<td>$13$ abort // $G_1$-G_2</td>
<td>$29$ abort</td>
</tr>
<tr>
<td>$14$ out $\leftarrow A^{\text{OPEN},(G \times H)}(c)$</td>
<td>$30$ abort</td>
</tr>
<tr>
<td>$15$ return $\text{Rel}(M_a,m,I_{\text{out}}$</td>
<td>$31$ abort</td>
</tr>
</tbody>
</table>

Figure 7: Games $G_0,G_3$ 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 $\lambda$ and $\mu := \mu(\lambda)$ (polynomially bounded), for any SO-CPA adversary $A$ and relation $\text{Rel}$, there exist a simulator $S$ and an adversary $B^{ow}$ such that $T(S) \approx T(A) \approx T(B^{ow})$ and

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

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

**Proof.** Let $h : \mathcal{R}' \times \mathcal{C}' \rightarrow \mathcal{M}$ and $g : \mathcal{M}' \rightarrow \mathcal{R}'$ 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 $h : \mathcal{R}' \times \mathcal{C}' \rightarrow \mathcal{M}$ and $g : \mathcal{M}' \rightarrow \mathcal{R}'$, respectively.

Our proof consists of a sequence of games defined in Figure 7. We 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 $\mathcal{X} \times H$ such that $\mathcal{X} \times H(h',r,e) := (G(h'),H(r,e))$. If $A$ only queries $G(h')$, we view it as querying $\mathcal{X} \times H(h',r,e)$ for some dummy $(r,e)$ and ignoring $H(r,e)$ in the response. $A$ can query $\mathcal{X} \times H$ at most $q = q_H + q_G$ times. This was also used in [JZC+18]. $G_0$ is equivalent to $\text{REAL-SO-CPA}_{wPKE}$, thus

$$\Pr[\text{REAL-SO-CPA}_{wPKE}^A \Rightarrow 1] = \Pr[G_0^A \Rightarrow 1]$$

**Game $G_1$:** 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

$$|\Pr[G_0^A \Rightarrow 1] - \Pr[G_1^A \Rightarrow 1]| \leq \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|}$$
we define
\( g, g' \in H \)
\( \Gamma[0] := \text{return } \)
\( \text{for } i = 1 \text{ to } n_{\text{op}}: \)
\( (m_i, \text{in}_i) := F_0(\text{out}_{i-1}) \)
\( H_i := \text{Repro}_s(\text{in}_i, \text{H}_{i-1}) \)
\( H_i := H_i \)
\( \text{out}_i := A_i[H_i](\text{in}_i) \)
\( \Gamma[i] := \text{out}_i \)
\( \text{return } \text{Rel}(M_a, m, I, \Gamma[n_{\text{op}}]) \)

**INIT**
\( I := \emptyset \)
\( (pk, sk) \leftarrow K_G \)
\( M_a \leftarrow A_0^{(p \times k)}(pk) \)
\( \text{Let } g' \text{ and } h' \text{ be internal QROs.} \)
\( \text{for } i \in [\mu]: \)
\( m[i] := m_i \leftarrow M_a, r[i] := r_i \rightleftharpoons M' \)
\( R_i := g(r_i), R[i] := R_i \)
\( e_i := \text{Enc}_0(pk, r_i; R_i) \)
\( K_i := h(r_i, e_i), d_i := K_i \oplus m_i \)
\( c[i] := (e_i, d_i) \)
\( s := (M_a, m, r, R, c), \text{in}_0 := c \)
\( \text{Let } G_0 \times H_0 \text{ be a QRO such that } G_0 \times H_0(x) := \begin{cases} g \times h(x), & (x \notin S_0) \\ g' \times h'(x), & (\text{else}) \end{cases} \)
\( \text{return } ((s, \text{in}_0), (G \times H), (G_0 \times H_0)) \)

**Game \( G_1 \)**: \( R_i \) and \( K_i \) in the challenge ciphertexts are chosen randomly, instead of using \( G \) and \( H \). If \( A \) queries 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 \( G_1 \) and \( G_2 \). In \( G_2 \), \( A \)'s OPEN queries will make QRO \( G \times H \) reprogrammed, while in \( G_1 \), QRO \( G \times H \) does not get reprogrammed. So, we can view \( G_1 \) and \( G_2 \) as concrete cases of NONADA and ADA, respectively. For simplicity, we denote \( A := (A_0, (A_{1,0}, ..., A_{1,n_{\text{op}}})) \), where \( A_0 \) is the initial stage of \( A \) and cannot query OPEN, and \( (A_{1,0}, ..., A_{1,n_{\text{op}}}) \) is the stage that \( A \) receives the challenge ciphertexts \( c \) and can query OPEN. Let \( A_1 := (A_{1,0}, ..., A_{1,n_{\text{op}}}) \). \( A_1 \)'s initial state is the final state of \( A_0 \). \( A_{1,k} \) is defined with respect to OPEN queries.

- Before any OPEN query (i.e., at the 0-th stage), \( A_{1,0} \) takes \( \text{in}_0 := c \) as input and outputs the first opening index \( \text{out}_0 := (i_1) \).
- At \( k \)-th stage \((1 \leq k \leq n_{\text{op}} - 1)\), \( A_{1,k} \) receives \( \text{in}_k = (m_{ik}, r_{ik}) \) as the result of the \((k-1)\)-th OPEN query and finishes the stage by outputting the \((k+1)\)-th opening index \( \text{out}_k := (i_{k+1}) \).
- Finally, at the \( n_{\text{op}} \) stage, \( A_{1,n_{\text{op}}} \) receives \( \text{in}_{n_{\text{op}}} = (m_{in_{\text{op}}}, r_{in_{\text{op}}}) \) and terminates by outputting \( \text{out}_{n_{\text{op}}} = \text{out} \) (the final output of SO adversary).

To formally show why \( G_1 \) and \( G_2 \) are concrete cases of NONADA and ADA, respectively, in Figure 8, we define \( \text{INIT}, F_s, \text{Repro}_s, G'_1 \) and \( G'_2 \). Games \( G'_1 \) and \( G'_2 \) 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 $G_1$ and $G_2$ are concrete cases of NonAda and Ada, respectively. Namely, $G_1$ and $G_2$ in Figure 7 are equivalent to $G'_1$ and $G'_2$ in Figure 8, respectively. Firstly, algorithm INIT in Figure 8 run the codes from Item 01 to Item 12 in Figure 7. Since in $A_0$’s view, $G_1$ is the same as $G_2$ (it does not see any challenge ciphertexts), the distribution of $M_a$ and $m$ in $G_1$ is the same as the one in $G_2$, and thus the output of INIT and the final state of $A_0$ in INIT in $G'_1$ are the same as those in $G'_2$. Secondly, $F_x$ simulates the OPEN oracle and $Repro$ simulates the reprogramming operations on $G$ and $H$. In $G'_1$, $G$ and $H$ will not be reprogrammed, but in $G'_2$, $G$ and $H$ will be reprogrammed, according to $A$’s output. This is the same as in $G_2$.

Moreover, when running $A_{1,k}$, our $Repro$ defines a set

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

where $I_k := \{i_1, ..., i_k\}$ is the opening index set $I$ in $A_1$’s $k$-th stage. Answers of $G \times H$ on $S_k$ are only different in $G_1$ (i.e., NonAda) and $G_2$ (i.e., Ada). For $k = 0$, $S_0$ is defined at line 35 and $I_0 = \emptyset$.

Now we consider the probability that $\text{Rel}(M_a, m, I, out) = 1$. $I$ and $out$ are determined by $A_1$. $M_a$ is output by $A_0$, and $m$ is determined by $M_a$. Since in $A_0$’s view, $G_1$ is the same as $G_2$ (since it does not see challenge ciphertexts), thus the distribution of $M_a$ and $m$ in $G_1$ is the same as the one in $G_2$. Therefore, the probability difference between the classical event that $\text{Rel}(M_a, m, I, out) = 1$ in $G_1$ and the similar event in $G_2$, is determined by the probability difference between the event that $A_1$ outputs a particular $(I, out)$ (i.e., $\Gamma$ in Figure 8) in $G_1$ and the similar event in $G_2$. Therefore, we have

$$|Pr[G_1^A \Rightarrow 1] - Pr[G_2^A \Rightarrow 1]| \leq |Pr[G_1'^{A_1} \Rightarrow 1] - Pr[G_2'^{A_1} \Rightarrow 1]| + \frac{2\mu q}{\sqrt{|R|}}$$

This bound includes a term $\frac{2\mu q}{\sqrt{|R|}}$, since $A_0$ also has quantum access to $|G \times H|$, and this term is the probability that the first stage (i.e., $A_{1,0}$) of $A_1$ learns $r_i$ before seeing challenge ciphertexts. Such probability is only information-theoretic.

We now use Lemma 3.1 to bound Equation (4). Since $G'_1$ is a NonAda game and $G'_2$ is an Ada game, by Lemma 3.1, there exist adversaries $B_i$ $(0 \leq i \leq n_{\text{Op}})$, which take $in_0 = c$ as its input and output $x \in S_k$ where the set $S_i$ is defined in (3), such that

$$|Pr[G'_1^{A_1} \Rightarrow 1] - Pr[G'_2^{A_1} \Rightarrow 1]| \leq \sum_{k=0}^{n_{\text{Op}}} \sum_{i=0}^{k} 2q_i \sqrt{Pr[x \leftarrow B_i \text{ s.t. } x \in S_i : G_2^{R_i}]}
$$

Here $B_i$ proceeds the same as $(A_{1,0}, ..., A_{1,i})$ except that it randomly measures a QRO query issued by $A_{1,i}$. Moreover, since $A_{1,0}$’s initial state is the final state of $A_0$, $B_i$ starts with state of $A_0$ (cf. Item 07).

Based on $B_i$, we construct an adversary $B_{i}^{\text{ow}}$ (in Figure 9) to breaks OW-CPA security of PKE. By definition of $B_{i}^{\text{ow}}$, if $A_1$ does not open $t^*$, and $r$ or $r'$ equals the solution of $e^*$, then $B_{i}^{\text{ow}}$ wins. So the winning probability for $B_{i}^{\text{ow}}$ to breaks the OW-CPA challenge is:

$$\text{Adv}_{\text{ow}}^{\text{OW-CPA}}(B_{i}^{\text{ow}}) = \frac{1}{2} \frac{\mu - n_{\text{Op}}}{\mu} \frac{1}{\mu - n_{\text{Op}}} \text{Pr}[x \leftarrow B_i \text{ s.t. } x \in S_i],$$

and thus we have

$$\text{Pr}[x \leftarrow B_i \text{ s.t. } x \in S_i : G_2^{R_i}] \leq 2\mu \text{Adv}_{\text{ow}}^{\text{OW-CPA}}(B_{i}^{\text{ow}})$$

Let $B_{i}^{\text{ow}}$ be the adversary that has highest advantage against PKE among $\{B_{i}^{\text{ow}}\}_{i \in [0, ..., n_{\text{op}}]}$. Then equation (6) can be written as:

$$\text{Pr}[x \leftarrow B_i \text{ s.t. } x \in S_i : G_2^{R_i}] \leq 2\mu \text{Adv}_{\text{ow}}^{\text{OW-CPA}}(B_{i}^{\text{ow}}), \text{ for } \forall i \in [\mu]$$

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

$$|Pr[G_1^A \Rightarrow 1] - Pr[G_2^A \Rightarrow 1]| \leq 2(n_{\text{Op}} + 1)^2 q \sqrt{2\mu \text{Adv}_{\text{ow}}^{\text{OW-CPA}}(B_{i}^{\text{ow}})} + \frac{2\mu q}{\sqrt{|R|}}$$
$B_i^{ow}(pk^*, e^*)$ is a OW-CPA challenge of PKE

01 $I := \emptyset$
02 $((s, in_0), (G \times H), (G_0 \times H_0)) \leftarrow \text{INIT}$ $\text{ // INIT is defined in Figure 8 and it uses pk^* instead of KG_0}$
03 parse $(\mathcal{M}_a, m, r, R, c) := s$
04 parse $c := in_0$
05 $t^* \leftarrow [\mu], (e_{t^*}, d_{t^*}) := c[t^*]$
06 $c[t^*] := (m, e_{t^*}, d_{t^*}), in_0 := c$
07 Initialize $B_i$ with $\mathcal{A}_0$’s final state in INIT.
08 if $i = 0$: goto line 18
09 $\text{out}_0 \leftarrow B_i^{G_0 \times H_0} (in_0)$
10 if $\text{out}_0 = t^*$: abort
11 $(in_1, in_1') := F_s(\text{out}_0)$ $\text{ // F_s is defined in Figure 8}$
12 $(G_1 \times H_1) := \text{Repro}_a (in_1', (G_0 \times H_0))$ $\text{ // Repro}_a is defined in Figure 8$
13 for $j = 1$ to $i - 1$:
14 $(in_{j+1}, in_{j+1}') := F_s(\text{out}_j)$
15 $(G_{j+1} \times H_{j+1}) := \text{Repro}_a (in_{j+1}', (G_j \times H_j))$
16 $(r_{j'}, (r_{j'}, e')) \leftarrow B_i^{G_j \times H_j} (in_j)$
17 $b \leftarrow \{0, 1\}$
18 $r^* := r_{b}'$
19 return $r^*$

Figure 9: The constructions of OW-CPA adversaries $B_i^{ow}$ for $i \in \{0, \ldots, n_{\text{OR}}\}$. $B_i^{ow}$ simulates $G_2$ (which is a concrete case of $\text{Ada}$ in Figure 4) for $B_i$ to break PKE. $F$ and $\text{Repro}_a$ are defined as in Figure 8.

**Game $G_3$:** 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 $G_2$, all $K_i$ are independently and uniformly random. In $G_1$, we excluded any collision of $K_i$, so, in $G_3$, it is equivalent to sample $d_i$ in a collision-free way. Therefore, we have

$$\Pr[G_3^A \Rightarrow 1] = \Pr[G_3^A \Rightarrow 1]$$

**Construction of SO simulator $S$** We construct a SO simulator $S$ that is simulating $G_3$ for $\mathcal{A}$ and interacts with the IDEAL-SO-CPA$_{\text{wpKE}}^S$ game. The simulation process is shown in Figure 10. Obviously, $S$ can perfectly simulates $G_3$. So, we have

$$\Pr[G_3^A \Rightarrow 1] = \Pr[\text{IDEAL-SO-CPA}^S_{\text{wpKE}} \Rightarrow 1]$$

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

$$\left| \Pr[\text{REAL-SO-CPA}^A_{\text{wpKE}} \Rightarrow 1] - \Pr[\text{IDEAL-SO-CPA}^S_{\text{wpKE}} \Rightarrow 1] \right| \leq 2(n_{\text{OR}} + 1)^2 \sqrt{2\mu} \text{Adv}_{\text{PKE}}^\text{OW-CPA} (B^{ow}) + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{M}'|} + \frac{2\muq}{\sqrt{|R|}}$$

\[
\square
\]

**4.2 Selective Opening Security against Chosen-Ciphertext Attacks**

Let $\text{MAC} = (\text{Tag}, \text{Vrfy})$ be a MAC scheme with key space $\mathcal{K}_{\text{MAC}}$, and let $H : \mathcal{C} \times \mathcal{C}' \to \mathcal{M} \times \mathcal{K}_{\text{MAC}}$ be a hash function, where $\mathcal{C}$ is the ciphertext space of $\text{PKE}$. The second PKE scheme $\text{PKE'} = (\text{sKG}, \text{sEnc}, \text{sDec})$ (Figure 11) is a combination of a modular Fujisaki-Okamoto’s transformation $\text{FO}^\mu$ [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
S

0.1 Chooses QR0s g, h at random
0.2 I := \emptyset
0.3 (pk, sk) \leftarrow \text{wKG}
0.4 M_a \leftarrow \mathcal{A}(pk)
0.5 Outputs M_a and receives m''
0.6 for i \in [\mu]
0.7 r_i \notin \mathcal{M}', \mathcal{R}, \mathcal{R}'
0.8 e_i := \text{Enc}_0(pk, r_i; R_i)
0.9 d_i \notin \mathcal{M}\{-d_1, ..., d_{i-1}\}
10 c[i] := (e_i, d_i)
11 out \leftarrow A^{\text{PSPE,$OG \times H$}}(c)
12 return out

\text{OPEN(i)}
13 I := I \cup \{i\}
14 Queries \text{OPEN}' on i and receives m_i
15 return \(m_i, r_i\)
16 if \exists i \in I \text{ s.t. } (r, e) = (r_i, e_i)
17 return \(d_i \oplus m_i\)
18 return \(h(r, e)\)
19 G(r)
20 if \exists i \in I \text{ s.t. } r = r_i
21 return \(R_i\)
22 return \(g(r)\)

Figure 10: The simulator S of the proof of Theorem 4.1.

\begin{tabular}{|c|c|c|}
\hline
\text{SKG} & \text{sEnc}(pk, m \in \mathcal{M}) & \text{sDec}(sk, k, (e, d, \tau)) \\
\hline
01 (pk, sk) \leftarrow \text{KG}_0 & 06 r \notin \mathcal{M}' & 12 r' := \text{Dec}_0(sk, e) \\
02 k \notin \mathcal{M}' & 07 e := \text{Enc}_0(pk, r; G(r)) & 13 if \ r' = \bot \\
03 pk' := pk & 08 (K, K^\text{mac'}) := H(r, e) & 14 or e \neq \text{Enc}_0(pk, r'; G(r')) \\
04 sk' := (sk, k) & 09 d := K \oplus m & 15 (K, K^\text{mac'}) := H(k, e) \\
05 return (pk', sk') & 10 \tau := \text{Tag}(K^\text{mac'}, d) & 16 else (K, K^\text{mac'}) := H(r', e) \\
& 11 return (e, d, \tau) & 17 if \ \text{Vrfy}(K^\text{mac'}, \tau) = 1 \\
& & 18 m := K \oplus d \\
& & 19 else \ m := \bot \\
& & 20 return m \\
\hline
\end{tabular}

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

(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 ciphertext. The proof of Theorem 4.2 is given in Supp. Mat. F.

Theorem 4.2 If PKE is OW-CPA secure and \(\delta\)-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 \(\lambda\) and integer \(\mu := \mu(\lambda)\) (polynomially bounded) for any SO-CCA adversary \(A\) and relation \(R\), there exist a simulator \(S\) and adversaries \(B^\text{ow}\) and \(F\) such that \(T(S) \approx T(A) \approx T(B^\text{ow}) \approx T(F)\) and

\[
\text{Adv}_{\text{PSPE}}^{\text{SO-CCA}}(A, S, \mu, R) \leq 3\mu\text{Adv}_{\text{MAC}}^{\text{otSUF-CMA}}(F) + 6(\nu_{\text{Or}} + 1)\sqrt{2\mu\text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(B^\text{ow}) + \mu\text{Adv}_{\text{MAC}}^{\text{otSUF-CMA}}(F)} + \frac{2q_h}{\sqrt{\mu}} + 16(\mu + n_{\text{Dec}} + q + 1)^2\delta + \frac{\mu^2}{|\mathcal{M}|} + \frac{\mu^2}{|\mathcal{R}'|} + \frac{6\mu q}{|\mathcal{C}'|} + \frac{\mu n_{\text{Dec}}}{|\mathcal{M}'|} + \frac{(2 + \mu)q}{\sqrt{|\mathcal{M}'|}}
\]

where \(\mu, q, q_h, n_{\text{Or}},\) and \(n_{\text{Dec}}\) are the maximum numbers of \(A\)'s challenge ciphertexts, \(A\)'s queries to \(G, H, \text{OPEN, and DEC}\), 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 \(\text{PKE}_1 := (\text{KG}_1, \text{Enc}_1, \text{Dec}_1)\) be a PKE scheme with message space \(\mathcal{M}'\) and randomness space \(\mathcal{R}'\). \(\text{PKE}_1\) is lossy if it has the following properties:
• PKE₁ is correct according to Definition 2.1.

• Key indistinguishability: We say PKE₁ has key indistinguishability if there is an algorithm LKG₁ such that, for any adversary B, the advantage function

\[
\text{Adv}_{\text{PKE₁}}^{\text{ind-key}}(B) := |\Pr[B(pk₁) \Rightarrow 1] - \Pr[B(lpk₁) \Rightarrow 1]|
\]

is negligible, where \((pk₁, sk₁) \leftarrow \text{LKG₁}\) and \((lpk₁, lsk₁) \leftarrow \text{LKG₁}\).

• Lossiness: Let \((lpk₁, lsk₁) \leftarrow \text{LKG₁}\) and \(m, m'\) be arbitrary messages in \(\mathcal{M'}\), the statistical distance between \(\text{Enc}_1(lpk₁, m)\) and \(\text{Enc}_1(lpk₁, m')\) is negligible.

• Weak Openability: Let \((lpk₁, lsk₁) \leftarrow \text{LKG₁}\) and \(m, m'\) be arbitrary messages, and \(r\) be arbitrary randomness. For ciphertext \(c := \text{Enc}_1(lpk₁, m; r)\), there exists an algorithm \(\text{open}_1\) such that \(\text{open}_1(lsk₁, lpk₁, c, r, m')\) outputs \(r'\) where \(c = \text{Enc}_1(lpk₁, m'; r')\) and \(r'\) is distributed uniformly. \(\text{open}_1\) 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 \((lpk₁, lsk₁) \leftarrow \text{LKG₁}\), \(\mu\) be the number of challenge, and \(m₁, m'_1, ..., mₙ, m'_n\) be arbitrary messages in \(\mathcal{M'}\). Multi-challenge Lossiness requires that statistical distance between \(\{\text{Enc}_1(lpk₁, m_i)\}_{i \in [\mu]}\) and \(\{\text{Enc}_1(lpk₁, m'_i)\}_{i \in [\mu]}\) is negligible. We write the distance as \(\mu\)-ind-enc \(\text{PKE₁}\).

### 5.1 Construction

Let PKE₁ = (KG₁, Enc₁, Dec₁) be a lossy encryption with message space \(\mathcal{M'}\), randomness space \(\mathcal{R'}\), ciphertext space \(\mathcal{C'}\), and an opening algorithm \(\text{open}_1\). Let MAC = (Tag, Vrfy) be a MAC scheme with key space \(\mathcal{K}^{\text{mac}}\), and \(G : \mathcal{M'} \rightarrow \mathcal{R'}, H : \mathcal{R'} \times \mathcal{C'} \rightarrow \mathcal{M} \times \mathcal{K}^{\text{mac}}\) be two hash functions. Our PKE scheme sPKE = (sKG, sEnc, sDec) is defined in Figure 12, which has the same structure with the scheme in Figure 11.

![Figure 12](image-url)

Figure 12: A PKE scheme sPKE = (sKG, sEnc, sDec) based on lossy encryption PKE₁.

**Theorem 5.3** shows that sPKE is tightly SO-CCA secure when modeling \(G\) and \(H\) as QROs. Although there is a loss \(\mu\) 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.
there exist a simulator $S$ where we show that a multi-user version of $sPKE$ Theorem 6.1 reprograms our framework, we divide $A$ Dec use the encrypt-then-hash technique to simulate Corrupt oracle. But the proof idea is similar: we change the games so that the game simulator can change 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 $FO^L$ [JZC+18, HHK17] (Section 6.1). The second scheme is based on another modular FO transformation $U_m^L$ [HHK17] (Section 6.2).

### 6.1 Bi-SO Security of $FO^L$

We show that a multi-user version of $sPKE$ (Figure 11) is Bi-SO-CCA-secure in the QROM. Using the same building blocks $PKE = (KG_0, Enc_0, Dec_0)$ and MAC as $sPKE$, we propose $sPKE_{bi}$ (Figure 13). This scheme can be viewed as a combination of a modular FO transformation $FO^L$ [PKE, G, H] in [JZC+18, HHK17], one-time pad, and the a MAC scheme MAC. Moreover, in $sPKE_{bi}$, each user includes its public key as an input to the hash functions $G, H, H'$.

Theorem 6.1 shows that $sPKE_{bi}$ 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 $A_1$ 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 $sPKE_{bi}$ in Figure 13 is adaptive Bi-SO-CCA secure (Definition 2.6). Concretely, for any Bi-SO-CCA adversary $A$ and relation $Rel$, there exist a simulator $S$ and adversaries $B^{ow}$ and $F$ such that $T(S) \equiv T(A) \equiv T(B^{ow}) \equiv T(F)$ and

\[
\text{Adv}_{\text{Bi-SO-CCA}}^{sPKE_{bi}}(A, S, p, \mu, Rel) \leq 6(n_{CO} + n_{OP} + 1)^2 q \sqrt{\frac{2p\mu\text{Adv}_{\text{OW-CPA}}^{PKE}(B^{ow}) + \mu\text{Adv}_{\text{MAC}}^{\text{CMA}}(F) + p \muKG_{in}}{|C'| - n_{DEC}}} + 3\mu\text{Adv}_{\text{MAC}}^{\text{CMA}}(F) + \frac{p \mu_{DEC}}{|C'| - n_{DEC}} + \frac{p^2 \mu^2}{|C'|} + \frac{p^2 \mu^2}{|C'|} + \frac{p^2 \mu^2}{|C'|} + \frac{p^2 \mu^2}{|C'|} + \frac{p^2 \mu^2}{|C'|}
+ \frac{6\mu q}{\sqrt{|C'|}} + 6\mu q + \mu n_{DEC} + q + qH' + 1)^2 \delta + \frac{2(n_{CO} + 1)^2 q \text{Adv}_{\text{MAC}}^\text{CMA}(F) + 2pqH' + \mu q}{\sqrt{|C'|}}
\]
where \(p, q_G, q_H, q_{H'}, n_{OP}, n_{CO}, \) and \(n_{Dec}\) are the number of user in the games, the maximal number of challenge ciphertexts per user, \(A\)'s queries to \(G, H, H', Open, Corrupt,\) and \(Dec,\) respectively, \(q = q_G + q_H.\)

### 6.2 Bi-SO security of \(U_f^m\)

Let \(PKE = (K_{G_0}, Enc_0, Dec_0)\) be a deterministic PKE scheme with public space \(\mathcal{P}_G',\) plaintext space \(\mathcal{M}',\) ciphertext space \(\mathcal{C}',\) and plaintext distribution \(\mathcal{D}_{\mathcal{M}}'.\) Let MAC be a one-time MAC as in \(sPKE_{bi}\). Let \(H, H' : \mathcal{P}_G' \times \mathcal{C}' \rightarrow \mathcal{M} \times K_{mac}\) be two hash functions. We define \(sPKE_{bi}^m\) as in Figure 14. \(sPKE_{bi}^m\) can be viewed as a combination of \(U_f^m\) [HHK17], one-time pad and one-time MAC. Similar to \(sPKE_{bi},\) each user includes its public key into the input of hash functions.

---

<table>
<thead>
<tr>
<th>sKG_{bi}</th>
<th>sEnc_{bi}(pk, m \in \mathcal{M})</th>
<th>sDec_{bi}(pk, k, (e, d, \tau))</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (pk, sk) (\leftarrow K_{G_0})</td>
<td>06 (r \leftarrow D_{\mathcal{M}}')</td>
<td>12 (r' := Dec_0(sk, e))</td>
</tr>
<tr>
<td>02 (k \leftarrow \mathcal{M}')</td>
<td>07 (e := Enc_0(pk, r; G(pk, r)))</td>
<td>13 if (r' = \perp)</td>
</tr>
<tr>
<td>03 (pk' := pk)</td>
<td>08 ((K, K_{mac}) := H(pk, r, e))</td>
<td>14 (or e \in Enc_0(pk, r'; G(pk, r')))</td>
</tr>
<tr>
<td>04 (sk' := (pk, sk, k))</td>
<td>09 (d := K \oplus m)</td>
<td>15 ((K, K_{mac}) := H'(pk, k, e))</td>
</tr>
<tr>
<td>05 return ((pk', sk'))</td>
<td>10 (\tau \leftarrow \text{Tag}(K_{mac}, d))</td>
<td>16 else ((K, K_{mac}) := H(pk, r', e))</td>
</tr>
<tr>
<td>11 return ((e, d, \tau))</td>
<td></td>
<td>17 if (\text{Vrfy}(K_{mac}, \tau) = 1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18 (m := K \oplus d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19 else (m := \perp)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 return (m)</td>
</tr>
</tbody>
</table>

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

Here we consider a variant of OW-CPA security: \(D_{\mathcal{M}}'-OW-CPA\) security, namely, OW-CPA security with challenge messages chosen following \(D_{\mathcal{M}}'.\) The definition of \(D_{\mathcal{M}}'-OW-CPA\) is given in Definition H.1. Moreover, we require that \(PKE\) is rigid correct [BP18], namely, for all \((pk, sk)\) generated from \(K_{G_0}\), ciphertext \(e,\) and plaintext \(r, (e = Enc_0(pk, r))\) if and only if \((\text{Dec}_0(sk, e) = r)\). Theorem 6.2 shows that \(sPKE_{bi}^m\) 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 \(D_{\mathcal{M}}'-OW-CPA\) secure, then the PKE scheme \(sPKE_{bi}^m\) in Figure 14 is adaptive Bi-SO-CCA secure (Definition 2.6). Concretely, for any Bi-SO-CCA adversary \(A\) and relation \(Rel,\) there exist a simulator \(S\) and adversaries \(B^{ow}\) and \(F\) such that \(T(S) \approx T(A) \approx T(B^{ow}) \approx T(F)\) and

\[
\text{Adv}_{sPKE_{bi}}^{\text{Bi-SO-CCA}}(A, S, p, \mu, p, Rel) \\
\leq 6(n_{CD} + n_{OP} + 1)^2 \sqrt{2\mu p\text{Adv}_{OW-PKE, D_{\mathcal{M}}'}^{OW-CPA}(B^{ow}) + \mu p^{2} \text{Adv}_{\text{MAC}}^{\text{CMA-SUF-CMA}(F)}} \\
+ 3p^{2} \text{Adv}_{\text{MAC}}^{\text{CMA-SUF-CMA}(F)} + \frac{6p^{2} \text{Adv}_{\text{MAC}}^{\text{MAC-SUF-CMA}(F)}}{2^{\sigma_{\mathcal{M}'} - n_{Dec}}} + \frac{p^{2} \mu^{2} + p^{2} \mu^{2}}{|\mathcal{M}'|} + \frac{p^{2} \mu^{2}}{|\mathcal{M}'|} \\
+ p\mu^{2} + \frac{2(n_{CD} + 1)^2 \sqrt{pH H' + 2pH' + \mu q}}{\sqrt{|\mathcal{M}'|}}
\]
where \(p, \mu, q_H, q_{H'}, n_{Op}, n_{Co}, \) 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 \(H, H',\) \(\text{OPEN},\) \(\text{CORRUPT},\) and \(\text{DEC},\) respectively. \(\epsilon_{\mathcal{D}_M'}\) is the minimum entropy of \(\mathcal{D}_M'.\)

References


[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.)


[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.)


[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.)


[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.)


Supporting Material

A Review of Adaptive One-way-to-hiding

Let $\mathcal{HF} := \{\{0, 1\}^n \rightarrow \{0, 1\}^n\}$ be a set containing all functions that have $\{0, 1\}^n$ as domain and $\{0, 1\}^n$ as codomain. Let $A = (A_0, A_1)$ 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

\[
P_0^A := \Pr[b' = 1 : \mathcal{H} \xleftarrow{$random$} \mathcal{HF}, m \leftarrow A_0^H(), x \xleftarrow{$random$} \{0, 1\}^l, b' \leftarrow A_0^H(x, \mathcal{H}(x|m))] 
\]

\[
P_1^A := \Pr[b' = 1 : \mathcal{H} \xleftarrow{$random$} \mathcal{HF}, m \leftarrow A_1^H(), x \xleftarrow{$random$} \{0, 1\}^l, B \xleftarrow{$random$} \{0, 1\}^n, b' \leftarrow A_1^H(x, B)]
\]

\[
P_C := \Pr[(x'||m') = (x|m) : \mathcal{H} \xleftarrow{$random$} \mathcal{HF}, m \leftarrow A_0^H(), x \xleftarrow{$random$} \{0, 1\}^l, B \xleftarrow{$random$} \{0, 1\}^n, j \xleftarrow{$random$} [q_0], x'||m' \xleftarrow{$random$} C^H(j, x, B)]
\]

where $q_0, q_1$ are the numbers of time $A_0, A_1$ queries $\mathcal{H}$ respectively. $C$ is an algorithm that has quantum access to $\mathcal{H}$ and on input $(j, B, x)$, runs $A_1^H(x, B)$ until its $j$-th query, measures the QRO query in the computational basis, output the measurement outcome. Then

\[
|P_0^A - P_1^A| \leq 2q_1\sqrt{P_C} + q_02^{-l/2+2}
\]

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., $q_02^{-l/2+2}$), and the second part is the advantage of $A_1$ to distinguish two QROs: $\mathcal{H}(x|m) \rightarrow B$ and $\mathcal{H}$, where $\mathcal{H}(x|m) \rightarrow B$ is the same as $\mathcal{H}$ except that $\mathcal{H}(x|m) \rightarrow B(x|m) = B$. 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 $\text{ADA}$ and $\text{NONADA}$ 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 $|x_1\rangle, |x_2\rangle$, $\text{TD}(|x_1\rangle, |x_2\rangle) = \text{TD}(|x_1\rangle\langle x_1|, |x_2\rangle\langle x_2|)$. Following [NC16], if $\{\alpha_i\}i$ (e.g., the distribution of some random variable $\alpha$ that $\Pr[\alpha = i] = \alpha_i$) and $\{\alpha'_i\}i$ are two distributions with the same index set, then we write $\text{TD}(\{\alpha_i\}i, \{\alpha'_i\}i)$ as the $L_1$-distance of the two distributions:

\[
\text{TD}(\{\alpha_i\}i, \{\alpha'_i\}i) = \frac{1}{2} \sum_i |\alpha_i - \alpha'_i|
\]

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 $\text{TD}(|\phi_0\rangle, |\phi_1\rangle) \leq |||\phi_0\rangle - |\phi_1\rangle||$
We are interested in the trace distance of two pure states that obtains from an algorithm $A$ 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 $X$ and $Y$) and $S$ be a subset of $X$. Let $f'$ and $f$ be two functions with the same preimage set $X$ and image set $Y$ such that for all $x \in X \setminus S$, $f(x) = f'(x)$. Let $U_f : |x,y\rangle \rightarrow |x,y \oplus f(x)\rangle$ and $U_{f'} : |x,y\rangle \rightarrow |x,y \oplus f'(x)\rangle$ be the unitary transformations corresponding to $f$ and $f'$, respectively. Then we have

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

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., $P_S = \sum_{t \in S}|t\rangle\langle t|$.\hfill $\square$

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

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

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

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

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., $P_S = \sum_{t \in S}|t\rangle\langle t|$.\hfill $\square$

**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

$$TD[|\phi_{i+1}\rangle, |\psi_{i+1}\rangle] = TD[T_i U_f |\phi_i\rangle, T_i U_{f'} |\psi_i\rangle]$$
$$= TD[U_f |\phi_i\rangle, U_{f'} |\psi_i\rangle]$$
$$\leq (\ast) TD[U_f |\phi_i\rangle, U_{f'} |\psi_i\rangle] + TD[U_f |\psi_i\rangle, U_{f'} |\psi_i\rangle]$$
$$\leq TD[|\phi_i\rangle, |\psi_i\rangle] + 2\|P_S |\psi_i\rangle\|, \quad (\text{By Lemma B.2})$$

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

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

$\square$

24
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 $\sum p_i |x_i\rangle \langle x_i|$. Let $\Psi$ and $\Phi$ be two density operators, the trace distance between $\Psi$ and $\Phi$ is written as $\|\Psi - \Phi\|$.

For an quantum adversary $A$, it can be modeled as a sequence of unitary transformations $U, U_\Theta, \ldots, U, U_\Theta$ [BDF+11, Unr14b], where $U$ is the transition unitary of $A$ (we can also model it as $U_n, U_\Theta, \ldots, U_1, U_\Theta$, which does not influence our results in this paper). We directly write $O$ to denote the unitary $U_\Theta$.

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 $A$ consists of three quantum registers $A, X,$ and $Y$ without loss of generality, where $A$ is used to store $A$'s internal state, and $X$ and $Y$ are used to store quantum random oracle queries.

- $\Gamma_i (i \in \{1, \ldots, n\})$: $\Gamma_i = (\text{out}_0, \ldots, \text{out}_{i-1})$, where $\text{out}_j$ is the output of $A_j$ for $0 \leq j \leq i - 1$. That is, $\Gamma_i$ stores all outputs of $A$'s first $i$-th stages (i.e., $(A_0, \ldots, A_{i-1})$). 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 $A$.
- $U_{in}, \in$ (or simply $U$): the unitary transformation that models the operations of $A$ when $A$ receives $in_i$ in its $i$-th stage. Without loss of generality, we can assume that $F$ is deterministic (since if $F$ is not deterministic, we can include all randomness in $s$). So, $in_i$ is determined by $\text{out}_i$ for fixed parameter $s$. This means that, if we fixed $A$'s output list $\Gamma$, of its previous $i$ stages (from 0 to $i - 1$), then $in_i$ and $U_i$ are determined.
- $H, H, H_i$: the unitary operations of quantum oracle access to QRO $H$. Specifically, $H : |a, x, y\rangle \rightarrow |a, x, y \oplus H(x)\rangle$. In NONADA, $H = H$ and $H_i$ is independent of $\Gamma_i$. While in ADA, $H = H_i$ in $A$'s $i$-th stage (i.e., the QRO that $A_i$ queries is $H_i$ in ADA) and $H_i$ is dependent to $\Gamma_i$. These unitary transformations do not influence registers $A$ and $X$.

Similarly, we define the final state of $A_0$ in NONADA (resp., in ADA) as $|\phi_{s, in_0, \Gamma_0}\rangle := (UH)^{\#_s} |\varphi_0\rangle$ (resp., $|\psi_{s, in_0, \Gamma_0}\rangle := (UH_0)^{\#_s} |\varphi_0\rangle$). Without loss of generality, we define $|\phi_{s, in_0, \Gamma_{i-1}}\rangle := |\psi_{s, in_0, \Gamma_{i-1}}\rangle := |\varphi_{0}\rangle$ to deal with boundary cases.

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

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

- $\beta_{s, in_0}$: We define $\beta_{s, in_0} := \Pr \left[ (s, in_0) = (s', in'_0) : (s', in'_0, H, H') \leftarrow \text{INIT} \right]$, as the probability that INIT outputs a specific $(s, in_0)$.
- $\alpha_{s, in_0}(\Gamma)$ and $\alpha_{s, in_0}(\Gamma)$ (1 \leq i \leq n+1): $\alpha_{s, in_0}(\Gamma)$ is the probability that $A$'s output list in NONADA is $\Gamma_i$ right after its first $i$ stages (i.e., $(A_0, \ldots, A_{i-1})$). Similarly, $\alpha_{s, in_0}(\Gamma)$ is the probability that $A$'s output list in ADA is $\Gamma_i$ right after its first $i$ stages. To deal with boundary cases, without loss of generality, we define $\alpha_{s, in_0}(\Gamma_0) = \alpha_{s, in_0}(\Gamma_0) = 1$. 

25
• $\alpha_{s,i_0\Gamma_i}(\text{out}_i)$ and $\alpha'_{s,i_0\Gamma_i}(\text{out}_i)$ ($0 \leq i \leq n$): for fixed $(s, i_0)$ and $\Gamma_i$, $\alpha_{s,i_0\Gamma_i}(\text{out}_i)$ is the probability that $A_i$ outputs out$_i$ in NONADA conditioned on $(s, i_0)$ and $\Gamma_i$. Similarly, $\alpha'_{s,i_0\Gamma_i}(\text{out}_i)$ is the probability that $A_i$ outputs out$_i$ in ADA conditioned on $(s, i_0)$ and $\Gamma_i$. We also define $\alpha_{s,i_0\Gamma_i}(\text{out}_1, ..., \text{out}_i)$ (resp., $\alpha'_{s,i_0\Gamma_i}(\text{out}_1, ..., \text{out}_i)$) as the probability that, conditioned on $(s, i_0)$ and $\Gamma_i$, $A_i$ outputs out$_1$, $A_{i+1}$ outputs out$_{i+1}$, ..., and $A_j$ outputs out$_j$ in NONADA (resp., in ADA).

**Lemma 3.1.** Let $\Phi_n$ be the density operator (cf. Supp. Mat. B) of $A_i$'s final state (which is also the final state of $A_n$ so here we add the index $n$) in NONADA$^4$ and $\Psi_n$ be the density operator of $A_i$'s final state in ADA$^4$. By the notation introduced before, these operators can written as:

$$\Phi_n = \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha_{s,i_0}(\Gamma_n) \langle \phi_{s,i_0}, \Gamma_n \rangle \langle \phi_{s,i_0}, \Gamma_n \rangle$$

$$\Psi_n = \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \langle \psi_{s,i_0}, \Gamma_n \rangle \langle \psi_{s,i_0}, \Gamma_n \rangle$$

Similarly, we can define $\Phi_k (0 \leq k \leq n)$ be the density operator of $A_k$'s final state in NONADA$^4$ and $\Psi_k$ be the density operator of $A_k$'s final state in ADA$^4$.

$$\Phi_k = \sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha_{s,i_0}(\Gamma_k) \langle \phi_{s,i_0}, \Gamma_k \rangle \langle \phi_{s,i_0}, \Gamma_k \rangle$$

$$\Psi_k = \sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_k) \langle \psi_{s,i_0}, \Gamma_k \rangle \langle \psi_{s,i_0}, \Gamma_k \rangle$$

Since the event $\text{Ev}$ is a classical event, the probability that $\text{Ev}$ happens in NONADA$^4$ equals to the probability that a binary measurement on the final state of $A$ outputs 1 (indicating the event occurs). Similarly, $\Pr [\text{Ev} : \text{ADA}^4]$ 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),

$$\Pr [\text{Ev} : \text{NONADA}^4] - \Pr [\text{Ev} : \text{ADA}^4] = \text{TD} [\Phi_n, \Psi_n]$$

$$= \text{TD} \left[ \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha_{s,i_0}(\Gamma_n) \langle \phi_{s,i_0}, \Gamma_n \rangle \langle \phi_{s,i_0}, \Gamma_n \rangle, \right.$$ 

$$\left. \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \langle \psi_{s,i_0}, \Gamma_n \rangle \langle \psi_{s,i_0}, \Gamma_n \rangle \right]$$

$$\leq \text{TD} \left[ \{ \beta_{s,i_0} \alpha_{s,i_0}(\Gamma_n) \} s,i_0, \Gamma_n, \{ \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \} s,i_0, \Gamma_n \right]$$

$$+ \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \text{TD} [\langle \phi_{s,i_0}, \Gamma_n \rangle, \langle \psi_{s,i_0}, \Gamma_n \rangle],$$

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 $A_i$'s distributions of $\Gamma_n = (\text{out}_0, ..., \text{out}_{n-1})$ in NONADA and ADA. The quantity on the right of “+” measures the probability difference between $A_i$'s “behaviors” in NONADA and ADA if $A_i$ has the same distribution of $\Gamma_n$ in these two games.

The LHS trace distance bound the probability difference between $A$'s output distributions of $(\text{out}_0, ..., \text{out}_{n-1})$, and these outputs are determined when $A$ completed its first $n$ stages (i.e., $(A_0, ..., A_{n-1})$). Therefore, this trace distance can be bounded by the trace distance between the final states of $A_{n-1}$ in NONADA and ADA. That is,

$$\text{TD} \left[ \{ \beta_{s,i_0} \alpha_{s,i_0}(\Gamma_n) \} s,i_0, \Gamma_n, \{ \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \} s,i_0, \Gamma_n \right] \leq \text{TD} [\Phi_{n-1}, \Psi_{n-1}],$$

and so we have

$$\text{TD} [\Phi_n, \Psi_n] \leq \text{TD} [\Phi_{n-1}, \Psi_{n-1}] + \sum_{s,i_0,\Gamma_n} \beta_{s,i_0} \alpha'_{s,i_0}(\Gamma_n) \text{TD} [\langle \phi_{s,i_0}, \Gamma_n \rangle, \langle \psi_{s,i_0}, \Gamma_n \rangle].$$

26
We can get similar inequalities for \(1 \leq k \leq n\). For \(k = 0\), we have \(\text{TD} [\Phi_0, \Psi_0] \leq \sum_{s,i_0} \beta_{s,i_0} \text{TD} [\langle \phi_{s,i_0,G_0}, \psi_{s,i_0,G_0} \rangle] \). By a simple induction, we have

\[
\text{TD} [\Phi_0, \Psi_0] \leq \sum_{k=0}^{n-1} \sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_k) \text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] .
\]

(13)

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

\[
\sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_k) \text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] ,
\]

(14)

which can be bounded via applying Lemma B.3 \((k + 1)\) times, since here \(\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle\) and \(\langle \psi_{s,i_0,G_k}, \Gamma_k \rangle\) have the same distribution of \(\Gamma_k\). We firstly fix \(s, i_0\) and \(\Gamma\), and look at \(\text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] \).

\[
\text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] = \text{TD} [(UH)^{q_k} U_k \langle \phi_{s,i_0,G_k}, \Gamma_k \rangle, (UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k}, \Gamma_k \rangle] 
\]

\[
\leq \text{TD} [(UH)^{q_k} U_k \langle \phi_{s,i_0,G_k}, \Gamma_k \rangle, (UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k}, \Gamma_k \rangle] 
\]

\[
+ \text{TD} [(UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k}, \Gamma_k \rangle, (UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k}, \Gamma_k \rangle] 
\]

\[
\leq \text{TD} [\langle \phi_{s,i_0,G_k-1}, \psi_{s,i_0,G_k-1} \rangle] + \sum_{j=0}^{q_k-1} 2\|P_{S_k} (UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k-1} \rangle\| ,
\]

where the last inequality comes from Lemma B.3 (our framework assumed that \(H_k \setminus S_k = H \setminus S_k\) for some set \(S_k\)). By induction, we have similar inequalities for \(1 \leq i \leq k\), and thus

\[
\text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] \leq \text{TD} [\langle \phi_{s,i_0,G_0}, \psi_{s,i_0,G_0} \rangle] 
\]

\[
+ \sum_{i=1}^{k-1} \sum_{j=0}^{q_{i-1}-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| .
\]

\[
\text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] \]

can be also bounded by using Lemma B.3, since \(\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle = (UH_k)^{q_k} U_k \langle \phi_{s,i_0,G_k} \rangle\) and \(\langle \psi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle = (UH_k)^{q_k} U_k \langle \psi_{s,i_0,G_k} \rangle\). Therefore,

\[
\text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] \leq \sum_{i=0}^{k-1} \sum_{j=0}^{q_{i-1}-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| .
\]

(15)

Now we have

\[
\sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_k) \text{TD} [\langle \phi_{s,i_0,G_k}, \psi_{s,i_0,G_k} \rangle] 
\]

\[
\leq \sum_{s,i_0,\Gamma_k} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_k) \sum_{i=0}^{k} \sum_{j=0}^{q_i-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| 
\]

\[
\leq \sum_{i=0}^{k} \sum_{s,i_0,\Gamma_i} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_i) \sum_{j=0}^{q_{i-1}-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| 
\]

\[
= \sum_{i=0}^{k} \left( \sum_{s,i_0,\Gamma_i} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_i) \sum_{j=0}^{q_{i-1}-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| \right) 
\]

\[
= \sum_{i=0}^{k} \left( \sum_{s,i_0,\Gamma_i} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_i) \sum_{j=0}^{q_{i-1}-1} 2\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\| \right) 
\]

\[
= \sum_{i=0}^{k} 2q_i \left( \sum_{s,i_0,\Gamma_i} \sum_{j=0}^{q_{i-1}-1} \beta_{s,i_0} \alpha_{s,i_0}^{\prime} (\Gamma_i) \frac{1}{q_i} \left( \sqrt{\|P_{S_i} (UH_i)^{q_i} U_i \langle \psi_{s,i_0,G_{i-1}} \rangle\|} \right)^2 \right) 
\]

27
where (16) comes from Jensen's inequality for concave functions. Now consider the adversary \( B \), in Figure 5. \( B \) firstly chooses a uniformly random \( t^* \leftarrow [q_i] \). Then it simulates \( \text{Ada} \) for \( A \) and performs projective measurement on \( A \)’s \( t^* \) RO-query in \( A \)’s \( i \)-th stage in computational basis. For fix \( t^*, s, in_0 \), the pure state of \( A \) right before the measurement is \( (U(H_i)^{t-1}U_i|\psi_{s,in_0,\Gamma_{i-1}}^i) \), and \( \Gamma_1 \), and the probability that the measurement outcome falls into \( S_i \) is \( \| P_0(U(H_i)^{t-1}U_i|\psi_{s,in_0,\Gamma_{i-1}}^i) \|^2 \), so

\[
\sum_{s, in_0, \Gamma_i, j = 0}^{q_i-1} \beta_{s,in_0} \alpha_{s,in_0}^j (\Gamma_i) \frac{1}{q_i} \| P_0(U(H_i)^{t-1}U_i|\psi_{s,in_0,\Gamma_{i-1}}^i) \|^2 = \text{Pr}[x' \leftarrow B_i^H \text{ s.t. } x' \in S_i : \text{Ada}^{B_i}] > 0
\]

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

\[
\text{TD}_k[\Psi_n, \Psi_n] \leq \sum_{k = 0}^{n-1} \sum_{s, in_0, \Gamma_k} \beta_{s,in_0} \alpha_{s,in_0}^j (\Gamma_k) \text{TD}_k[\phi_{s,in_0,\Gamma_k} | \psi_{s,in_0,\Gamma_k}]
\]

\[
= \sum_{k = 0}^{n-1} \sum_{s, in_0, \Gamma_k} \beta_{s,in_0} \alpha_{s,in_0}^j (\Gamma_k) \text{TD}_k[\phi_{s,in_0,\Gamma_k} | \psi_{s,in_0,\Gamma_k}]
\]

\[
\leq \sum_{k = 0}^{n} \sum_{i = 0}^{q_i} 2q_i \text{Pr}[x' \leftarrow B_i^H \text{ s.t. } x' \in S_i : \text{Ada}^{B_i}] > 0
\]

**Remark C.1** We note that using Unruh’s adaptive OW2H Lemma (cf. Supp. Mat. A and [Unruh14a]) 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 \( A_0, ..., A_i \) are interacting with the same QRO as in NonADA (namely, \( H \) that has never been reprogrammed), and \( A_{i+1}, ..., A_n \) are interacting with the same QRO as in ADA (namely, \( A_j \) 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 [Unruh14a]. However, it is unclear how the reduction can simulate \( H_{i+2}, ..., H_n \) such that \( H \setminus S_i = H_j \setminus S_{i+1} \) 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 \( S_{i+2} \subseteq S_{i+1} \). Now, knowing \( S_{i+2} \), the reduction already breaks the adaptive OW2H lemma without using \( A \).

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 \( \mathcal{L}_o \) according to their notion).

### D Proof of Theorem 5.3

**Theorem 5.3.** Let \( h : M' \times C' \rightarrow M \times K^{\text{ac}} \) and \( g : M' \rightarrow R' \) 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 \( G \times H(R', r, c) := (G(r'), H(r, c)) \). \( A \) can query \( G \times H \) at most \( q = q_H + q_G \) times.

During the proof, we implicitly assume that \( A \) will not query \( \text{Dec} \) on \( (e, d, \tau) \) with \( (e, d) = (e_j, d_j) \) before seeing the challenge ciphertexts \( c \). Since \( r_i \)'s are independent of \( A \)'s view before it sees \( c \), the probability that \( A \) queries \( \text{Dec} \) on such ciphertexts is \( \frac{\mu_{\text{Dec}}}{|C - \text{Dec}|} + \frac{\mu_g}{|M'|} \), 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 \( K_i^{\text{ac}} \)'s. This introduce collision bounds \( \frac{\mu^2}{|M'|} + \mu_G^2 + \frac{\mu^2}{|M'|} + \mu_{\text{Dec}}^2 \). For simplicity, we just add these probability into our final bound, and do not consider it in the game sequences.
Games $G_{0} - G_{3}$

<table>
<thead>
<tr>
<th>Games $G_{0} - G_{3}$</th>
<th>$\text{DEC}(c = (e, d, \tau))$: for $c \notin e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{0}$</td>
<td>$\text{DEC}(c = (sk, e))$</td>
</tr>
<tr>
<td>$G_{3}$</td>
<td>$\text{DEC}(c = (pk, e))$</td>
</tr>
<tr>
<td>$G_{1}$</td>
<td>$\text{DEC}(c = (pk, e))$</td>
</tr>
<tr>
<td>$G_{2}$</td>
<td>$\text{DEC}(c = (pk, e))$</td>
</tr>
</tbody>
</table>

Figure 15: Games $G_{0} - G_{3}$ for the proof of Theorem 5.3.

In games $G_{0} - G_{3}$ (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 $G_{2}$ and $G_{3}$ to deal with the correctness of PKE when using the encrypt-than-hash technique.

In games $G_{3} - G_{5}$ (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 $R_{t}$ from QROs $G$ and $H$. Finally, by using multi-challenge lossiness of PKE$_{1}$, we bound the probability that $A$ learns $r_{t}$ before opening $c[i]$.

**Game $G_{0}$:** This game is equivalent to REAL-SO-CCA$_{\text{spke}}^{A}$, so

$$\Pr[\text{REAL-SO-CCA}_{\text{spke}}^{A} \Rightarrow 1] = \Pr[G_{0} \Rightarrow 1]$$

**Game $G_{1}$:** The DEC oracle computes $(K, K_{\text{nac}}) = h'(e)$ rather than $h(k, e)$ if $r' = \perp$ or $e \neq \text{Enc}_{1}(pk, r'; G(r'))$, where $h': M' \times C' \rightarrow M \times K_{\text{nac}}$ is an internal quantum-accessible random oracles independent of $h$ and $g$. By Lemma 2.7, we have

$$|\Pr[G_{0} \Rightarrow 1] - \Pr[G_{1} \Rightarrow 1]| \leq 2\eta_{h}/\sqrt{|M'|}$$

**Game $G_{2}$:** We restrict the range of $G$ to be the “good” randomness space defined by $(pk, sk)$. Namely, we define the set

$$\mathcal{R}_{\text{good}}(pk_{1}, sk_{1}, r) := \{r' \in \mathcal{R}' \mid \text{Dec}_{1}(sk_{1}, \text{Enc}_{1}(pk_{1}, r; r')) = r\}$$

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

$$|\Pr[G_{1} \Rightarrow 1] - \Pr[G_{2} \Rightarrow 1]| \leq 8(\mu + \eta_{\text{Dec}} + q_{G} + q_{H} + 1)^{2}\delta.$$

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

**Game $G_{3}$:** We set $H(r, e) = h_{1}(e)$ if $e = \text{Enc}_{1}(pk_{1}, r; G(r))$, where $h_{1}: C' \rightarrow M \times K_{\text{nac}}$ 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, $\text{Enc}_{1}(pk_{1}, \cdot; G(h))$ is an injective function and thus $h_{1}(\text{Enc}_{1}(pk_{1}, \cdot; G(h)))$ can be also viewed as a random oracle. Therefore, we have $\Pr[G_{2} \Rightarrow 1] = \Pr[G_{3} \Rightarrow 1]$. 

29
Game $G_4$: We “merge” $h_1$ and $h'$, namely, $\text{Dec}$ always computes $(K, K^{\text{acc}}) := h_1(e)$ regardless of whether $e = \hat{\text{Enc}}_1(pk, r'; G(r'))$. 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 $\text{Dec}$ on honestly-generated $e$, and the only way to learn information from $h'$ is to query $\text{Dec}$ on invalid $e$. $e$ is invalid means that $\text{Dec}_1(s_{K_1}, e) = r$ but $r = \perp$ or $\text{Enc}_1(pk_1, r; G(r)) \neq e$. However, since in this game the randomness in $\text{Enc}_1(pk_1, \cdot; \cdot)$ is generated by $g'$ (always “good” randommesses 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 $\text{Dec}$. In $\mathcal{A}$’s view, the responses of $\text{Dec}$ and $H$ in $G_3$ and $G_4$ still have the same distribution, and thus $\Pr[G_4^A \Rightarrow 1] = \Pr[G_4^A \Rightarrow 1]$.

Game $G_5$: The oracle $G$ is simulated using $g$ instead of $g'$. Similar to the difference between $G_1$ and $G_2$, we have

$$\Pr[G_4^A \Rightarrow 1] - \Pr[G_5^A \Rightarrow 1] \leq 8(\mu + q_G + q_H + 1)^2 \delta.$$ 

Note that in this game, $\text{Dec}$ does not query $G$. Now the game simulator does not need $sk_1$ to simulate $\text{Dec}$.

---

### Figure 16: Games $G_5$-$G_8$ for the proof of Theorem 5.3.

#### Games $G_5$-$G_8$

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>$(pk_1, sk_1, k) \leftarrow sKG, pk := pk_1$</td>
</tr>
<tr>
<td>02</td>
<td>$(lpk_1, lsk_1) \leftarrow \text{LKG}_1, pk := lpk_1$</td>
</tr>
<tr>
<td>03</td>
<td>$\mathcal{M}<em>a \leftarrow A</em>{\text{Dec}, (G \times H)}(pk)$</td>
</tr>
<tr>
<td>04</td>
<td>for $i \in [\mu]$</td>
</tr>
<tr>
<td>05</td>
<td>$m[i] := m_i \leftarrow \mathcal{M}_a$</td>
</tr>
<tr>
<td>06</td>
<td>$r_i \leftarrow \mathcal{M}'$</td>
</tr>
<tr>
<td>07</td>
<td>$R_i := G(r_i)$</td>
</tr>
<tr>
<td>08</td>
<td>$R_i \leftarrow \mathcal{R}'$</td>
</tr>
<tr>
<td>09</td>
<td>$c_i := \hat{\text{Enc}}_1(pk, r_i, R_i)$</td>
</tr>
<tr>
<td>10</td>
<td>$(K_i, K_i^{\text{acc}}) = H(r_i, c_i)$</td>
</tr>
<tr>
<td>11</td>
<td>$d_i := K_i \oplus m_i$</td>
</tr>
<tr>
<td>12</td>
<td>$d_i \leftarrow \mathcal{M}$</td>
</tr>
<tr>
<td>13</td>
<td>$K_i^{\text{acc}} \leftarrow K^{\text{acc}}$</td>
</tr>
<tr>
<td>14</td>
<td>$\tau_i := \text{Tag}(K_i^{\text{acc}}, d_i)$</td>
</tr>
<tr>
<td>15</td>
<td>$c[i] := (c_i, d_i, \tau_i)$</td>
</tr>
<tr>
<td>16</td>
<td>out $\leftarrow A_{\text{OPEN}, \text{Dec}, (G \times H)}(e)$</td>
</tr>
<tr>
<td>17</td>
<td>return $\text{Rel}(\mathcal{M}_a, m, I, out)$</td>
</tr>
<tr>
<td>18</td>
<td>OPEN$(i)$</td>
</tr>
<tr>
<td>19</td>
<td>$I := I \cup {i}$</td>
</tr>
<tr>
<td>20</td>
<td>$(K, K^{\text{acc}}) := h_1(e)$</td>
</tr>
<tr>
<td>21</td>
<td>if $\exists i \in [\mu] \setminus I$ such that $e = e_i$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>22</td>
<td>return $\perp$</td>
</tr>
<tr>
<td>23</td>
<td>if $\exists i \in I$ such that $e = e_i$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>24</td>
<td>$(K, K^{\text{acc}}) := (K_1, K_1^{\text{acc}})$</td>
</tr>
<tr>
<td>25</td>
<td>if $\text{Vrfy}(K^{\text{acc}}, \tau) = 1$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>26</td>
<td>$m := K \oplus d$</td>
</tr>
<tr>
<td>27</td>
<td>else $m := \perp$</td>
</tr>
<tr>
<td>28</td>
<td>return $m$</td>
</tr>
<tr>
<td>29</td>
<td>$H(r, e)$</td>
</tr>
<tr>
<td>30</td>
<td>if $\exists i \in I$ such that $(r, e) = (r_i, e_i)$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>31</td>
<td>return $(d_i \oplus m_i, K_i^{\text{acc}})$</td>
</tr>
<tr>
<td>32</td>
<td>if $e = \hat{\text{Enc}}_1(pk, r; G(r))$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>33</td>
<td>return $h_1(e)$</td>
</tr>
<tr>
<td>34</td>
<td>$G(r)$</td>
</tr>
<tr>
<td>35</td>
<td>if $\exists i \in I$ such that $r = r_i$ then $\text{G}_6$-$\text{G}_8$</td>
</tr>
<tr>
<td>36</td>
<td>return $g(r)$</td>
</tr>
</tbody>
</table>

### Game $G_6$: The public key is switched to lossy mode by $(lpk_1, lsK_1) \leftarrow \text{LKG}_1$ (see Item 02). Note that this game can be simulated without using $sk_1$. By the key indistinguishability of $\text{PKE}_1$, we have

$$\Pr[G_6^A \Rightarrow 1] - \Pr[G_6^A \Rightarrow 1] \leq \text{Adv}_{\text{PKE}_1}(\mathcal{A})$$

Game $G_7$: The decryption oracle always returns $\perp$ if the adversary queries a ciphertext $(c, d, \tau)$ that $e$ is the $\text{PKE}_1$ 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 $\text{PKE}_1$.

Let $\text{Bad}$ be the event that $\mathcal{A}$ queries $\text{Dec}$ on a ciphertext $(c, d, \tau)$ that $e = e_i$ for an $i \in [\mu]$ and $\text{Vrfy}(K_i^{\text{acc}}, d, \tau) = 1$. That is, $\mathcal{A}$ forges valid MAC codes of some unopened ciphertext. Let $\text{Bad}_j := \text{Bad} : G_7^A$ be the event that $\text{Bad}$ happens in $G_7^A(j \geq 7)$. If $\text{Bad}_j$ does not occur, then the winning probabilities of $\mathcal{A}$ in $G_6$ and in $G_7$ are the same. This is because if $e = e_i$ for some $i$ but $\text{Vrfy}(K_i^{\text{acc}}, d, \tau) = 0$, then
by Item 26, the Dec oracle will still rejects the ciphertexts. Thus

\[ \Pr[\text{G}_6^0 \Rightarrow 1] - \Pr[\text{G}_7^d \Rightarrow 1] \leq \Pr[\text{Bad}_7]. \]

Now we cannot bound \( \Pr[\text{Bad}_7] \) by constructing an MAC adversary, since in \( \text{G}_7 \), unopened MAC keys \( K_i^{\text{nac}} \) are related to \( H \). We introduce the next game \( \text{G}_8 \) to bound it.

**Game G_8:** the game simulator generates challenge ciphertexts independent of \( G \) and \( H \) (see Items 08, 12 and 13). To keep \( A \)'s view consistent, when \( A \) 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 \( \Pr[\text{G}_6^d \Rightarrow 1] - \Pr[\text{G}_8^d \Rightarrow 1] \). Firstly, we view \( A \) as \((A_0, A_1) = (A_0, (A_{1,0},...,A_{1,n_{op}}))\), where \( n_{op} \) is the number of OPEN queries. Essentially, \( A_1 \) is divided into \((n_{op}+1)\) stages w.r.t OPEN queries:

- Before any OPEN query (i.e., at the 0-th stage), \( A_{1,0} \) takes \( i_0 := (st, c) \) as input and outputs the first opening index \( out_0 := (i_1) \).
- For \( j \in \{1,...,n_{op}−1\} \), \( A_{1,j} \) receives \( i_j = (m_{ij}, r_{ij}) \) and ends the stage by outputting the \((j+1)\)-th opening index \( out_j := i_{j+1} \).
- Finally, \( A_{1,n_{op}} \) receives \( i_{n_{op}} = (m_{i_{n_{op}}, r_{i_{n_{op}}}}) \) and terminates by outputting \( out_{n_{op}} := out \) (i.e., the final output of \( A_1 \) in Item 16).

For simplicity, we do not consider the \( A_0 \) part, since the output distribution of \( A_0 \) in \( \text{G}_7 \) is the same as in \( \text{G}_8 \). We only assume that \( A_1 \) takes \( A_0 \)'s final state as its initial state.

In Figure 17, we define \( \text{INT}, \text{F}_s \), and \( \text{Repro}_s \) such that \( \text{G}_7 \) is a NONADA game and \( \text{G}_8 \) is a ADA. In \( \text{G}_8 \), when \( A \) 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 \( F_s \) and \( \text{Repro}_s \) in our framework in Figure 4. Therefore, \( \text{G}_7 \) and \( \text{G}_8 \) can be viewed as concrete cases of NONADA and ADA, respectively.

For \( k \in \{0,...,n_{op}\} \), let \( G_k, H_k \) be the QROs that interacts with \( A_{1,k} \) in \( \text{G}_8 \), and let \( G', H' \) be the QROs that interacts with \( A_{1} \) in \( \text{G}_7 \) (the QROs in \( \text{G}_7 \) 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 \( A_{1,k} \). By the definition of \( \text{G}_7 \) and \( \text{G}_8 \) in Figure 16, we always have \( G(r_i) = R_i \) (resp., \( H(c_i, r_i) = (K_i, K_i^{\text{nac}}) \)) for all \( i \in [\mu] \) in \( \text{G}_7 \). But in \( \text{G}_8 \), we have \( G(r_i) = R_i \) (resp., \( H(c_i, r_i) = (K_i, K_i^{\text{nac}}) \)) only if \( i \in I_k \). That is, in \( \text{G}_8 \), \( G(r_i) \neq R_i \) and \( H(c_i, r_i) \neq (K_i, K_i^{\text{nac}}) \) before \( A \) queries OPEN(i). Moreover, for all \( r \notin \{r_i\}_{i \in [\mu]} \) and \( c \notin \{c_i\}_{i \in [\mu]} \), \( G'(r) \) has the same distribution with \( G_k(r) \) and \( H'(c, r) \) has the same distribution with \( H_k(c, r_i) \) for all \( k \in \{0,...,n_{op}\} \). Therefore, answers of \( G_k \times H_k \) differs with answers of \( G' \times H' \) in the following set

\[
S_k := \{ r \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } r = r_i \} \times \{ (r', c') \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } (r', c') = (r_i, c_i) \}
\]

\[
= \{ (r, (r', c')) \mid \exists i \in [\mu] \setminus I_k \text{ s.t. } r = r_i \text{ or } (r', c') = (r_i, c_i) \}. \tag{18}
\]

Similar to the argument in the proof of Theorem 4.1, by using Lemma 3.1, there exist adversaries \( \{B_k\}_{k \in \{0,...,n_{op}\}} \) such that the probability difference between the event that \( G_6 \) output 1 (i.e., \( \text{Rel}(M_{out}, m, I, out) = 1 \) in \( \text{G}_6 \)) and similar event in \( \text{G}_7 \) is bounded by

\[
\left| \Pr[G_7^d \Rightarrow 1 \mid \neg \text{Bad}_7] - \Pr[G_8^d \Rightarrow 1 \mid \neg \text{Bad}_8] \right|
\leq \sum_{k=0}^{n_{op}} \sum_{i=0}^{k} 2\eta \sqrt{\Pr \left[ (r, (r', c')) \leftarrow B_i^{(G \times H)} \text{ s.t. } (r, (r', c')) \in S_k : G_8^i \right] + \frac{2\mu}{\sqrt{|R|}}},
\]

where \( B_k \) plays \( G_8 \) (and also simulates \( G_8 \) for \( A \)), randomly measures \( A_{1,k} \)’s QRO queries, and outputs the measurement outcome. A detailed description of \( B_k \) will be given in Figure 22. Similarly, since \( \text{Bad} \) (defined in \( \text{G}_7 \)) is classical event, by Lemma 3.1 again, we also have

\[
\left| \Pr[\text{Bad}_7] - \Pr[\text{Bad}_8] \right| = \left| \Pr[\text{Bad} : G_7^d] - \Pr[\text{Bad} : G_8^d] \right|
\leq \sum_{k=0}^{n_{op}} \sum_{i=0}^{k} 2\eta \sqrt{\Pr \left[ (r, (r', c')) \leftarrow B_i^{(G \times H)} \text{ s.t. } (r, (r', c')) \in S_k : G_8^i \right] + \frac{2\mu}{\sqrt{|R|}}},
\]
We have the lossiness of Lemma D.2, ability that Lemma D.1 are internal quantum-accessible random oracles. Here the adversary also has classical access to

\[ \mathcal{M}_a \leftarrow \mathcal{A}^{(G \times H), \text{Dec}(pk)} \]  
\[ \text{Dec} \text{ is simulated as in } G_7 \text{ and } G_8 \]

\[ \text{for } i \in [\mu]: \]
\[ m[i] := m_a \leftarrow \mathcal{M}_a, r[i] := r_i \leftarrow \mathcal{M}' \]
\[ R_i := G(r_i), R[i] := R_i \]
\[ e_i = \text{Enc}_i(pk, m_i, R_i) \]
\[ (K_i, K_i^{\text{asc}}) := H(r_i, e_i) \]
\[ d_i := K_i \oplus m_i \]
\[ \tau_i := \text{Tag}(K_i^{\text{asc}}, d_i) \]
\[ c[i] := (e_i, d_i, \tau_i), K_i^{\text{asc}} := K_i^{\text{asc}} \]
\[ s := (\mathcal{M}_a, m, r, c, K_i^{\text{asc}}), \text{ in}_0 := c \]
\[ S_0 := \{ r_i \}_{i \in [\mu]} \times \{ (r_i, e_i) \}_{i \in [\mu]} \]

Let \( G_0 \times H_0 \) be a QRO such that \( G_0 \times H_0(x) := \begin{cases} G \times H(x), & (x \notin S_0) \\ G \times H(x'), & \text{(else)} \end{cases} \)

\[ \text{return } ((s, \text{in}_0), G \times H, G_0 \times H_0) \]

\[ \text{parse } i := \text{out} \]
\[ (\mathcal{M}_a, m, r, c, K_i^{\text{asc}}) := s \]
\[ I := I \cup \{ i \} \]
\[ r_i := r[i], R_i := R[i], m_i := m[i] \]
\[ (e_i, d_i, \tau_i) := c[i], K_i^{\text{asc}} := K_i^{\text{asc}}[i] \]
\[ \text{in} := (r_i, m_i) \]
\[ \text{in}' := (r_i, m_i, e_i, d_i, K_i^{\text{asc}}) \]
\[ \text{return } (\text{in}', G \times H) \]

Figure 17: Construction of INIT, INIT, F\textsc{a}, and Rep\textsc{o} 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 \( \text{Dec} \). Since \( \text{Dec} \) will not make \( G \times H \) reprogrammed, allowing the adversary to query \( \text{Dec} \) does not change the bound of Lemma 3.1.

These bounds include a term \( \frac{\mu q \sqrt{\mathcal{M}}}{\sqrt{n}} \), since \( \mathcal{A}_0 \) also has quantum access to \( |G \times H| \), and this term is the probability that the first stage (i.e., \( A_1, g \)) of \( A_1 \) learns \( r_i \) before opening challenge ciphertexts. Such probability is only information-theoretic.

In \( G_8, K_i^{\text{asc}} \) are independent of challenge ciphertexts \( c \) (before \( A \) queries \( \text{OPEN}(i) \)), so we can upper bound \( \Pr[\text{Bad}_8] \) by the otSUF-CMA security of otSUF-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 \( F \) (cf. Figure 21) such that

\[ \Pr[\text{Bad} : G_8^F] \leq \mu \text{Adv}_{\text{otSUF-CMA}} \]

We also bound the winning probability of \( B_k \) in Lemma D.2. This probability captures the “probability” that \( A \) learns \( r_i \) where \( c[i] \) is not opened. Intuitively, since the public key in \( G_8 \) is lossy, by the lossiness of \( \text{PKE}_1 \), ciphertexts that encrypted by lossy key statistically hide the information of their plaintexts. The concrete bound is given in D.2.

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

\[ \Pr[(r, (r', e')) \leftarrow B_k^{(G \times H)} \text{ s.t. } (r, (r', e')) \in S_k : G_8^{B_k}] \leq \epsilon_{\text{ind-enc}} + \frac{\mu q}{\mathcal{M}} \]

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

\[ |\Pr[G_8^F \Rightarrow 1] - \Pr[G_8^F \Rightarrow 1]| \]
\[
\leq 4(n_{OP} + 1)^2 q \sqrt{e_{\text{in-dec}} + \frac{\mu q}{|M'|}} + \mu \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(F) + \frac{4\mu q}{\sqrt{|R'|}}
\]

Now we can construct a simulator \( S \) that simulates \( G_8 \) for \( A \) and interacts with the IDEAL-SO-CCA_{\text{PKE}} game. Its simulation process is given in Figure 18. If \( A \) outputs \( out \), then \( S \) also outputs \( out \) except that \( \text{Bad}_S \) happens. We have

\[
\Pr[\text{IDEAL-SO-CCA}_{\text{PKE}}^S \Rightarrow 1] \leq \Pr[G_8^A \Rightarrow 1] + \mu \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(F)
\]

By combining all the probability bounds, we have

\[
\text{Adv}_{\text{PKE}}^{\text{SOF-CCA}}(A, S, \mu, \text{Rel}) = \left| \Pr[\text{REAL-SO-CCA}_{\text{PKE}}^D \Rightarrow 1] - \Pr[\text{IDEAL-SO-CCA}_{\text{PKE}}^S \Rightarrow 1] \right| \leq \text{Adv}_{\text{PKE}}^{\text{ind-key}}(A) + 3\mu \text{Adv}_{\text{MAC}}^{\text{SUF-CMA}}(F) + 6(n_{OP} + 1)^2 q \sqrt{e_{\text{in-dec}} + \frac{\mu q}{|M'|}} + 16(\mu + n_{\text{Dec}} + q + 1)^2 \delta + (2 + \mu) q \sqrt{|M'|} + 6\mu q \sqrt{|R'|} + \frac{\mu^2}{|M'|} + \frac{\mu^2}{|R'|} + \frac{\mu^2}{|K_\text{mac}|} + \frac{\mu^2}{|C' - n_{\text{Dec}}|} + \frac{\mu^2}{|M'|}
\]

\[\square\]

D.1 Bounding \( G_1 \) and \( G_2 \) in Theorem 5.3

In the proof of Theorem 5.3, we defined a set

\[
\begin{align*}
\mathcal{R}_\text{good}^\prime(\text{pk}_1, \text{sk}_1, r) &:= \{r' \in \mathcal{R}' \mid \text{Dec}_1(\text{sk}_1, \text{Enc}_1(\text{pk}_1, r; r')) = r\} \\
\mathcal{R}_\text{bad}^\prime(\text{pk}_1, \text{sk}_1, r) &:= \mathcal{R}' \setminus \mathcal{R}_\text{good}^\prime(\text{pk}_1, \text{sk}_1, r)
\end{align*}
\]

which contains all “good” randomness with respect to the key pair \((\text{pk}_1, \text{sk}_1)\), namely, if a randomness \( r' \in \mathcal{R}_\text{good}^\prime(\text{pk}_1, \text{sk}_1, r) \) is sampled in encrypting a message \( r \), then the resulting ciphertext will be decrypted
Based on this, we define a set that contains “bad” randomness:

\[ R_{bad}^\prime(p_k, s_k, r) = R^\prime \setminus R_{good}^\prime(p_k, s_k, r). \]

Based on these two sets, we further define

\[ \delta(p_k, s_k) = \max_{r \in \mathcal{M}^\prime} \left( \frac{|R_{bad}^\prime(p_k, s_k, r)|}{|R^\prime|} \right) \]

where the former captures the maximal probability of decryption error with respect to a fixed key pair \((p_k, s_k)\), and the expectation of the latter is taken over \((p_k, s_k) \leftarrow KG_1\). By Definition 2.1, \(\delta\) 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 \(G_1\) and \(G_2\) 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 \(A\) issuing at most \(q\) quantum queries,

\[ |\Pr[GDPB_{\lambda,0}^A \Rightarrow 1] - \Pr[GDPB_{\lambda,1}^A \Rightarrow 1]| \leq 8(q + 1)^2 \lambda, \]

where games \(GDPB_{\lambda,b}^A\) are defined in Figure 19.

---

**Figure 19**: Game GDPB_{\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 \(B\) in Figure 20 that plays \(GDPB_{b(p_k, s_k),0}^B\) where \((p_k, s_k) \leftarrow KG_1\). \(Samp\) is a sampling process and \(f\) is a random function used to generate randomness \(Samp\). \(B\) can construct such \(Samp\) and \(f\) since it is unbounded.

If \(B\) is playing \(GDPB_{b(p_k, s_k),0}^B\), then \(F(r)\) always outputs 0, and thus \(G(r)\) always outputs “good” randomness. This corresponds to \(G_2\); Otherwise, \(F(r)\) outputs 1 with probability \(|R_{bad}^\prime(p_k, s_k, r)|/|R^\prime|\) and thus \(G(r)\) outputs “bad” randomness with such probability. This means that the \(G(r)\) outputs are uniformly distributed over \(R^\prime\) and thus it behaves as in \(G_1\). Considering the expectation over \((p_k, s_k) \leftarrow KG_1\), we have

\[ |\Pr[G_1^A \Rightarrow 1] - \Pr[G_2^A \Rightarrow 1]| \]

\[ = \mathbb{E}_{(p_k, s_k) \leftarrow KG_1} \left[ |\Pr[GDPB_{b(p_k, s_k),1}^B \Rightarrow 1] - \Pr[GDPB_{b(p_k, s_k),0}^B \Rightarrow 1]| \right] \]

\[ = 8(q + 1)^2 \mathbb{E}_{(p_k, s_k) \leftarrow KG_1} [\delta(p_k, s_k)] \]

\[ = 8(\mu + n_{Dec} + q_G + q_H + 1)^2 \delta, \]

since \(B\) issues at most \(\mu + n_{Dec} + q_G + q_H\) quantum queries to \(F\).

---

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

**Lemma D.1.** In Figure 21, we construct a forger \(F\) that simulates \(G_A\) for \(A\) and forges a valid message-tag of otSUF-CMA. By Definition 2.4, \(F\) has access to oracles \(TAG\) (at most one query) and \(VRFY\).
We assume that a pair (see Items Open Lemma) that the ciphertext \( \tau \) key of \( B \) is picked by game Figure 21: Picks a random function \( F \), Chooses QROs \( e, m \) (I I I 19 Output \( m \) else D.1 5.3 return Bad return return M 15 In if shows the modified game The forger \( d \) else if return 5.3 m 22 Bad to \( \lambda \) c if return 3.1 if (19 3.1 for \( \mu \) Adv in this proof is shown in Figure 8 to fit into By our framework in Lemma B A The probability that B s output. That is, \( \forall k \in \{0, \ldots, n_{QRO}\} \) is an adversary that runs \( A_{1,0} \ldots A_{1,k} \) and randomly measures \( A_{1,k} \)'s QRO query. The construction of \( B_k \) in this proof is shown in Figure 22.

By our framework in Lemma 3.1, \( B_k \) interacts with an ADA game. As shown in the proof of Theorem 5.3, \( G_8 \) can be viewed as an ADA game, and the OPEN oracle is the \( F_k \) function, since queries to OPEN will make QROs reprogrammed.

Since \( B_k \) finally outputs the measurement outcome of one of \( A_{1,k} \)'s QRO query, we slightly modify \( G_8 \) to fit into \( B_k \). Figure 23 shows the modified game \( G_8' \). Game \( G_8' \) is the same as \( G_8 \) except that \( G_8' \) runs \( B_k \) and outputs \( B_k \)'s output. That is,

\[
\Pr \left[ (r, (r', e')) \leftarrow B_k^{(G \times H)} \text{ s.t. } (r, (r', e')) \in S_k : G_8 \right]
\]
$B_k^{\text{Open, Dec} : (G \times H)}(c), k \in [0, \ldots, n_{op})$

1. $t^* \leftarrow [q_k]$
2. $i_{0} := c$
3. for $j = 0$ to $k - 1$:
   4. $out_j := i \leftarrow A_{1,j}^{(G \times H), \text{Dec}}(i_{j})$ \hspace{1cm} // $A_{1,j}$ wants to open ciphertext $c[i]$
   5. $(m_{i}, r_{i}) := \text{OPEN}(out_j)$ \hspace{1cm} // Queries OPEN(i) and ...
   6. $i_{j+1} := (m_{i}, r_{i})$ \hspace{1cm} // ... gets the message and randomness of $c_i$
7. Runs $A_{1,j}^{(G \times H), \text{Dec}}(i_{n})$ until it issues $t^*$-th quantum query to $G \times H$
8. Let $|\varphi|$ be the $t^*$-th quantum query to $G \times H$
9. $(r, (r', e')) \leftarrow \text{Measure}(|\varphi|)$
10. return $(r, (r', e'))$

Figure 22: The constructions of $B_k(0 \leq k \leq n_{op})$ in the proof of Lemma D.2. If $A_{1,j}$ queries $G \times H$ or $\text{Dec}$, $B_k$ just forwards these queries to its game simulator and then forwards the response to $A_{1,j}$. Since $(A_{1,0}, \ldots, A_{1,n_{op}})$ is obtained by dividing $A_1$ into $(n_{op} + 1)$ stages with respect to $A_1$’s queries to $\text{OPEN}$, $A_{1,j}$ terminates with outputting the $j$-th opening index $i_j$.

$$= \Pr \left[ G_k^{G_{s_k}} \Rightarrow (r, (r', e')) \text{ s.t. } (r, (r', e')) \in S_k \right],$$

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., $B_k$ does not open $c[i]$). To bound the probability that $B_k$ outputs $(r, (r', e')) \in S_k$, we consider the games $G_k'$ and $G_0'$ in Figure 23.

### Games $G_k'$ and $G_0'$ for $B_k(k \in [0, \ldots, n_{op}])$

1. $(pk_{k}, (sk_k, k)) \leftarrow \text{sKG}, pk := pk_k$
2. $(lpk_k, sk_k) \leftarrow \text{LKG}(pk_k), pk := lpk_k$
3. $M_{a} \leftarrow A_0^{(G \times H), \text{Dec}}(pk)$
4. for $i \in [\mu]$
   5. $m[i] := m_i \leftarrow M_{a}$
   6. $r_{i} \leftarrow \text{M}'$
   7. $R_{i} \leftarrow R'$
   8. $e_i := \text{Enc}_1(pk, r_i; R_i)$
   9. $r'_i \leftarrow \text{M}', R'_i \leftarrow R'$
   10. $e_i := \text{Enc}_1(pk, r'_i; R'_i)$
   11. $R_i \leftarrow \text{OPEN}(lsk_k, lpk_k, e_i, R'_i, r_i)$
   12. $d_i \leftarrow M, K^{\text{Enc}} \leftarrow K^{\text{Enc}}$
   13. $\tau_i := \text{Tag}(K^{\text{Enc}}, e_i, d_i)$
   14. $c[i] := (e_i, d_i, \tau_i)$
15. $(r, (r', e')) \leftarrow B_k^{\text{Open, Dec} : (G \times H)}(st, c)$
16. return $(r, (r', e'))$

### $\text{Dec}(c = (e, d, r))$: for $c \notin c$

19. $(K, K^{\text{Enc}}) := h_1(e)$
20. if $\exists i \in [\mu] \setminus I$ s.t. $e = e_i$
21. return $\bot$
22. if $\exists i \in I$ s.t. $e = e_i$
23. $(K, K^{\text{Enc}}) := (K_i, K^{\text{Enc}})$
24. if $\forall r_i(K^{\text{Enc}}, r) = 1$
25. $m := K \oplus d$
26. else $m := \bot$
27. return $m$
28. $H(r, e)$
29. if $\exists i \in I$ s.t. $(r, e) = (r_i, e_i)$
30. return $(d_i \oplus m_i, K^{\text{Enc}})$
31. if $e = \text{Enc}_1(pk, r; G(r))$
32. return $h_1(e)$
33. return $h(r, e)$
34. $G(r)$
35. if $\exists i \in I$ s.t. $r = r_i$
36. return $R_i$
37. return $g(r)$

Figure 23: Games $G_k'$ and $G_0'$ and constructions of $B_k(k \in [0, \ldots, n_{op}])$ for the proof of Lemma D.2.

**Game $G_k'$**: 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 $\text{open}_1$ to claim the ciphertext $e_i$ to the PKE$_1$ message $r_i$ with randomness $R_i$. Similar to $G_k$, 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 $\text{open}_1$, $R_i$ has the same distribution with $R'_i$. Since the public key in $G_k'$ and $G_0'$ is lossy, by the
multi-challenge lossiness of \( \text{PKE}_1 \) (Definition 5.2), we have
\[
\left| \Pr \left[ G_8^{B_k} \Rightarrow (r, (r', e')) \text{ s.t. } (r, (r', e')) \in S_k \right] - \Pr \left[ G_9^{B_k} \Rightarrow (r, (r', e')) \text{ s.t. } (r, (r', e')) \in S_k \right] \right| \leq \epsilon_{\text{m-ind-enc}}^{\text{PKE}_1}.
\]

Moreover, in \( G_9^i \), \( r_i \) and \( R_i \) are uniformly random and independent of \( B_k \)'s view before \( B_k \) opens \( c[i] \).
By using a union bound over all QRO queries, we get
\[
\Pr \left[ G_9^{B_k} \Rightarrow (r, (r', e')) \text{ s.t. } (r, (r', e')) \in S_k \right] \leq \frac{q \mu}{|M'|}.
\]
Therefore, we have
\[
\Pr \left[ (r, (r', e')) \leftarrow B_k^{(G \times H)} \text{ s.t. } (r, (r', e')) \in S_k : G_8^{B_k} \right] \leq \epsilon_{\text{m-ind-enc}}^{\text{PKE}_1} + \frac{q \mu}{|M'|},
\]
as stated in Lemma D.2.

\[\square\]

### 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, \( \chi \) be a discrete distribution over \( \mathbb{Z}_q \). We say that the \( \text{LWE}_{n, m, q, \chi} \) assumption holds, if for every PPT algorithm \( B \), the following advantage is negligible in \( n \):
\[
\text{Adv}_{\text{LWE}_{n, m, q, \chi}}(B) := \left| \Pr[B(A, b) = 1 \mid A \leftarrow \mathbb{Z}_q^{n \times m}, B \leftarrow \mathbb{Z}_q^m] - \Pr[B(A, A^T s + e) = 1 \mid A \leftarrow \mathbb{Z}_q^{n \times m}, s \leftarrow \mathbb{Z}_q^n, e \leftarrow \chi^m] \right|.
\]

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

Moreover, for any \( y \in \mathbb{Z}_q^n \) and large enough \( s = O(\sqrt{n \log q}) \), the randomized algorithm \( \text{SampleD}(R, B, y, s) \) samples from a distribution within \( \text{negl}(n) \) statistical distance of \( D_{\Lambda_B^\top}(B, s \omega(\sqrt{n})) \), where \( \Lambda_B^\top(B) \) is defined as the set \( \{ z \in \mathbb{Z}^m \mid Bz = y \} \).

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

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

Let integer \( m \geq 2n \cdot \log q \) and \( D_{\mathbb{Z}_q^m} \), be the discrete Gaussian distribution over \( \mathbb{Z}_q^m \) with parameter \( r \). Our lossy encryption \( \text{LWE}_{\text{PKE}_1} \) with message space \( M' := \{0, 1\}^\ell \) is described in Figure 24. It has randomness space \( R' := D_{\mathbb{Z}_q^m} \), and ciphertext space \( C' := \mathbb{Z}_q^m \times \mathbb{Z}_q^r \). In its lossy mode, it requires the \( G \)-trapdoor technique [MP12] to achieve weak openability, and we recall the useful lemma as follow.
Theorem E.3 Let λ be a security parameter. If we use the following parameter setting \( n = \text{poly}(\lambda), \ell = O(n), \) prime \( q \in [n^2, n^4], m = O(n \log(q)), r = O(\sqrt{m \log(n)}), \alpha = \frac{1}{(m \log(n))}, \) then the PKE scheme \( \text{LWEPEKE}_1 \) in Figure 24 is a lossy encryption. Specifically, \( \text{LWEPEKE}_1 \) is \( (1 - \negl(n)) \)-correct, \( \epsilon_{\text{ind-enc}}^{\text{LWEPEKE}_1} = \negl(n), \) and for any adversary \( A, \) we have

\[
\text{Adv}_{\text{LWEPEKE}_1}^{\text{ind-key}}(A) \leq \ell \cdot \text{Adv}_{\text{LWEPEKE}}^{\text{LWEPEKE}_1(n,m)}(B) + \negl(n).
\]

Proof. Correctness. This is very similar to [GPV08]. By the parameters in Theorem E.3 and the following lemma, the scheme \( \text{PKE}_1 \) is \( (1 - \negl(n)) \)-correct.

Lemma E.4 ([GPV08, Lemma 8.2]) If \( q > 5r m \) and \( \alpha \leq 1/(r \sqrt{m} \cdot \omega(\sqrt{\log(n)})), \) then \( \text{Dec}_1 \) in Figure 24 decrypts correctly with overwhelming probability.

Key indistinguishability. We use the \( \text{LWE}_{n,m,q,x} \) assumption \( \ell \) times to show that \( P \) generated by \( \text{KG}_1 \) is indistinguishable from a random matrix in \( \mathbb{Z}^{n \times m}_q. \) Hence, our real key \( \text{pk}_1 \) is pseudorandom. Moreover, by Lemma E.2, our lossy key \( \text{lpk}_1 \) generated using \( \text{GenTrap} \) is \( \negl(n) \)-far from uniform. Hence, we have

\[
\text{Adv}_{\text{LWEPEKE}_1}^{\text{ind-key}}(A) \leq \ell \cdot \text{Adv}_{\text{LWEPEKE}}^{\text{LWEPEKE}_1(n,m)}(B) + \negl(n),
\]

Lossiness. Let \( (A, P) \) be generated by \( \text{LKG}_1. \) Again, by Lemma E.2, \( B = \left( \begin{array}{c} A \\ P \end{array} \right) \in \mathbb{Z}^{(n+\ell) \times m}_q \) is \( \negl(n) \)-close to uniform. If \( B \) is uniform, by the leftover hash lemma in [AP08, Section 2.2.1], the distribution of \( \text{Be} \) is statistically close to the uniform distribution over \( \mathbb{Z}^{n+\ell}_q. \) Hence, \( \epsilon_{\text{ind-enc}}^{\text{LWEPEKE}_1} = \negl(n). \)

(weak) Openability. The openability of our scheme \( \text{LWEPEKE}_1 \) 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 \( e. \) By the lossiness, any ciphertext is a valid ciphertext of \( m'. \) According to \( \text{open}_1 \) in Figure 24, \( e' \leftarrow \text{SampleD}(R, B, c, s) \) will satisfy \( \text{Be'} + \left( \begin{array}{c} 0 \\ \text{Enc}_e(m') \end{array} \right) = \left( \begin{array}{c} u' \\ v \end{array} \right) \) and \( e' \in D_{\mathbb{Z}^{n \times m}}. \)

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.
During the proof, we implicitly assume that $A_0$ will not query $\text{Dec}$ on $(e, d, r)$ with $(e, d) = (e_i, d_i)$ before seeing the challenge ciphertexts $c_i$. Since $r_i's$ are independent of $A_i$'s view before it sees $c_i$, the probability that $A$ queries $\text{Dec}$ on such ciphertexts is $\frac{\mu_{\text{Dec}}}{|\mathcal{M}|} + \frac{\mu_{\text{Enc}}}{|\mathcal{M}|}$, 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 $K_i^{\text{asc}}'s$. This introduces collision bounds $\frac{\mu_{\text{Enc}}}{|\mathcal{M}|} + \frac{\mu_{\text{Dec}}}{|\mathcal{M}|}$. For simplicity, we just add these probability into our final bound, and do not consider it in the game sequences.

$$\begin{array}{|c|}
\hline
\text{Game } G_0 - G_7 & \text{Dec}(e = (e, d, r)): \text{ for } e \notin c_i \\
\hline
01 & (\text{pk}, (\text{sk}, k)) \leftarrow \text{sKG} \\
02 & M_a \leftarrow A^{\text{Enc}, \text{Dec}; \text{Enc}, \text{Dec} \mid G \times H}(\text{pk}) \\
03 & \text{for } i \in [\mu] \\
04 & m_i \leftarrow M_a \\
05 & r_i \leftarrow \mathcal{M}' \\
06 & R_i \leftarrow \text{G}(r_i) \quad \text{\# } G_0 - G_6 \\
07 & R_i \leftarrow \text{R}' \quad \text{\# } G_7 \\
08 & e_i \leftarrow \text{Enc}_0(\text{pk}, r_i; R_i) \\
09 & (K_i, K_i^{\text{asc}}) \leftarrow \text{H}(r_i, e_i) \quad \text{\# } G_0 - G_6 \\
10 & d_i \leftarrow K_i \oplus m_i \quad \text{\# } G_0 - G_6 \\
11 & K_i^{\text{asc}} \leftarrow K_i^{\text{asc}} \quad \text{\# } G_7 \\
12 & \tau_i \leftarrow \text{Tag}(K_i^{\text{asc}}, d_i) \\
13 & c_i \leftarrow (e_i, d_i, \tau_i) \\
14 & \text{out} \leftarrow A^{\text{Enc, Dec}, \text{Enc} \mid G \times H}(c) \\
15 & \text{return } \text{Rel}(M_a, m_i, \text{out}) \\
\hline
\text{Oracle } H(r, e) & \text{Open}(i) \\
17 & \text{if } \exists i \in \text{ s.t. } (r, e) = (r_i, e_i) \quad \text{\# } G_7 \\
18 & \text{return } (d_i \oplus m_i, K_i^{\text{asc}}) \quad \text{\# } G_7 \\
19 & \text{if } e = \text{Enc}_0(\text{pk}, r_i; G(r)) \quad \text{\# } G_5 - G_6 \\
20 & \text{return } h_1(e) \quad \text{\# } G_5 - G_6 \\
21 & \text{return } h(r, e) \\
\hline
\end{array}$$

Figure 25: Games $G_0 - G_7$ for the proof of Theorem 4.2.

Game $G_0$: is equivalent to REAL-SO-CCA$^4_{\text{PKE}}$, so

$$\Pr[\text{REAL-SO-CCA}^4_{\text{PKE}} \Rightarrow 1] = \Pr[G_0^4 \Rightarrow 1]$$

**Game $G_1$:** The Dec oracle computes $(K, K^{\text{asc}}) = h'(e)$ rather than $h(k, e)$ if $r' = \bot$ or $e \notin \text{Enc}_0(\text{pk}, r'; G(r'))$. By Lemma 2.7, we have

$$|\Pr[G_0^4 \Rightarrow 1] - \Pr[G_1^4 \Rightarrow 1]| \leq 2\eta_H/\sqrt{|\mathcal{M}'|}$$

**Game $G_2$:** We restrict the range of $G$ to be the “good” randomness space wrt $(\text{pk}, \text{sk})$. Specifically, let $\mathcal{R}_{\text{good}}(\text{pk}, \text{sk}, r)$ be the set in $\mathcal{R}'$ such that $\text{Dec}_0(\text{sk}, \text{Enc}_0(\text{pk}, r; G(r'))) = r$ and $g' : \mathcal{M}' \rightarrow \mathcal{R}'$ be a quantum-accessible random oracle such that $g'(r)$ is sampled uniformly from $\mathcal{R}_{\text{good}}(\text{pk}, \text{sk}, r)$. If PKE is $(1 - \delta)$-correct (see Definition 2.1), then by a similar argument in Supp. Mat. D.1, we have

$$|\Pr[G_1^4 \Rightarrow 1] - \Pr[G_2^4 \Rightarrow 1]| \leq 8(\mu + n_{\text{Dec}} + q_G + q_H + 1)^2\delta.$$  

**Game $G_3$:** We set $H(r, e) = h_1(e)$ if $e = \text{Enc}_0(\text{pk}, r; G(r))$, where $h_1 : C \rightarrow \mathcal{M} \times K^{\text{asc}}$ is an internal quantum-accessible random oracle. Since the randomness generated by $G$ (i.e., $g'$) is always a “good” randomness, $\text{Enc}_0(\text{pk}, \cdot; G(\cdot))$ is an injective function and thus $h_1(\text{Enc}_0(\text{pk}, \cdot; G(\cdot)))$ can be also viewed as a random oracle. Therefore, we have $\Pr[G_2^4 \Rightarrow 1] = \Pr[G_3^4 \Rightarrow 1]$.  

40
Game $G_4$: We “merge” $h_1$ and $h'$, namely, Dec always computes $(K, K^{\text{acc}}) := h_1(e)$ regardless of the validity of $e$. Similar to the argument in the proof of Theorem 5.3, we have $\Pr[G_4^A \Rightarrow 1] = \Pr[G_4^A \Rightarrow 1]$.

Game $G_5$: The oracle $G$ is simulated using $g$ instead of $g'$. Similar to the difference between $G_1$ and $G_2$, we have

$$\Pr[G_1^A \Rightarrow 1] - \Pr[G_2^A \Rightarrow 1] \leq 8(\mu + n_{\text{Dec}} + q_G + q_H + 1)^2\delta.$$

Game $G_6$: The decryption oracle always returns $\bot$ if the adversary queries a ciphertext $(e, d, \tau)$ that $e$ is the PKE part of some unopened challenge ciphertexts, i.e., $\exists i \in [\mu]\setminus I, e = e_i$. This game is necessary for simulating the decryption oracle without secret key when constructing OW-CPA reduction.

Let $\text{Bad}_j$ be the event that $A$ queries $\text{Dec}$ on a ciphertext $(e, d, \tau)$ that $\exists i \in [\mu]\setminus I, e = e_i$ and $\text{Vrfy}(K^{\text{acc}}, d, \tau) = 1$ in $G_j(j \geq 6)$. That is, $A$ forges valid MAC codes of some unopened ciphertext. If $\text{Bad}_0$ does not occur, then the winning probabilities of $A$ in $G_5$ and $G_6$ are the same. This is because if $\exists i \in [\mu]\setminus I, e = e_i$ but $\text{Vrfy}(K^{\text{acc}}, d, \tau) = 0$, then by lines 19 and 31, the DEC oracle will still reject the ciphertexts. Thus $\Pr[G_5^A \Rightarrow 1] - \Pr[G_6^A \Rightarrow 1] \leq \Pr[\text{Bad}_0]$.

Note that now we cannot bound $\Pr[\text{Bad}_0]$ by constructing an MAC adversary, since in $G_6$, unopened MAC keys $K^{\text{acc}}$ are related to $H$. In the next game $G_7$, the MAC keys, and symmetric keys will be generated independent of $G$ and $H$ at first, and if $A$ opens a challenge ciphertext, we reprogram $G$ and $H$ to make the simulation consistent.

Game $G_7$: 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 $G_7$ will make the QRO reprogrammed, while the QRO in $G_6$ always remains the same. So, we can view $G_6$ and $G_6$ as concrete cases of NONADA and ADA, respectively. We focus on $A_1$ since $A_0$ will not make the QRO reprogrammed. Similar to the proof of Theorem 5.3 in Section 5, we view $A$ as $(A_0, A_1) = (A_0, (A_{1,0}, ..., A_{1,n_{\text{ev}}}))$ and $A_1$ is divided into $(n_{\text{ev}} + 1)$ stages wrt OPEN queries:

- Before any OPEN query (i.e., at the 0-th stage), $A_{1,0}$ takes $i_0 := (st, c)$ as input and outputs the first opening index $i_{0} := (i_1)$.
- For $j \in \{1, ..., n_{\text{ev}} - 1\}, A_{1,j}$ receives $i_j = (m_{i_j}, r_{i_j})$ and ends the stage by outputting the $(j + 1)$-th opening index $i_{j + 1} := i_j$.
- Finally, $A_{1,n_{\text{ev}}}$ receives $i_{n_{\text{ev}}} = (m_{i_{n_{\text{ev}}}}, r_{i_{n_{\text{ev}}}})$ and terminates by outputting $\text{out}_{n_{\text{ev}}} := \text{out}$.

We do not consider the $A_0$ part and only assume that $A_1$ takes $A_0$’s final state as its initial state. In Figure 26, we define $\text{INIT}_S$, $F_s$, and $\text{Repro}_S$. If we instantiate NONADA using $\text{INIT}$, $F_s$, and $\text{Repro}_S$, then the instantiated game is equivalent to the part of $G_6$ that interacts with $A_1$. Similarly, it we instantiate ADA using $\text{INIT}$, $F_s$, and $\text{Repro}_S$, then the instantiated game is equivalent to the part of $G_7$ that interacts with $A_1$. Moreover, at $A$’s $k$-th stage, our $F_s$ defines a set

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

where $I_k := \{i_1, ..., i_k\}$ is the opening index set $I$ in $A_1$’s $k$-th stage (i.e., $A_{1,k}$). Answers of $G \times H$ in $G_6$ (i.e., NONADA) and $G_7$ (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 $G_6$ output 1 (i.e., $\text{Rel}(M_{i_0}, m, I, \text{out}) = 1$ in $G_6$) and similar event in $G_7$ is bounded by

$$\Pr[G_6^A \Rightarrow 1 | \neg \text{Bad}_0] - \Pr[G_7^A \Rightarrow 1 | \neg \text{Bad}_7] \leq \sum_{k=0}^{n_{\text{ev}}} \sum_{i=0}^{k} 2q_i \Pr[(r, (r', e')) \leftarrow B_i^{(G \times H)} \text{ s.t. } (r, (r', e')) \in S_i : G_i^{S_i}] + \frac{2\mu q}{\sqrt{|K|}},$$

where $\text{Ev}$ here is the event that $A_1$ outputs a particular opening index set $I$ and final output $\text{out}$, and $\text{INIT}$, $F_s$, and $\text{Repro}_S$ are defined in Figure 26. It includes a term $\frac{2\mu q}{\sqrt{|K|}}$, since $A_0$ also has quantum access to $|G \times H|$, and this term is the probability that the first stage (i.e., $A_{1,0}$) of $A_1$ learns $r_i$ before
adversary that simulates opening challenge ciphertexts. Moreover, similar to the argument in bounding are internal quantum-accessible random oracles. Here the adversary also has classical access to

Figure 26: Construction of INIT, INIT, F_s, and Repro_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 × 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 G_7 and G_s in the proof of Theorem 5.3, we have

\[ |\Pr[\text{Bad}_6] - \Pr[\text{Bad}_7]| \leq \sum_{k = 1}^{n_{\text{op}}} \sum_{i = 0}^{k} 2^{2k} \sqrt{\Pr[(r_i, (r', c')) \leftarrow B_i^{(G \times H)} \text{s.t. } (r_i, (r', c')) \in S_i : G_7^{S_i}]} + \frac{2\mu q}{\sqrt{|R|}}. \]

Based on B_k (1 ≤ k ≤ n_{\text{op}}), we can construct an OW-CPA adversary B_k^{ow} (in Figure 27) against PKE that has the same structure with the adversaries in Figure 9, except that B_k^{ow} needs to simulate the decryption oracle. Since in G_7, the decryption oracle can be simulated without secret key, B_k^{ow} can simulates G_s for B_k perfectly if Bad_7 does not occur. Therefore, similar to the argument in Theorem 4.1, there exists an OW-CPA adversary B^{ow} such that

\[ |\Pr[G_\delta \Rightarrow 1]_{\text{Bad}_6} - \Pr[G_\delta \Rightarrow 1]_{\text{Bad}_7}| \leq 2(n_{\text{op}} + 1)^2 q \sqrt{2\mu \text{Adv}_{\text{OW-CPA}}} + \Pr[\text{Bad}_7] + \frac{2\mu q}{\sqrt{|R|}}. \]

G_7 also enable us to construct an otSUF-CMA reduction to bound Pr[Bad_7]. Let F be an otSUF-CMA adversary that simulates G_s for A. F has similar structure with the one in Figure 21. It firstly chooses t^* \sim [\mu] uniformly and sets K_1^{\text{mac}} \leftarrow \bot. To generates \tau_*, F queries its TAG oracle on d_\tau* and sets the responding tag as \tau_* . F aborts the game if A opens \epsilon_\tau. When A queries the DEC oracle on
\[ \mathcal{B}_k^{ow}(pk^*, e^*) \text{ is an OW-CPA challenge of PKE} \]

1. \((\langle s, in_0 \rangle, (G \times H), (G_0 \times H_0)) \leftarrow \text{INIT} \) // INIT is defined in Figure 26, where \(pk := pk^*\)

2. \text{parse} \((\mathcal{M}_a, m, r, R, e, K_{OW}^*) := s, c := in_0\)

3. \(t^* \leftarrow [\mu], (e_{t^*}, d_{t^*}, \tau_{t^*}) := c[t^*]\)

4. \(c[t^*] := (e_{t^*}, d_{t^*}, \tau_{t^*})\), \(in_0 := c\) // embed the challenge

5. \(\text{if} \ k = 0: \text{goto line 15}\)

6. \(out_0 \leftarrow B_{sPKE(G_0 \times H_0), Dec}(in_0)\) // Dec is simulated as in \(G_s\) in Figure 25

7. \(\text{if} \ out_0 = t^*: \text{abort}\)

8. \((in_1, in_1^*) := F_a(out_0, (G_0 \times H_0))\) // \(F_a\) is defined in Figure 26

9. \((G_1 \times H_1) := \text{Repro}_a(in_1, (G_0 \times H_0))\) // \(\text{Repro}_a\) is defined in Figure 26

10. \(\text{for} \ j = 1 \text{ to } k - 1;\)

11. \(\text{out}_j \leftarrow B_{sPKE(G_j \times H_j), Dec}(in_j)\)

12. \(\text{if} \ out_j = t^*: \text{abort}\)

13. \((in_{j+1}, in_{j+1}^*) := F_a(out_j, (G_j \times H_j))\)

14. \((G_{j+1} \times H_{j+1}) := \text{Repro}_a(in_{j+1}, (G_j \times H_j))\)

15. \((r_0, (r_1, e^*)) \leftarrow B_{sPKE(G_k \times H_k), Dec}(in_k)\) // perform measurement

16. \(b \leftarrow \langle 0, 1 \rangle, r^* := r_k\) // randomly choose a solution

17. \text{return} \(r^*\)

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

input \((e, d, \tau)\) that \(e = e_{t^*}\) and \(VRFY(d, \tau) = 1\), \(\mathcal{F}\) outputs \((d, \tau)\). If \(\mathcal{A}_1\) finally outputs \(out\) but the event \(\mathcal{B}_7\) does not occur, then \(\mathcal{F}\) aborts. If the event \(\mathcal{B}_7\) occurs, then the probability that the ciphertext \((e, d, \tau)\) (that raises this event) satisfies \(e = e_{t^*}\) is \(\frac{1}{\mu - n}\). The probability that \(\mathcal{A}_1\) does not open \(e_{t^*}\) is \(\frac{\mu - n}{\mu}\), so we have

\[
\Pr[\mathcal{B}_7] = \mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F})
\]

and thus we have

\[
\left| \Pr[G_{t}^A = 1] - \Pr[G_{t}^A = 1] \right| \\
\leq 4(n_{OP} + 1)q \sqrt{2 \mu \text{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{B}_{ow}) + \mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F})} + \frac{4\mu q}{\sqrt{|R^t|}} + \mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F})
\]

Now we can construct a simulator \(\mathcal{S}\) that is simulating \(G_{t}\) for \(\mathcal{A}\) and interacts with the IDEAL-SO-CCA^S_{\text{PKE}} 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 \(G_{t}\) (without using \(sk\)), so we have

\[
\left| \Pr[G_{t}^A \Rightarrow 1] - \Pr[\text{IDEAL-SO-CCA}^{S}_{\text{PKE}} \Rightarrow 1] \right| \leq \mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F})
\]

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

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

\[
\leq 3\mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F}) + \left(6(n_{OP} + 1)^2q \sqrt{2 \mu \text{Adv}^{\text{OW-CPA}}_{\text{PKE}}(\mathcal{B}_{ow}) + \mu \text{Adv}^{\text{SUF-CMA}}_{\text{MAC}}(\mathcal{F})} + \frac{2\mu q}{\sqrt{|R^t|}} + 16(\mu + n_{Dec} + qG + qH + 1)^2\delta + \frac{\mu^2}{|K_{MAC}|} + \frac{6\mu q}{\sqrt{|C^t|}} + \frac{\mu n_{Dec}}{|M^t|} + \frac{(2 + \mu)q}{\sqrt{|M^t|}}\right)
\]

\(\square\)
This introduce collision bounds $pk_n$ such ciphertexts is to simulate $\text{Dec}$. To make the simulation consistent, if $A$ queries $(K, K^{\text{acc}}) := (K_i, R_i^{\text{acc}})$ then $\text{Verf}(K^{\text{acc}}, \tau) = 1$.

Moreover, we also assume that there is no collision among outputs of $G$. Similar to the proof of Theorem 3.1, if the decryption oracle can be simulated without secret key. However, in the Bi-SO setting, we cannot

<table>
<thead>
<tr>
<th>$i$</th>
<th>$j$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
</tr>
<tr>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
<td>35</td>
</tr>
<tr>
<td>36</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>39</td>
<td>40</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>43</td>
<td>44</td>
</tr>
<tr>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>51</td>
<td>52</td>
<td>53</td>
</tr>
<tr>
<td>54</td>
<td>55</td>
<td>56</td>
</tr>
<tr>
<td>57</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
</tr>
<tr>
<td>63</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>66</td>
<td>67</td>
<td>68</td>
</tr>
<tr>
<td>69</td>
<td>70</td>
<td>71</td>
</tr>
</tbody>
</table>

Figure 28: Simulator $S$ in the proof of Theorem 4.2.

## G Bi-SO Security Proof of FO$^\mathcal{L}$

### 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 $G \times H(pk', pk', r, e) := (G(pk', r), H(pk, r, e))$. $A$ can query $G \times H$ at most $q = q_H + q_G$ times.

Let $g, g_{pk_1}, \ldots, g_{pk_j} h, h_{pk_1}, \ldots, h_{pk_j}, h', h'_{pk_1}, \ldots, h'_{pk_j}, h_{pk_1}, \ldots, h_{pk_j}$ 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 $A_0$ queries DEC on $(j, e, d, \tau)$ such that $(e, d) = (e_{j, i}, d_{j, i})$ before seeing the challenge ciphertexts $c$. The probability that $A_0$ queries DEC on such ciphertexts is $\frac{2^{p_{\text{min}}}}{n_{\text{out}}} + \frac{2^{p_{\text{max}}}}{\sqrt{\mid M \mid}}$. For simplicity, we just add this probability into our final bound. Moreover, we also assume that there is no collision among outputs of $k_{j, i}, r_{j, i}, s$, $R_{j, i}$, $K_{j, i}$, and $K_{j, i}^{\text{acc}}$. This introduce collision bounds $\frac{2^p}{\sqrt{\mid M \mid}} \cdot \frac{2^p}{\sqrt{\mid M \mid}} + \frac{2^p}{\sqrt{\mid M \mid}} \cdot \frac{2^p}{\sqrt{\mid M \mid}} + \frac{2^p}{\sqrt{\mid M \mid}} \cdot \frac{2^p}{\sqrt{\mid M \mid}} + \frac{2^p}{\sqrt{\mid M \mid}}$ 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

$$\Pr[G_0^{\mathcal{A}} \Rightarrow 1] = \Pr[\text{REAL-Bi-SO-CCA}^{\mathcal{A}_\text{PKE}} \Rightarrow 1]$$

**Game G_1:** If $A$ queries $\text{Dec}(j, (e, d, \tau))$ where $e$ is invalid or cannot pass the re-encryption check, then the oracle computes $(K, K^{\text{acc}})$ as $h'_{pk_j}(e)$ instead of $H'(pk_j, k_j, e)$ (see Item 48). Moreover, to make the simulation consistent, if $A$ queries $H'(pk_j, k_j, e)$ where $pk_j$ is corrupted and $e$ is an invalid ciphertext or cannot pass the re-encryption check, then $H'(pk, k, e)$ returns $h'_{pk_j}(e)$ instead of $H'(pk, k, e)$ (see Items 26 to 30). The latter modification can be seen as we reprogram $H'[pk_j, k_j, e] \rightarrow h'_{pk_j}(e)$ for
some $e$ if $A$ 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 $A$ into $n_{CO} + 1$ stages with respect to its CORRUPT queries. In $G_0$, the QRO $H'$ does not change during the game, while in $G_1$, corrupting user will make $H'$ reprogrammed.

We can define algorithms INT, $F_s$, and Repro$_s$ in a natural way such that $G_0$ is a NONADA game and $G_1$ is an ADA game and construct an adversary $B_k$ to bound the probability that $A$ "queries" $k_j$ where user $j$ is uncorrupted. Since for all $j, k_j$ are chosen at independently uniformly random, thus the probability that $B_k$ find $k_j$ for any $j \notin J$ is $\frac{pq_{H'}}{|M'|}$, where $q_{H'}$ is the number of time that $A$ queries $H'$. For simplicity, we ignore the details. By Lemma 2.7 and Lemma 3.1, we have

$$\left| \Pr[G_0^A \Rightarrow 1] - \Pr[G_1^A \Rightarrow 1] \right| \leq \frac{2(n_{CO} + 1)^2 \sqrt{pq_{H'}} + 2pq_{H'}}{\sqrt{|M'|}}$$

**Game $G_2$:** We restrict the range of $G(pk_j, \cdot)$ to be the "good" randomness space $\mathcal{R}'_{good} \subset \mathcal{R}'$ respect to $(pk_j, sk_j)$. Specifically, let $\mathcal{R}'_{good}(pk_j, sk_j, r)$ be the set $\{r' \in \mathcal{R}' | Deco(sk_j, Enc_0(pk_j, r'; r')) = r\}$ and $\mathcal{R}'_{Bad}(pk_j, sk_j, r)$ be the set $\mathcal{R}' \setminus \mathcal{R}'_{good}(pk_j, sk_j, r)$. Let $g'_pk : \mathcal{M}' \rightarrow \mathcal{R}'$ be a quantum-accessible random oracle such that $g'_pk_j(r)$ is sampled uniformly from $\mathcal{R}'_{good}(pk_j, sk_j, r)$. 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 $\text{PKE}$ is $\delta$-correct (see Definition 2.1), then

$$\left| \Pr[G_4^A \Rightarrow 1] - \Pr[G_4^A \Rightarrow 1] \right| \leq 8\rho(\mu + n_{\text{Dec}} + \gamma G + \gamma H + \gamma H' + 1)^2 \delta.$$  

Game $G_3$: We set $H(pk_j, r, e) = h_{pk_j}(e)$ if $e = \text{Enc}_0(pk_j, r; G(pk_j, r))$ (see Items 22 to 24). Since the randomness generated by $G$ (i.e., $g_{pk_j}$) is always a “good” randomness, the map $\text{Enc}_0(pk_j, :) \rightarrow G(pk_j, :) - \text{injective}$ and thus the map $h_{pk_j}(\text{Enc}_0(pk_j, :) : G(pk_j, :))$ be also viewed as an random oracle. Therefore, we have

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

Game $G_4$: 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 $A$ to $h_{pk_j}$ and to $h_{pk_j}'$ in $G_3$ (regardless of whether user $j$ is corrupted) are disjoint, and so we can use the same internal QRO to respond these queries. We have

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

Game $G_5$: In this game, the oracle $G(pk, :)$ is simulated using $g_{pk}$ instead of $g_{pk}'$. Similar to the difference between $G_3$ and $G_2$, we have

$$\left| \Pr[G_4^A \Rightarrow 1] - \Pr[G_5^A \Rightarrow 1] \right| \leq 8\rho(\mu + n_{\text{Dec}} + \gamma G + \gamma H + \gamma H' + 1)^2 \delta.$$  

Game $G_6$: In this game, the decryption oracle always returns $\bot$ if the adversary queries a ciphertext $(e, d, \tau)$ that $e$ is the PKE part of some unopened ciphertexts, i.e., $\exists(j, i) \in [p] \times [\mu] \setminus (J' \cup I)$ such that $e = e_{j,i}$.

Let $\text{Bad}_k$ be the event that $A$ queries $\text{Dec}$ on a ciphertext $(e, d, \tau)$ that $\exists(j, i) \in [p] \times [\mu] \setminus (J' \cup I)$ s.t. $e = e_{j,i}$ and $\text{Vrfy}(g_{pk}_j, d, \tau) = 1$ in $G_6(k \geq 6)$. That is, $A$ forges valid MAC codes of some unopened ciphertext. Similar to the arguments in the proof of Theorem 4.2, if $\text{Bad}_k$ does not occur, then the winning probabilities of $A$ in $G_5$ and in $G_6$ are the same. Thus $\Pr[G_5^A \Rightarrow 1] - \Pr[G_6^A \Rightarrow 1] \leq \Pr[\text{Bad}_6]$.  

Game $G_7$: The game simulator generates challenge ciphertexts independent of $G, H$. If $A$ 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 $G(pk_j, r_{j,i}) = R_{j,i}$ and $H(pk_j, r_{j,i}, c_{j,i}) = d_{j,i} \oplus m_{j,i}$, and the responds of $\text{Dec}$ are modified to make the simulation consistent. See Items 19 to 21 and Items 56 to 58.

In $G_7$, there are two types of query (OPEN and CORRUPT) from $A$ will make $G$ and $H$ be reprogrammed. In Figure 30 we define INIT, $F_k$, and $\text{Repro}_k$ such that $G_6$ is a NONADA game and $G_7$ is an ADA game in Figure 4. The queries to $H'$ and $\text{Dec}$ do not influence the distribution of $G \times H$.

We firstly view $A = (A_0, A_1)$, where $A_0$ is $A$ in the stage cannot issues OPEN or CORRUPT queries and $A_1$ is $A$ in the stage that can issues OPEN and CORRUPT queries. Suppose that $A$ corrupts at most $n_{\text{CO}}$ users and opens at most $n_{\text{OR}}$ challenge ciphertexts. $A_0$ can be further divided into $(n_{\text{CO}} + n_{\text{OR}} + 1)$ stages wrt OPEN queries or CORRUPT queries: Before any OPEN or CORRUPT query (i.e., at the 0-th stage), $A_1$ takes $n_0 := c$ as input and outputs $\text{out}_0$, where $\text{out}_0$ 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 $\leq k \leq n_{\text{CO}} + n_{\text{OR}} - 1$), $A_1$ receives $n_k$ and ends the stage by outputting $\text{out}_k$, where $n_k$ is the secret key of a user or the message-randomness pair of a opened ciphertext, and $\text{out}_k$ is an opening index or a corrupting index. Finally, at the $(n_{\text{CO}} + n_{\text{OR}})$-th stage, $A_1$ receives $n_n$ and terminates by outputing out.

Moreover, at $A_1$'s $k$-th stage, our $\text{Repro}_A$ defines a set

$$S_k := \{(pk_j, r_{j,i}), (pk_{j'}, r_{j',i'}, e_{j',i'}) | (j, i) \in [p] \times [\mu] \setminus (J'_k \cup I_k)$$

$$\text{or} \quad (j', i') \in [p] \times [\mu] \setminus (J'_k \cup I_k)\}, \quad (21)$$

where $J'_k, I_k$ are the lists $J', I$ respectively just after $A_1$ issues the $k$-th query to OPEN or CORRUPT.

Let $q_k$ be the number of query to $G \times H$ issued by $A_1$ at its $k$-th stage. By Lemma 3.1, there exists $B_k$ for $k \in \{0, \ldots, n_{\text{CO}} + n_{\text{OR}}\}$ such that:

$$\left| \Pr[G_7^A \Rightarrow 1] - \Pr[\text{Bad}_7] \right| = \left| \Pr[G_7^A \Rightarrow 1] - \Pr[\text{Bad}_7] \right|$$

45
Figure 30: Constructions of \textsc{Init}, \textsc{F}, and \textsc{Repro}_a in the proof of Theorem 6.1. We assume that all pk_j's, k_j's, d_j,i's, K_{j,i}^{ac}'s are different. I, J, J' are publicly accessible.
$\sum_{k=0}^{n_{CO}+n_{OP}} \sum_{i=0}^{k} 2q_{i} \sqrt{\Pr \left[ x' \leftrightarrow B_{i}^{G \times H_{i}, [H'], \text{Dec}} \text{ s.t. } x' \in S_{i} : G_{k}^{S_{i}} \right] + \frac{2\mu q}{\sqrt{|R'|}}}$

where sets $S_{k}$ is defined as Equation (21). Based on $B_{k}$, we can construct adversaries $B_{k}^{ow}$ against the OW-CPA security of PKE. See Figure 31. By the construction of $B_{k}^{ow}$, as long as $A$ does not corrupt $j^{*}$, $B_{k}^{ow}$ can always simulate Dec and $G_{8}$.

Figure 31: The detailed constructions of OW-CPA adversaries $B_{k}^{ow}, 0 \leq k \leq n_{CO} + n_{OP}$.

By the construction of $B_{k}^{ow}$, if $A_{j}$ does not corrupt user $j^{*}$ and does not open $c_{j}^{*}, i^{*}$, and $r$ or $r'$ equals the solutions of $e^{*}$, then $B_{k}^{ow}$ wins. So the winning probability for $B_{k}^{ow}$ to breaks the OW-CPA challenge is:

$$\text{Adv}_{\text{OW-CPA}}^{\text{PKE}}(B_{k}^{ow}) = \frac{1}{2^{k}} \Pr \left[ x \leftrightarrow B_{k}^{G \times H_{i}, [H'], \text{Dec}} \text{ s.t. } x \in S_{k} : G_{k}^{S_{k}} \right]$$

and thus we have

$$\left| \Pr [G_{6}^{A} \Rightarrow 1] - \Pr [G_{7}^{A} \Rightarrow 1] \right| \leq \sum_{k=0}^{n_{CO}+n_{OP}} \sum_{i=0}^{k} 2q_{i} \sqrt{2^{k} \Pr \left[ \text{Adv}_{\text{OW-CPA}}^{\text{PKE}}(B_{k}^{ow}) + \Pr [\text{Bad}_{7}] \right] + \frac{2\mu q}{\sqrt{|R'|}}}$

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

$$\left| \Pr [G_{6}^{A} \Rightarrow 1] - \Pr [G_{7}^{A} \Rightarrow 1] \right| \leq 2(n_{CO} + n_{OP} + 1)^{2} q \sqrt{2^{k} \Pr \left[ \text{Adv}_{\text{OW-CPA}}^{\text{PKE}}(B^{ow}) + \Pr [\text{Bad}_{7}] \right] + \frac{2\mu q}{\sqrt{|R'|}}}$

Now we can construct an otSUF-CMA reduction to bound $\Pr [\text{Bad}_{7}]$. Let $F$ be an otSUF-CMA adversary that simulates $G_{8}$ for $A$. It firstly chooses $j^{*} \in [p], i^{*} \in [\mu]$ uniformly and sets $K_{8}^{mac, \ast} := \bot$. To generates $\tau_{j^{*}, i^{*}}$, $F$ queries its tag oracle on $d_{j^{*}, i^{*}}$ and sets the responding tag as $\tau_{j^{*}, i^{*}}$. $F$ aborts the game if $A$ opens $c_{j^{*}, i^{*}}$ or corrupts $j^{*}$. When $A_{1}$ queries the Dec oracle on input $(pk_{j}, c, d, \tau)$ that $c = c_{j^{*}, i^{*}}$ and $\text{MAFy}(d, \tau) = 1$, $F$ outputs $(d, \tau)$. If $A_{1}$ finally outputs out but the event $\text{Bad}_{7}$ does not occur, then $F$ aborts. Similar to the argument in the proof of Theorem 6.1, we have

$$\Pr [\text{Bad}_{7}] = p_{\mu} \text{Adv}_{\text{otSUF-CMA}}^{\text{MAC}}(F)$$

Since the events $\text{Bad}_{d_{j}}$ and $\text{Bad}_{d_{\tau}}$ 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 $| \Pr [G_{6}^{A} \Rightarrow 1] - \Pr [G_{7}^{A} \Rightarrow 1] |$.
Therefore, there exists adversaries $\mathcal{B}^{ow}$ and $\mathcal{F}$ such that
\[
\left| \Pr[\text{Bad}_B] - \Pr[\text{Bad}_F]\right| \\
\leq 2(n_c + n_{ov} + 1)^2 q \sqrt{2p\mu \text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(B^{ow}) + p\mu \text{Adv}_{\text{MAC}}^{\text{SUf-CMA}}(F) + \frac{2p\mu q}{\sqrt{|R|^2}}},
\]
and thus we have
\[
\left| \Pr[G_0^4 \Rightarrow 1] - \Pr[G_0^4 \Rightarrow 1]\right| \\
\leq 4(n_c + n_{ov} + 1)^2 q \sqrt{2p\mu \text{Adv}_{\text{PKE}}^{\text{OW-CPA}}(B^{ow}) + p\mu \text{Adv}_{\text{MAC}}^{\text{SUf-CMA}}(F)} + p\mu \text{Adv}_{\text{MAC}}^{\text{SUf-CMA}}(F) + \frac{4p\mu q}{\sqrt{|R|^2}}}
\]

Now we can construct a simulator $\mathcal{S}$ that interacts with the $\text{IDEAL-SO-CCAS}_{\text{PKE}_B}^{\text{S}}$ game and simulates $\mathcal{G}_7$ for $\mathcal{A}$. $\mathcal{S}$’s construction is shown in Figure 32, and it perfectly simulates $\mathcal{G}_7$ for $\mathcal{A}$ except that $\text{Bad}_7$ happens, and thus
\[
\left| \Pr[G_7^4 \Rightarrow 1] - \Pr[\text{IDEAL-SO-CCAS}_{\text{PKE}_B}^{\text{S}} \Rightarrow 1]\right| \leq p\mu \text{Adv}_{\text{MAC}}^{\text{SUf-CMA}}(F).
\]

Therefore,
\[
\text{Adv}_{\text{PKE}_B}^{\text{BI-SO-CCAS}}(A, \mathcal{S}, p, \mu, \text{Rel})
\]
H Bi-SO Security Proof of $U_m^L$

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 $\mathcal{D}_M$. Hence, we consider $\mathcal{D}_M$-OW-CPA security, namely, OW-CPA security with challenge messages chosen following $\mathcal{D}_M$. We require that $\mathcal{D}_M$ itself has high minimum entropy. Otherwise, $\mathcal{D}_M$-OW-CPA security can be broken trivially.

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

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

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

**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 $h, h_{pk_1}, ..., h_{pk_y}, h', h'_{pk_1}, ..., h'_{pk_y}$ 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 game transition, we consider the case that $\mathcal{A}_0$ queries DEC on $(j, e, d, \tau)$ such that $(e, d) = (e_{j,i}, d_{j,i})$ before seeing the challenge ciphertexts $c$. The probability that $\mathcal{A}_0$ queries DEC on such ciphertexts is $\frac{\mu_{\text{prDec}}}{|\mathcal{C}|^\omega_{\text{prDec}}} + \frac{\mu_g}{\sqrt{|\mathcal{M}|}}$. 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 $K_i^{\text{acc}}$'s. This introduce collision bounds $\frac{\mu_{\text{prDec}}^2 + \mu_g^2}{|\mathcal{C}|^\omega_{\text{prDec}}} + \frac{\mu_g^2}{|\mathcal{M}|} + \mu_g \eta_{\mathcal{KG}}$. For simplicity, we just add these probability into our final bound. The games sequence is given in Figure 33. We have

$$\Pr[G_0^A \Rightarrow 1] = \Pr[\text{REAL-Bi-SO-CCA}_{\text{PKE}}^A \Rightarrow 1]$$

**Game 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 $(K, K^{\text{acc}})$ as $h'_{pk_j}(e)$ instead of $H'(pk_j, k_j, e)$. Moreover, to make the simulation consistent, if $\mathcal{A}$ queries $H'(pk_j, k_j, e)$ where $pk_j$ is corrupted and $e$ is an invalid ciphertext, then $H'(pk, k, e)$ returns $h'_{pk_j}(e)$ instead of $h'(pk, k, e)$. The latter modification can be seen as we reprogram $H'([pk_j, k_j, e] \rightarrow h'_{pk_j}(e))$ for some $e$ if $\mathcal{A}$ corrects 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

$$\Pr[G_0^A \Rightarrow 1] - \Pr[G_1^A \Rightarrow 1] \leq \frac{2(n_{\text{co}} + 1)^2 \sqrt{\mu_{\text{prDec}}} + 2p \mu_{\text{prDec}}}{\sqrt{|\mathcal{M}|}}$$

**Game G_2:** We set $H(pk_j, r) = h_{pk_j}(\text{Enc}_0(pk_j, r))$. Since PKE is rigid correct, the map $\text{Enc}_0(pk_j, \cdot)$ is injective and thus $h_{pk_j}(\text{Enc}_0(pk_j, \cdot))$ can be also viewed as a random oracle. Therefore, we have

$$\Pr[G_1^A \Rightarrow 1] = \Pr[G_2^A \Rightarrow 1].$$
 Game $G_5$: 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 $A$’s view since all indirect queries from $A$ to $h_{pk_j}$ and to $h'_{pk_j}$ in $G_2$ (regardless of whether user $j$ is corrupted) are disjoint. We have

$$\Pr[|G_2^d| \Rightarrow 1] = \Pr[|G_3^d| \Rightarrow 1].$$

 Game $G_4$: In this game, the decryption oracle always returns $\perp$ if the adversary queries a ciphertext $(e, d, \tau)$ that $e$ is the PKE part of some unopened ciphertexts, i.e., $\exists (j, i) \in [p] \times [\mu \setminus (J' \cup I)], e = e_{j,i}$. Let $Bad_4$ be the event that $A$ queries $Dec$ on a ciphertext $(e, d, \tau)$ that $\exists (j, i) \in [p] \times [\mu \setminus (J' \cup I)]$ s.t. $e = e_{j,i}$ and $Verf(K_{j,i}^{mac}, d, \tau) = 1$ in $G_4(k \geq 4)$. Similar to the arguments in the proof of Theorem 4.2, if $Bad_4$ does not occur, then the winning probabilities of $A$ in $G_3$ and in $G_4$ are the same. Thus $|\Pr[|G_3^d| \Rightarrow 1] - \Pr[|G_4^d| \Rightarrow 1]| \leq \Pr[|Bad_4|]$.  

 Game $G_5$: The game simulator generates challenge ciphertexts independent of $H$. If $A$ corrupted the user $j$ (which means that all $e_{j,i}$ for $i \in [\mu]$ are opened), or opened $e_{j,i}$, then $H$ will be reprogrammed such that $H(pk_j, r_{j,i}) = (d_{j,i} \oplus m_{j,i}, K_{j,i}^{mac})$, and the responds of $Dec$ are modified to make the simulation consistent.

Similar to the proof of Theorem 6.1, there are two types of query (Open and Corrupt) from $A$ will make $H$ be reprogrammed. Lemma 3.1 still can be used here. One can define algorithms INIT, $F_4$, and $Repr_4$ such that $G_4$ is a NONADA game and $G_4$ is an 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

Figure 33: Games $G_0$-$G_6$ in the proof of Theorem 6.2
need the QRO $G_i$, for sake of simplicity, we ignore the details here, and only give a final bound.

$$\Pr[G^A_4 \Rightarrow 1 | \neg \text{Bad}_i] - \Pr[G^A_5 \Rightarrow 1 | \neg \text{Bad}_i]$$

$$\leq \sum_{k=0}^{n_{D_M} + \epsilon_{D_M}} k \cdot 2q_i \sqrt{\Pr[x' \leftarrow B^i_{G \times H'}, \text{Dec} \text{ s.t. } x' \in S_i : G^S_{5_i}]} + \frac{2p \mu q}{2^{2^{r'}}}$$

Here $\epsilon_{D_M}$ is the minimal entropy of the distribution $D_M$. As we did in the proof of Theorem 6.1, here we can also construct $B_{\text{OW}}$ and $F$ such that

$$\Pr[G^A_4 \Rightarrow 1] - \Pr[G^A_5 \Rightarrow 1]$$

$$\leq \mu \text{Adv}_{\text{MAC}}^{\text{OW}-\text{CMA}}(F) + \frac{4p \mu q}{2^{2^{r'}}} + 4(n_{\text{CO}} + n_{\text{OP}} + 1)2q_i \sqrt{2p \mu \text{Adv}_{\text{PKE}, D_M}^{\text{OW}-\text{CPA}}(B_{\text{OW}}) + \mu \text{Adv}_{\text{MAC}}^{\text{SUF}-\text{CMA}}(F)}$$

Now we can construct a simulator $S$ that interacts with the IDEAL-SO-CCA\textsuperscript{S}_{\text{PKE}} game and simulates $G_5$ for $A$. $S$'s construction is shown in Figure 32, and it perfectly simulates $G_5$ for $A$, and thus $\Pr[G^A_4 \Rightarrow 1] - \Pr[\text{IDEAL-SO-CCA}_{\text{PKE}}^S \Rightarrow 1] \leq \mu \text{Adv}_{\text{MAC}}^{\text{OW}-\text{CMA}}(F)$. Therefore,

$$\text{Adv}_{\text{PKE}}^{\text{SUF-CCA}}(A, S, p, \mu, \lambda)$$

$$\leq |\Pr[\text{REAL-SO-CCA}_{\text{PKE}}^A \Rightarrow 1] - \Pr[\text{IDEAL-SO-CCA}_{\text{PKE}}^S \Rightarrow 1]|$$

$$\leq \frac{p \mu \text{Adv}_{\text{MAC}}^{\text{OW}-\text{CMA}}(F) + p^2 \mu^2}{|C|} + \frac{p^2 \mu^2}{|M|} + \frac{p^2 \mu^2}{|K_{\text{MAC}}|} + p \mu |\text{CO}| + \frac{2p \mu q}{2^{2^{r'}}}$$
\[ \begin{align*}
+ \frac{2(n_{CO} + 1)^2 \sqrt{pqh^2 + 2pqh' + pqq}}{|M'|} &+ 3p\mu\text{Adv}^{\text{sUF-CMA}}_{\text{MAC}}(F) \\
+ 6(n_{CO} + n_{OP} + 1)^2 q \sqrt{2p\mu\text{Adv}^{\text{OW-CPA}}_{\text{PKE,D}_{\text{w}}}(B^{\text{ow}})} &+ p\mu\text{Adv}^{\text{sUF-CMA}}_{\text{MAC}}(F)
\end{align*} \]