# Cryptomania v.s. Minicrypt in a Quantum World

Longcheng Li* Qian Li† Xingjian Li‡ Qipeng Liu§

**Abstract**

We prove that it is impossible to construct perfect-complete quantum public-key encryption (QPKE) with classical keys from quantumly secure one-way functions (OWFs) in a black-box manner, resolving a long-standing open question in quantum cryptography.

Specifically, in the quantum random oracle model (QROM), no perfect-complete QPKE scheme with classical keys, and classical/quantum ciphertext can be secure. This improves the previous works which require either unproven conjectures or imposed restrictions on key generation algorithms. This impossibility extends to QPKE with quantum public key in natural settings, which is tight to all known QPKE constructions with quantum public key.

## 1 Introduction

Quantum information and computation are topics with growing importance in cryptography. They reshape people's views on cryptography drastically, including breaking classical secure cryptosystems [Sho99], creating primitives that are impossible for classical [Wie83], and weakening assumptions [BB14]. However, quantum cryptography is not an all-powerful tool, as it also has its own limits. Therefore, characterizing the boundary of quantum cryptography under different assumptions has become a topic of great interest.

Boundaries between classical cryptographic primitives have already been studied extensively. In the seminal work by Impagliazzo and Rudich [IR89], they proposed the methodology of black-box separation. They showed that one-way functions are insufficient to build public key encryption (PKE) schemes in a black-box manner. In the famous work by Impagliazzo [Imp95], he characterized five possibilities on the hardness of NP problems and their complexity consequences. Among them, he used the word 'Minicrypt' to refer to a world where one-way functions exist, and the word 'Cryptomania' for a world with public key cryptography primitives. Thus, the separation result by Impagliazzo and Rudich can be viewed as a separation between Minicrypt and Cryptomania.

It turns out that the landscape of quantum cryptography varies depending on the definition of Minicrypt. For example, it is known that with quantum communication, many primitives in classical Cryptomania can be built from one-way functions, including key agreement [BB14], oblivious transfer [GLSV21, BCKM21], public key encryption (with quantum public keys) [Col23, MW24, KMNY24, BGH+23]. On the other hand, none of these constructions are known in the

*University of Cambridge. Email: lilongcheng116@gmail.com

†Shenzhen International Center For Industrial And Applied Mathematics, Shenzhen Research Institute of Big Data. Email: liqian.ict@gmail.com

‡Tsinghua University. Email: lxj22@mails.tsinghua.edu.cn

§University of California San Diego. Email: qipengliu0@gmail.com

quantum computation classical communication (QCCC) setting. Quantum communication has various drawbacks in practice, including the difficulties of authenticating, broadcasting, reusability, and potentially adding interactions. In this work, we focus on the fundamental question regarding Minicrypt and Cryptomania in a quantum world [HY24]:

*Does there exist any separation between Minicrypt and Cryptomania in the QCCC setting?*

**Previous works.** Several works have attempted to address this question, but classical proof techniques often fail due to fundamental differences between quantum and classical algorithms/information, including challenges related to cloning, rewinding, and the unique structure of quantum queries. As a result, all previous approaches have either relied on unproven conjectures or applied only to highly restricted QPKE schemes.

In the work by Austrin et al. [ACC+22], they initialized the study of separations between quantum key agreements and one-way functions in the quantum random oracle model (QROM). They showed that under some *conjecture named 'polynomial compatibility conjecture'*, quantum key agreements with perfect correctness don't exist. Since quantum PKE implies a two-round key agreement scheme, their result also implies a separation between quantum PKE and one-way functions. The same idea was also applied in separating PKE with quantum ciphertext and one-way functions [BGVV23], which we will discuss later. To show a separation between key agreement and one-way function in the QROM, one needs to construct an eavesdropper that breaks the security of the key agreement by making polynomially many queries to the oracle. In the paper [ACC+22], the authors construct an eavesdropper who only makes classical queries to the oracle, while Alice and Bob can make quantum queries in general. Because honest parties get quantum queries and the eavesdropper only classical queries, the conjecture may be too strong.

In another line of work, Li, Li, Li, and Liu [LLLL24] approach the problem from a different perspective. They introduce tools from quantum Markov chains [FR15] to construct an eavesdropper for quantum public key encryption schemes that have a *classical key generation process*. On a high level, they construct an Eve who uses a polynomial number of quantum queries and creates a simulated view for Alice called $\mathsf{A}'$, denoting the view of Alice before she tries to decrypt a ciphertext using the random oracle. Using the quantum Markov chain, they can argue that the state over the real Alice and Bob $\mathsf{AB}$ is statistically close to that over the simulated Alice and Bob $\mathsf{A}'\mathsf{B}$. Finally, they finish running the simulated Alice with one caveat: the simulated Alice may not be consistent with the real random oracle, and they have to find an appropriate oracle to complete the execution. The work [LLLL24] solves this issue by only focusing on the key generation algorithm which makes only classical queries. In a followup paper [LLLL25], the authors introduce a view from boolean function analysis as an attempt to attack PKE with quantum-query key generation. They made some conjecture about the zero point distributions of low-degree polynomials and proved that the conjecture implies a separation between perfect-complete PKE and OWF in the QCCC model.

Both works fail to establish the separation once the key-generation algorithm is allowed to quantumly query a random oracle. More broadly, lifting separations proved in the classical-accessible-oracle model to the quantum-accessible-oracle model can be notoriously difficult. While several landmark separations are known with classically accessible oracles — e.g., quantum money v.s. one-way functions [AHY23] and QCMA v.s. QMA [LLPY23, LMY25] — whether these separations persist when the oracle admits quantum queries still remains a major open problem in quantum complexity and quantum cryptography; either these separation results completely fail when quantum access is provided, or they have to rely on unproven conjectures.

| | [ACC+22] | [LLLL24] | [LLLL25] | This Work (Theorem 1.1) |
|---|---|---|---|---|
| Gen | Q | C | Q | Q |
| Enc | Q | Q | Q | Q |
| Dec | Q | Q | Q | Q |
| conjecture | ✓ | | ✓ | |

Table 1: Comparing all impossibility results on QPKE with prefect completeness in the QROM. 'Q' denotes quantum, 'C' denotes classical.

### 1.1 Our Main Result

In this work, we refine the separation framework of [LLLL24, LLLL25] and present a unified approach to proving lower bounds and separations for multiparty-computation related primitives in the QROM. We hope the improved framework can be used for achieving lower bounds for other MPC related protocols with prefect completeness.

Using that, we obtain a full separation between perfectly complete quantum PKE and one-way functions, i.e. the quantum version of [IR89] for public-key encryption, closing the gap in previous works.

**Theorem 1.1** (Informal)**.** *Perfect-complete quantum public key encryption, with classical keys and classical ciphertext, does not exist in the quantum random oracle model.*

**Remark 1.2.** *Our result removes the conjecture used in [ACC+22, LLLL25] and the restriction of a classical key generation algorithm in [LLLL24], leaving perfect completeness as the only requirement. Perfect completeness is a natural requirement satisfied by many cryptographic schemes, both classical and quantum, including all known quantum PKE schemes even with quantum keys [Col23, MW24, KMNY24, BGH+23]. Therefore, focusing on the perfect-complete setting does not impose a strong restriction. We conjecture that allowing non-perfectness does not change the impossibility result; we hope that the separation framework proposed in this work is helpful for removing the last restriction on QPKE.*

**Result 2: Extending to quantum ciphertext.** Using a quantum public key can cause challenges in public-key distribution, authentication, and reusability. [BGVV23] raised the open question of whether quantum PKE from one-way functions is possible when using classical keys and a quantum ciphertext; since ciphertext does not require to be distributed and reused, it does not share any difficulties of quantum public keys. They proved that it is impossible, as long as the conjecture in [ACC+22] was true. We extend Theorem 1.1 to the quantum ciphertext case, improving the work [BGVV23].

**Theorem 1.3.** *Perfect-complete quantum public key encryption, with classical keys and quantum ciphertext, does not exist in the quantum random oracle model.*

**Result 3: Extending to quantum public keys.** As discussed earlier, we believe achieving reusability and non-interactivity are the core of public key encryption; thus we focused on classical keys. Still, we examine the possibility to achieve QPKE with quantum keys from one-way functions. Particularly, we show that:

**Theorem 1.4.** *Perfect-complete quantum public key encryption, with quantum public keys, classical secret keys and classical/quantum ciphertext, does not exist in the quantum random oracle model, if the public key is a deterministic function of the secret key, independent of the oracle.*

Here, a public key $|\mathsf{pk}\rangle$ is said to be uniquely determined by a secret key $\mathsf{sk}$ if it can be generated by a procedure that depends only on $\mathsf{sk}$ but not on the random oracle. Notably, even when public keys depend solely on secret keys, it remains unclear how to dequantize the quantum state into a classical string. As a result, we cannot directly reduce Theorem 1.4 to Theorem 1.1 and must instead adopt a different approach.

Our impossibility result is tight for all known QPKE constructions with quantum keys [Col23, MW24, KMNY24, BGH+23], as they all require $|\mathsf{pk}\rangle$ to depend on both $\mathsf{sk}$ and the random oracle.

## 1.2 Other Related Works

There have also been attempts to separate QCCC key agreement from weaker primitives in "Nanocrypt" (primitives that are presumably weaker than one-way functions). The separation between the worlds "Nanocrypt" and Cryptomania in the QCCC setting has also been studied in [AGL25], where the authors showed that there is no black box construction from long pseudo-random function-like states to QCCC key agreement schemes.

Our result proves a black box barrier to constructing public key encryption from one-way functions in the QCCC setting. A different line of research shows that, unlike the classical hierarchy, it is also impossible to have black-box constructions of one-way functions from QCCC key agreements. In [GMMY24], the authors show relative to a unitary oracle, it is possible to obtain QCCC key agreement while BQP=QCMA. Another work by Kretschmer et al. [KQT25] shows that relative to a classical oracle, it is possible to obtain a quantum computable trapdoor one-way function, while P=NP. As trapdoor OWFs imply public key encryption, it indicates that quantum public key encryption can exist even without one-way functions in the QCCC setting. We believe that our work together with their works provides a more complete characterization of the relation between one-way functions and other QCCC primitives.

## 1.3 Technical Overview

**The quantum Markov chain method and the limitation of prior works.**

The method was first proposed by [LLLL24] in the context of quantum cryptography. They consider the following scenario, where they construct a two-round key agreement from PKE:

1. Alice first runs the key generation algorithm (with oracle access to $H$), sends the public key $m_0 := \mathsf{pk}$ to Bob and keeps the secret key $\mathsf{sk}$ herself.

2. Bob then encrypts a random key $k$ by running the encryption algorithm (with oracle access to $H$) $\mathsf{Enc}(\mathsf{pk}, k) \to \mathsf{ct}$, and sends back the ciphertext $m_1 := \mathsf{ct}$ to Alice.

3. Alice runs the decryption algorithm (with oracle access to $H$) to retrieve the key $k$.

One can, without loss of generality, assume Bob has a small query weight on every input. This step can be guaranteed by running, measuring Bob multiple times and thus removing all heavy

queries. We will handle it carefully in the main body; for the purpose of the overview, we simply assume Bob has no heavy queries.

Their idea is to create some Eve such that the conditional mutual information $I(\mathsf{A} : \mathsf{B}|\mathsf{E}) \leq \epsilon$ by making $\mathsf{poly}(1/\epsilon)$ queries, which implies the three systems form an approximate Markov chain. By the operational meaning of the quantum Markov chain shown in [FR15], there exists some channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{A}'$ that generates a copy of Alice system, while guaranteeing the joint state $\sigma_{\mathsf{A}'\mathsf{E}\mathsf{B}} = \mathcal{T}(\rho_{\mathsf{E}\mathsf{B}})$ is $O(\sqrt{\epsilon})$ close to the original joint state $\rho_{\mathsf{A}\mathsf{E}\mathsf{B}}$. Here $\rho_{\mathsf{A}\mathsf{E}\mathsf{B}}$ and $\sigma_{\mathsf{A}'\mathsf{E}\mathsf{B}}$ are states right before Alice runs the decryption algorithm. The main difficulty that [LLLL24, LLLL25] aimed to solve is to execute the rest of Alice and produce the key, due to the potential inconsistency of $\mathsf{A}'$ and the real random oracle $H$. Intuitively, the simulated Alice's view is unlikely to match with the real random oracle; and any random oracle that is compatible with Alice's view does not automatically work with Bob's view.

In [LLLL24], they manage to solve the problem by enforcing the key generation algorithm to only make classical queries. By measuring the register $\mathsf{A}'$, they can generate a classical query record $R_{\mathsf{A}'}$ of polynomial size, and Eve will simulate the run of $\mathsf{A}$ on the oracle reprogrammed by $R_{\mathsf{A}'}$, denoted by $H^{R_{\mathsf{A}'}}$. They show that the simulated Alice's view, Bob's view, all transcripts and the oracle $H^{R_{\mathsf{A}'}}$ will be in the support of real executions of the protocol. Thus, by the perfect correctness of the protocol, we can conclude that Eve will get the correct key by running the decryption algorithm.

We summarize their algorithm as in Figure 1. It relies on maintaining a polynomial-sized query record $R_{\mathsf{A}}$ to ensure that reprogramming does not significantly disturb Bob's state (since we assume that Bob has no heavy query). Therefore, extending their result to a quantum key generation process seems difficult, as query transcripts are no longer well-defined in the quantum setting.

**Algorithm (Eve's Attack) in [LLLL24] (Informal).**
**Input:** random oracle $H : [N] \to \{0,1\}$ with $N = 2^n$, transcripts $(m_0, m_1)$, parameter $\epsilon$.
**Output:** key $k_E$ such that $\Pr[k_E = k_A] = 1 - O(\sqrt{\epsilon})$.

- **Step 1: Sample a simulated Alice's view.**

  Let $\mathsf{A}$ denote the registers of Alice, and $\mathsf{B}$ denote the registers of Bob right after Bob sends out the second message $m_1$.

  Run Bob repeatedly with input $m_0$ and oracle access to $H$ polynomial many times in $n, 1/\epsilon$. At this moment, $I(\mathsf{A} : \mathsf{B}|\mathsf{E}) \leq \varepsilon$.

  Apply the reconstruction channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{A}'$ from [FR15] to generate a fake copy $\mathsf{A}'$ of $\mathsf{A}$, then measure $\mathsf{A}'$ to obtain a simulated secret key $\mathsf{sk}'$ and classical transcript $R_{\mathsf{A}'}$.

- **Step 2: Reprogram the oracle $H$ and run the decryption algorithm.**

  Run Alice with $m_1$, the simulated $\mathsf{sk}'$ and oracle access to $H^{R_{\mathsf{A}'}}$. Obtain a key $k_E = k'$ and output it.

Figure 1: Framework of Eve's attack algorithm in [LLLL24].

In a following paper [LLLL25], the authors introduce a view from boolean function analysis

in an attempt to attack PKE with quantum key generation. It is known that the probability of Alice outputting certain $(\mathsf{sk}', m_0 = \mathsf{pk})$ can be written as a low-degree polynomial $f(H)$, here $H$ is treated as the truth table of the random oracle. The key observation is that if there is some polynomial-sized partial assignment $\mu$ such that $f(H^\mu) \neq 0$ for all $H$[1], the partial assignment $\mu$ can replace the $R_{\mathsf{A}'}$ in [LLLL24] and the algorithm can reprogram the oracle on the points defined by the partial assignment $\mu$. The authors make some conjecture on the existence of a distribution on such partial assignment $\mu$. They prove based on the conjecture that, there exists a separation between PKE and OWF in the QCCC model.

**Improving the separation framework.**

As explained above, the main obstacle is to construct an oracle that is consistent with Bob's view and with the simulated view of Alice. Our contribution is a refinement of the prior framework which eventually enables us to complete the proof. Following [LLLL24, LLLL25], if we can find an oracle $H'$ that meets all consistency conditions, then Eve can recover the correct key by running the decryption algorithm (as in Step 2) — this remains true even if the decryption algorithm has unbounded queries! Even though the attack is inefficient in this case, prior attacks did not exploit the fact that decryption makes only a polynomial number of random-oracle queries. The framework in Figure 1 looks for an $H'$ such that $(\mathsf{sk}', H')$ in the support of real executions. We relax this to computational closeness: i.e., $(\mathsf{sk}', H')$ only needs to be computationally close to some real execution. This weaker requirement still guarantees that the query-bounded algorithm (the decryption algorithm) produces outputs that are sufficiently close.

By leveraging the idea, we show that we can find $H'$ satisfying *either* $(\mathsf{sk}', H')$ is in the support of real executions, *or* $(\mathsf{sk}', H')$ is computationally close to some real execution. More specifically, we show a *win-win situation* that Eve can efficiently find a partial assignment $\mu$:

- Case (a). *Either*, the probability that Alice with oracle access to $H^\mu$ outputs $(\mathsf{sk}', m_0)$ is non-zero.
- Case (b). *Or*, there exists $\mu'$ (may not be efficiently computable by Eve) such that the probability that Alice with oracle access to $H^{\mu\cdot\mu'}$ outputs $(\mathsf{sk}', m_0)$ is non-zero and $(\mathsf{sk}', H^{\mu\cdot\mu'})$ and $(\mathsf{sk}', H^\mu)$ are indistinguishable by the decryption algorithm.
- Finally, $\mu$ and $\mu'$ must both be polynomial-sized. This is to guarantee that the reprogrammed oracle $H^\mu$ or $H^{\mu\cdot\mu'}$ are consistent with Bob who makes no heavy query.

Assuming Eve can find such a $\mu$ with polynomial number of queries, the above framework will produce a key with probability close to 1.

**Compute the partial assignment: a win-win argument.**

We discuss our main idea: how Eve can compute such a partial assignment $\mu$ of polynomial size. Let $f(H)$ be the polynomial as described above, denoting the probability that on oracle $H$, Alice outputs $(\mathsf{sk}', m_0)$. Since Alice only makes a polynomial number of queries, $f$ has a low degree

[1] In other words, the oracle $H^\mu$ can still produce $(\mathsf{sk}, m_0)$ with a non-zero probability. With perfect completeness, we can argue that under the oracle $H^\mu$, Alice can still produce the correct key.

**Algorithm (Eve's Attack) in this work (Informal).**
**Input:** random oracle $H : [N] \to \{0,1\}$ with $N = 2^n$, transcripts $(m_0, m_1)$, parameter $\epsilon$.
**Output:** key $k_E$ such that $\Pr[k_E = k_A] = 1 - O(\sqrt{\epsilon})$.

- **Step 1: Sample a simulated Alice's view.**
  Identical to Step 1 in Figure 1, measure the simulated Alice's view to get $\mathsf{sk}'$.
- **Step 2.0: Compute the partial assignment.**
  Find a partial assignment $\mu$ that satisfies either Case (a) or Case (b) above.
- **Step 2.1: Reprogram the oracle $H$ and run the decryption algorithm.**
  Run Alice with $m_1$, the simulated $\mathsf{sk}'$ and oracle access to $H^\mu$. Obtain a key $k_E$ and output.

Figure 2: Improved framework of Eve's attack algorithm in this work.

$d$. Given the fact that on some oracle, Alice can output $(\mathsf{sk}', m_0)$; $f$ must be not identically zero. Furthermore, since $f$ is completely determined by $(\mathsf{sk}', m_0)$, Eve knows $f$.

Eve starts by a **maximal**[2] **set** $S$ of **pairwise disjoint** maximum monomials $\{w_1, w_2, \ldots, w_t\}$, where a maximum monomial is defined as a monomial of degree equal to that of $f$. In other words, each $w_i$ consists of $\deg(f)$ variables and every $w_i, w_j$ are mutually disjoint. By [Alo99, Mid04], for every monomial $w_i$ and every $H$, there will be a partial assignment $\mu_i$ on variables in $w_i$, such that $f(H^{\mu_i}) \neq 0$.

If $S$ consists of a lot of such monomials (say more than some threshold $m$, which is a polynomial), we can argue that the current random oracle $H$ together with $\mathsf{sk}'$ will be close to $(H', \mathsf{sk}')$ for some real executions, in the view of the decryption algorithm. This is because the decryption algorithm will only make at most a polynomial number of queries; thus, at least for some $w_i$, the decryption algorithm will have a very small query weight on any variables associated with $w_i$. By [BBBV97], a small query weight means a small total variation distance between these two cases with oracle access to $H$ or oracle access to $H^{\mu_i}$. Therefore, Eve does not need to reprogram $H$ at all and $H$ itself (or an empty partial assignment) will work.

If $S$ only has fewer maximum monomials, every monomial may have a big query weight; thus, [BBBV97] no longer works. In this case, Eve knows the total number of variables with $S$ is $t \cdot \deg(f) \leq m \cdot \deg(f)$, which is polynomial in $n$. Eve can just fix all variables in $S$. Eve finds a partial assignment $\mu$ for all variables in $S$, such that $f(H^\mu) \not\equiv 0$, i.e., the function $f(H^\mu)$ is not always zero; this again can be guaranteed by [Alo99, Mid04]. Since $S$ is the maximal set, fixing all variables in $S$ will reduce the degree of $f$ by at least one! Otherwise, Eve can find another monomial of degree $\deg(f)$ with all associated variables being disjoint with $S$, contradicting with $S$ being maximal.

Therefore, in this case, Eve reduces the degree of $f$ by at least one, and only fixes at most polynomial number of variables. Eve can then repeat the whole process at most $\deg(f)$ times, either at some point it finds a lot of disjoint maximum monomials, or it reduces the degree to zero and eventually the polynomial becomes a constant non-zero function. In either case, Eve finds a satisfying $\mu$. We give the algorithm of finding a satisfying $\mu$ in Figure 3.

[2]Here, maximal means that for every maximum monomial $w \notin S$, the set $S \cup \{w\}$ is not pairwise disjoint.

**Compute a partial assignment $\mu$.**
**Input:** a non-zero polynomial $f$ of degree $d$, some threshold $m$.
**Output:** a partial assignment $\mu$.

- Initiate $\tilde{\mu} \leftarrow \emptyset$ and $\tilde{f} \leftarrow f$.
- Repeat until $\deg(\tilde{f}) = 0$:
  1. Find a maximal set $\mathcal{S}$ of maximum monomials in $\tilde{f}$.
  2. If $|\mathcal{S}| > m$: Break.
  3. Otherwise, find an assignment $\eta$ for variables in $\mathcal{S}$, such that $\tilde{f}(H^{\eta}) \not\equiv 0$.
     - Let $\tilde{\mu} \leftarrow \tilde{\mu} \cdot \eta$ and $\tilde{f}(H) := f(H^{\tilde{\mu}})$.
- Output $\mu := \tilde{\mu}$.

Figure 3: How to compute $\mu$.

## 2 Preliminaries

We assume familiarity with the basics of quantum computing and quantum information. For a comprehensive background, we refer the reader to [NC10]. Below, we present some backgrounds that are heavily used in this work.

### 2.1 Distance Measures

We recall the definitions of total variation distance and trace distance.

**Definition 2.1** (Total Variation Distance)**.** *Given two probability distributions $D_X$ and $D_Y$ over a finite domain $\mathcal{X}$, the total variation distance between them is defined as*

$$TV(D_X, D_Y) = \frac{1}{2} \sum_{x \in \mathcal{X}} |D_X(x) - D_Y(x)| .$$

*Here, $D_X(x)$ and $D_Y(x)$ denote the probability of $x$ drawn from $D_X$ and $D_Y$ respectively.*

**Definition 2.2** (Trace Distance)**.** *For any two quantum states $\rho$ and $\sigma$, the trace distance is defined by*

$$TD(\rho, \sigma) = \frac{1}{2} \operatorname{Tr}\left[\sqrt{(\rho - \sigma)^{\dagger}(\rho - \sigma)}\right] = \sup_{0 \le \Lambda \le I} \operatorname{Tr}\left[\Lambda(\rho - \sigma)\right] .$$

The following lemma is standard (e.g., see [LLLL24]). We include the proof for completeness.

**Lemma 2.3.** *For two probability distributions $D_X$ and $D_Y$ over the same classical domain, if $TV(D_X, D_Y) \le \epsilon$, we have that*

$$\Pr_{x \leftarrow D_X}[x \notin \mathsf{supp}(D_Y)] \le 2\epsilon.$$

*Proof.* $\sum_{x \notin \mathsf{supp}(D_Y)} D_X(x) \le \sum_x \left|D_X(x) - D_Y(x)\right| = 2TV(D_X, D_Y) \le 2\epsilon.$ □

## 2.2 Quantum Oracle Model and Random Oracle

In the quantum oracle model, a quantum algorithm $\mathcal{A}$ can make quantum queries to an oracle function $H : [2^n] \to \{0,1\}$ via a unitary transformation $U_H$ mapping $|i,b\rangle$ to $|i, b \oplus H(i)\rangle$. We denote such an algorithm by $\mathcal{A}^H$, which can be expressed as a sequence of unitaries: $U_1, U_H, U_2, U_H, \ldots, U_d, U_H, U_{d+1}$. Here $U_1, \ldots, U_{d+1}$ are local unitaries acting on $\mathcal{A}$'s internal register.

**Definition 2.4** (Query Weight)**.** *Consider a quantum algorithm $\mathcal{A}$ that makes $d$ queries to an oracle $H$. Denote the quantum state immediately after $t$ queries to the oracle as*

$$|\psi_t\rangle = \sum_{i,w} \alpha_{i,w,t} |i,w\rangle,$$

*where $w$ is the content of the workspace register. Define the query weight $q_i$ of input $i$ as*

$$q_i = \sum_{t=1}^{d} \sum_{w} |\alpha_{i,w,t}|^2.$$

**Lemma 2.5** ([BBBV97])**.** *Consider two oracles $H, \tilde{H}$, and a quantum query algorithm $\mathcal{A}$ which makes $d$ queries. Let $|\psi_d\rangle$ and $|\phi_d\rangle$ denote the final state before measurement when running $\mathcal{A}$ on $H$ and $\tilde{H}$ respectively, and $q_i$ denote the query weight of input $i$ when running $\mathcal{A}$ on $H$. Then we have that*

$$\||\psi_d\rangle - |\phi_d\rangle\| \leq 2\sqrt{d} \sqrt{\sum_{i:\, \tilde{H}(i) \neq H(i)} q_i}.$$

We will also consider the quantum random oracle model (QROM). In this setting, a quantum algorithm has access to a random oracle $H : [2^{n_\lambda}] \to \{0,1\}$, which is chosen from the uniformly random distribution over all functions mapping $[2^{n_\lambda}]$ to $\{0,1\}$.

## 2.3 Entropy and Information

**Definition 2.6** (Von Neumann Entropy)**.** *Let $\rho \in \mathbb{C}^{2^n}$ be a quantum state describing the system $\mathsf{A}$, and let $|\phi_1\rangle, |\phi_2\rangle, \ldots, |\phi_{2^n}\rangle$ be an eigenbasis for $\rho$, so that*

$$\rho = \sum_i \eta_i |\phi_i\rangle \langle\phi_i|.$$

*The Von Neumann entropy of $\rho$, denoted by $S(\rho)$ or $S(\mathsf{A})_\rho$, is defined as*

$$S(\mathsf{A})_\rho = S(\rho) = -\sum_i \eta_i \log(\eta_i).$$

*For a composite system $\mathsf{AB}$ with joint state $\rho_{\mathsf{AB}}$, the conditional Von Neumann entropy is defined by*

$$S(\mathsf{A}|\mathsf{B})_\rho = S(\mathsf{AB})_\rho - S(\mathsf{B})_\rho.$$

In the following, we will omit the subscript $\rho$ when the quantum state is clear from context. For example, we will write $S(\mathsf{A})$ instead of $S(\mathsf{A})_\rho$, and $I(\mathsf{A} : \mathsf{B})$ instead of $I(\mathsf{A} : \mathsf{B})_\rho$.

**Definition 2.7** (Mutual Information)**.** *Given a quantum state $\rho$ that describes the joint systems* $\mathsf{A}$ *and* $\mathsf{B}$*, the mutual information between* $\mathsf{A}$ *and* $\mathsf{B}$ *is given by*

$$I(\mathsf{A} : \mathsf{B}) = S(\mathsf{A}) + S(\mathsf{B}) - S(\mathsf{AB}).$$

**Definition 2.8** (Conditional Mutual Information, CMI)**.** *Let $\rho$ be a quantum state describing the three joint systems* $\mathsf{A}$*,* $\mathsf{B}$*, and* $\mathsf{C}$*. Then the conditional mutual information is defined as*

$$I(\mathsf{A} : \mathsf{B}|\mathsf{C}) = S(\mathsf{AC}) + S(\mathsf{BC}) - S(\mathsf{ABC}) - S(\mathsf{C}).$$

The strong subadditivity property states that both the mutual information and the conditional mutual information are always non-negative.

**Lemma 2.9** (Strong Subadditivity, [AL70])**.** *For Hilbert spaces* $\mathsf{A}$*,* $\mathsf{B}$*, and* $\mathsf{C}$*, it holds that*

$$S(\mathsf{AC}) + S(\mathsf{BC}) \geq S(\mathsf{ABC}) + S(\mathsf{C}).$$

*In its conditional form, for Hilbert spaces* $\mathsf{A}$*,* $\mathsf{B}$*,* $\mathsf{C}$*, and* $\mathsf{D}$*, we have*

$$S(\mathsf{AC}|\mathsf{D}) + S(\mathsf{BC}|\mathsf{D}) \geq S(\mathsf{ABC}|\mathsf{D}) + S(\mathsf{C}|\mathsf{D}).$$

Fawzi and Renner [FR15] provided an insightful characterization of quantum states when the conditional mutual information is nearly zero. Intuitively, a small value of $I(\mathsf{A} : \mathsf{B}|\mathsf{E})$ indicates that the system $\mathsf{A}$ can be approximately reconstructed from system $\mathsf{E}$. Formally,

**Theorem 2.10** (Approximate Quantum Markov Chain, [FR15])**.** *For any state $\rho_{\mathsf{AEB}}$ over systems* $\mathsf{AEB}$*, there exists a explicitly constructible channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{A}'$ such that the trace distance between the reconstructed state $\sigma_{\mathsf{A}'\mathsf{EB}} = \mathcal{T}(\rho_{\mathsf{EB}})$ and the original state $\rho_{\mathsf{AEB}}$ is at most*

$$\sqrt{\ln 2 \cdot I(\mathsf{A} : \mathsf{B}|\mathsf{E})_\rho}.$$

## 2.4 Notations in Boolean Function Analysis

Any function $f : \{0,1\}^N \to \mathbb{R}$ has a unique expression as a multilinear polynomial

$$f(x) = \sum_{S \subseteq [N]} a_S \cdot x_S,$$

where $x_S := \Pi_{i \in S} x_i$, and the coefficient $a_S$ is given by $a_S = 2^{-N} \sum_x f(x) \cdot x_S$. The *degree* of $f$, denoted $\deg(f)$, is defined as the degree of its multilinear polynomial expression, i.e., $\max\{|S| : a_S \neq 0\}$. A monomial $x_S$ is called *maximum* if $a_S \neq 0$ and it has degree $\deg(f)$, i.e., $|S| = \deg(f)$. Two monomials $x_S$ and $x_T$ are called *disjoint* if $S \cap T = \emptyset$. We say that $f$ is *not identically zero* if $f(x) \not\equiv 0$.

A *partial assignment* is a function $\mu : [N] \to \{0, 1, \star\}$. We define the support of $\mu$ as $\mathsf{supp}(\mu) := \{i | \mu(i) \neq \star\}$, and the size as $|\mu| := |\mathsf{supp}(\mu)|$. $\mu$ is called *empty* if $|\mu| = 0$. For $x \in \{0,1\}^N$, we define the modification of $x$ with $\mu$, denoted by $x^\mu$, as the string $x' \in \{0,1\}^N$ such that

$$x'_i := \begin{cases} \mu(i) & \text{if } i \in \mathsf{supp}(\mu), \\ x_i & \text{otherwise.} \end{cases}$$

Given two partial assignments $\mu$ and $\eta$, define their *product*, denoted by $\mu \cdot \eta$, to be the partial assignment satisfying that $x^{\mu\cdot\eta} = (x^\mu)^\eta$ for any $x \in \{0,1\}^N$. Note that the product operator is associative but not commutative. We say that two partial assignments $\mu$ and $\eta$ are *disjoint* if $\mathsf{supp}(\mu) \cap \mathsf{supp}(\eta) = \emptyset$.

**Lemma 2.11** ([BBC+01]). *Suppose a quantum algorithm makes $d$ queries to a Boolean string[3] $x \in \{0,1\}^N$, and the acceptance probability is denoted by $f(x)$. Then the function $f : \{0,1\}^N \to \mathbb{R}$ has degree at most $2d$. That is, $f$ can be expressed as*

$$f(x) = \sum_{|S|\le 2d} a_S \cdot x_S.$$

**Lemma 2.12** ([Mid04]). *Let $f : \{0,1\}^N \to \mathbb{R}$ be any function that is not identically zero, and $x_S$ be any maximum monomial of $f$. For any $x \in \{0,1\}^N$, there exists a $\mu$ with $\mathsf{supp}(\mu) = S$ such that $f(x^\mu) \neq 0$.*

## 2.5 Quantum Public-Key Encryption

This section provides the formal definition of Quantum Public-Key Encryption (QPKE) and Quantum Key Agreement (QKA) in QROM.

**Definition 2.13.** *Let $\lambda \in \mathbb{Z}_+$ be the security parameter and $H\colon [2^{n_\lambda}] \to \{0,1\}$ be a random oracle. A quantum public-key encryption scheme, relative to $H$, consists of the following three bounded-query quantum algorithms:*

- $\mathsf{Gen}^H(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$*: the key generation algorithm that generates a pair of classical public key* $\mathsf{pk}$ *and classical secret key* $\mathsf{sk}$*.*
- $\mathsf{Enc}^H(\mathsf{pk}, m) \to \mathsf{ct}$*: the encryption algorithm that takes as input the public key* $\mathsf{pk}$ *and the plaintext $m$, and produces a classical ciphertext* $\mathsf{ct}$*.*
- $\mathsf{Dec}^H(\mathsf{sk}, \mathsf{ct}) \to m'$*: the decryption algorithm that takes as input the secret key* $\mathsf{sk}$ *and the ciphertext* $\mathsf{ct}$*, and outputs the plaintext $m'$.*

*The algorithms need to satisfy the following requirements:*

**Perfect Completeness** $\Pr\left[\mathsf{Dec}^H\left(\mathsf{sk}, \mathsf{Enc}^H(\mathsf{pk}, m)\right) = m\colon \mathsf{Gen}^H(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})\right] = 1.$

**IND-CPA Security** *For any adversary $\mathcal{E}^H$ that makes $\mathsf{poly}(\lambda)$ quantum queries, for every two plaintexts $m_0 \neq m_1$ chosen by $\mathcal{E}^H$ after seeing* $\mathsf{pk}$*, we have*

$$\Pr_{b\leftarrow\{0,1\}}\left[\mathcal{E}^H\left(\mathsf{pk}, \mathsf{Enc}^H(\mathsf{pk}, m_b)\right) = b\right] \le \frac{1}{2} + \mathsf{negl}(\lambda).$$

For simplicity, we use "QPKE" to refer to quantum public-key encryption schemes with classical secret key, public key, and ciphertext, unless specified otherwise. Besides, we will also consider QPKE schemes with quantum public keys, defined as follows.

[3] We interpret the string $x \in \{0,1\}^N$ as a function $x : [N] \to \{0,1\}$, and model queries to $x$ as oracle queries to this function.

**Definition 2.14** (QPKE with quantum public key)**.** *Let $\lambda \in \mathbb{Z}_+$ be the security parameter and $H\colon [2^{n_\lambda}] \to \{0,1\}$ be a random oracle. A quantum public-key encryption scheme with quantum public key, relative to $H$, consists of the following four bounded-query quantum algorithms:*

- $\mathsf{SKGen}^H(1^\lambda) \to \mathsf{sk}$*: the secret key generation algorithm that generates a classical secret key* $\mathsf{sk}$*.*
- $\mathsf{PKGen}^H(\mathsf{sk}) \to \rho_{\mathsf{pk}}$*: the public key generation algorithm that takes the secret key* $\mathsf{sk}$ *and generates a quantum state $\rho_{\mathsf{pk}}$ as the public key.*
- $\mathsf{Enc}^H(\rho_{\mathsf{pk}}, m) \to \mathsf{ct}$*: the quantum encryption algorithm that takes the public key $\rho_{\mathsf{pk}}$ and the plaintext $m$, and produces a classical or quantum ciphertext $\rho_{\mathsf{ct}}$.*
- $\mathsf{Dec}^H(\mathsf{sk}, \rho_{\mathsf{ct}}) \to m'$*: the quantum decryption algorithm that takes the secret key* $\mathsf{sk}$ *and the ciphertext $\rho_{\mathsf{ct}}$, and outputs the plaintext $m'$.*

*The algorithms need to satisfy the following requirements:*

**Perfect Completeness**

$$\Pr\left[\mathsf{Dec}^H\left(\mathsf{sk}, \mathsf{Enc}^H(\rho_{\mathsf{pk}}, m)\right) = m \colon \mathsf{SKGen}^H(1^\lambda) \to \mathsf{sk}, \mathsf{PKGen}^H(\mathsf{sk}) \to \rho_{\mathsf{pk}}\right] = 1.$$

**IND-CPA Security** *For any adversary $\mathcal{E}^H$ that receives $\mathsf{poly}(\lambda)$ copies of the public key and can make $\mathsf{poly}(\lambda)$ queries, and every two plaintexts $m_0 \neq m_1$ chosen by the adversary, we have*

$$\Pr_{b \leftarrow \{0,1\}}\left[\mathcal{E}^H\left(\rho_{\mathsf{pk}}^{\otimes \mathsf{poly}(\lambda)}, \mathsf{Enc}^H(\rho_{\mathsf{pk}}, m_b)\right) = b\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

In this paper, we focus on the setting where the quantum public key is uniquely determined by the secret key $\mathsf{sk}$; that is, the quantum algorithm $\mathsf{PKGen}(\mathsf{sk})$ makes no queries to the oracle $H$. This setting covers all possible QPKE schemes with classical public keys, as we may assume, without loss of generality, that $\mathsf{sk}$ contains a copy of $\mathsf{pk}$. Furthermore, we may assume that $\rho_{\mathsf{pk}}$ is a pure state, since $\mathsf{sk}$ can be taken to include a purification of $\rho_{\mathsf{pk}}$[4].

Lastly, we define Quantum Key Agreement in QROM.

**Definition 2.15** (Quantum Key Agreement in the Oracle Model)**.** *Let $\lambda \in \mathbb{Z}_+$ be the security parameter and let $H : [2^{n_\lambda}] \to \{0,1\}$ be a random oracle. A Quantum Key Agreement (QKA) protocol involves two parties, Alice and Bob, who initially begin with all-zero states. They can perform any quantum operations, make $\mathsf{poly}(\lambda)$ quantum queries to the oracle $H$, and exchange classical messages. At the end of the protocol, Alice and Bob output classical strings $k_A$ and $k_B$, respectively.*

*The protocol needs to satisfy the following conditions:*

**Correctness** $\Pr[k_A = k_B] \geq 1/\mathsf{poly}(\lambda)$*, where the probability is taken over the randomness of Alice and Bob's channels, and the random oracle $H$.*

**Security** *For any eavesdropper Eve that makes $\mathsf{poly}(\lambda)$ quantum queries to $H$, eavesdrops on classical communication between Alice and Bob and outputs $k_E$, we have $\Pr[k_A = k_E] = \mathsf{negl}(\lambda)$.*

Similar to the perfect completeness in QPKE, a QKA protocol is said to be *perfect complete* if it satisfies $\Pr[k_A = k_B] = 1$.

[4] Assume $\mathsf{PKGen}(\mathsf{sk})$ first generates $\sum_x \alpha_x |x\rangle |\mathsf{pk}_x\rangle$, and then discards $|x\rangle$ to obtain $\rho_{\mathsf{pk}} := \sum_x |\alpha_x|^2 |\mathsf{pk}_x\rangle \langle \mathsf{pk}_x|$. Without affecting the scheme's functionality, we can include a copy of $x$ in $\mathsf{sk}$ so that the public key becomes a pure state $|\mathsf{pk}_x\rangle$.

# 3 Helper Lemmas

This section presents some helper lemmas, which may be of independent interest.

## 3.1 Information-Theoretic Tools

The following two information-theoretic lemmas from [LLLL24] will be used to prove our main results. We provide their proofs in Appendix A to make this paper self-contained. We assume the quantum algorithm always defers measurements to the end without loss of generality.

Lemma 3.1 upper bounds how much entropy a quantum algorithm can accumulate through oracle queries.

**Lemma 3.1** ([LLLL24]). *Consider an algorithm $\mathcal{A}$ that starts with a pure state, and makes $d$ quantum queries to a random oracle $H : [2^n] \to \{0,1\}$ without intermediate measurements. Let $\mathsf{A}$ denote the whole register of $\mathcal{A}$ and let $\rho$ be the quantum state right before the final measurement. Then, it holds that $S(\mathsf{A})_\rho \le 2d(n+1)$.*

Lemma 3.3 claims that repetition decreases CMI.

**Definition 3.2** (Permutation Invariance). *Let $\mathsf{A}, \mathsf{B}_1, \mathsf{B}_2, \cdots, \mathsf{B}_t$ be a $(t+1)$-partite quantum system. Given the joint state $\rho_{\mathsf{AB}_1\mathsf{B}_2\cdots\mathsf{B}_t}$, we say that $\mathsf{B}_1, \ldots, \mathsf{B}_t$ are permutation invariant if, for any permutation $\pi$ on $[t]$, it holds that*

$$\rho_{\mathsf{AB}_1\mathsf{B}_2\cdots\mathsf{B}_t} = \rho_{\mathsf{AB}_{\pi(1)}\mathsf{B}_{\pi(2)}\cdots\mathsf{B}_{\pi(t)}}.$$

**Lemma 3.3** (Lemma 4.2, [LLLL24]). *Let $\mathsf{A}, \mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_t, \mathsf{C}$ be a $(t+2)$-partite quantum system. Suppose the joint state $\rho_{\mathsf{ACB}_1\mathsf{B}_2\cdots\mathsf{B}_t}$ is fully separable, i.e., it can be written as $\sum_i p_i \rho_A^{(i)} \otimes \rho_C^{(i)} \otimes \rho_{B_1}^{(i)} \otimes \cdots \otimes \rho_{B_t}^{(i)}$. If $\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_t$ are permutation invariant, then there exists some $0 \le j \le t-1$ such that*[5]

$$I(\mathsf{B}_t : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_j)_\rho \le \frac{S(\mathsf{A})}{t}.$$

## 3.2 A Structural Property of Low-Degree Polynomials

In the proof of our main results, a key step involves explicitly reprogramming an unknown $x \in \{0,1\}^N$ (representing the oracle) by modifying at most $\mathsf{poly}(d)$ bits, in order to make a given degree-$d$ polynomial $f$ (representing the probability that $\mathsf{Gen}$ or $\mathsf{SKGen}$ outputs a particular $\mathsf{sk}$) evaluate to non-zero.

The following lemma, which builds heavily on Lemma 2.12, establishes a win-win situation: by obliviously modifying a small number of bits of the unknown $x$, denoted by a partial assignment $\mu$, either we can already guarantee that $f(x^\mu) \neq 0$, or there must exist many *disjoint* (albeit unknown) partial assignments to make $f$ evaluate to non-zero.

**Lemma 3.4.** *Let $m > 0$ be an integer. For any degree-$d$ function $f : \{0,1\}^N \to \mathbb{R}$ that is not identically zero, we can explicitly construct a partial assignment $\mu$ of $|\mu| \le md^2$ such that: either*

*(a) for any $x \in \{0,1\}^N$, $f(x^\mu) \neq 0$; or*

[5] We remark that $j$ cannot be simply set as $t-1$, since conditioning on more registers may increase CMI.

(b) *for any* $x \in \{0,1\}^N$*, there must exist* $m$ *pairwise disjoint partial assignments* $\mu_1, \ldots, \mu_m$ *of size at most* $d$ *such that* $f(x^{\mu_\ell \cdot \mu}) \neq 0$ *for all* $\ell \in [m]$.

*Proof.* We propose an algorithm to construct such a partial assignment $\mu$. The algorithm maintains a function $\tilde{f}$ and a partial assignment $\tilde{\mu}$. Initially, $\tilde{f} = f$ and $\tilde{\mu}$ is empty. The algorithm contains at most $\deg(f)$ rounds: in each but the last round, we extend $\tilde{\mu}$ by fixing at most $md$ additional bits, and reduce the degree of $\tilde{f}$ by at least 1. Specifically, in each round, the algorithm first constructs a maximal set $\mathcal{S}$ of disjoint maximum monomials of $\tilde{f}$ (so any maximum monomial of $\tilde{f}$ intersects with at least one monomial in $\mathcal{S}$); Then

1. If $|\mathcal{S}| > m$, then stop and return $\tilde{\mu}$;
2. Otherwise, fix all variables appearing in $\mathcal{S}$ while keeping the new $\tilde{f}$ not identically zero. To do this, process each variable $x_j$ in $\mathcal{S}$ one by one. For each, choose a value $b \in \{0,1\}$ such that $\tilde{f}$ remains not identically zero after setting $x_j$ as $b$. Such a choice always exists since $\tilde{f}$ is not identically zero. Let $\eta$ be the resulting partial assignment, and update $\tilde{\mu}$ as $\tilde{\mu} \cdot \eta$ and $\tilde{f}$ as $\tilde{f}^\eta$. Here, the function $\tilde{f}^\eta(x)$ is defined as $\tilde{f}(x^\eta)$. Now, if $\deg(\tilde{f}) = 0$, then stop and return $\tilde{\mu}$.

We now analyze the algorithm. First, we claim that the final $\tilde{\mu}$ satisfies either condition (a) or (b). This is because that:

- If the algorithm stops because $\deg(\tilde{f}) = 0$, then $\tilde{f}$ is a constant function that is not zero, say $\tilde{f}(x) \equiv c \neq 0$. Therefore, for any $x$, we have $f(x^{\tilde{\mu}}) = \tilde{f}(x) = c \neq 0$, and condition (a) is satisfied.
- If the algorithm stops because $|\mathcal{S}| > m$, then given any $x$, for each maximum monomial $\ell$ of $\tilde{f}$ in $\mathcal{S}$, by Lemma 2.12, there exists a partial assignment $\mu_\ell$ on the $\leq d$ variables of $\ell$ such that $f(x^{\mu_\ell \cdot \tilde{\mu}}) = \tilde{f}(x^{\mu_\ell}) \neq 0$. Recalling that the monomials in $\mathcal{S}$ are disjoint, we conclude that condition (b) is satisfied.

Next, we show that $|\mu| \leq md^2$, and therefore finish the proof. Since any maximum monomial of $\tilde{f}$ intersects with at least one monomial in $\mathcal{S}$, fixing all variables in $\mathcal{S}$ reduces $\deg(\tilde{f})$ by at least 1. Hence, the number of round is at most $d$. Moreover, in each round, $|\mu|$ increases by at most $md$. So we have $|\mu| \leq d \cdot (md) = md^2$. □

# 4 Impossibility of Perfect-Complete Quantum PKE

This section will prove that perfect-complete QPKE schemes do not exist in QROM. More formally, we have the following theorem.

**Theorem 4.1** (Restate of Theorem 1.1)**.** *For any perfect-complete QPKE in QROM, which makes* $d$ *queries to a random oracle* $H : [2^n] \to \{0,1\}$ *during each of* $\mathsf{Gen}, \mathsf{Enc}$ *and* $\mathsf{Dec}$*, there exists an adversary Eve that can break the scheme w.p.* $1 - O(\epsilon)$ *by making* $O\left(d^7 \log(d/\epsilon)/\epsilon^4 + nd^2/\epsilon^2\right)$ *queries to* $H$.

The remainder of this section presents the proof of Theorem 4.1. It is well-known that: given a perfect-complete QPKE scheme $(\mathsf{Gen}^H, \mathsf{Enc}^H, \mathsf{Dec}^H)$, one can construct a perfect-complete two-round key agreement protocol between two parties, Alice and Bob, as follows.

1. Alice computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}^H(1^\lambda)$ and sends $m_0 := \mathsf{pk}$ to Bob. Denote this stage by $\mathcal{A}_0$.

2. Bob randomly chooses $k_B \in \{0,1\}$, computes $\mathsf{ct} \leftarrow \mathsf{Enc}^H(m_0, k_B)$, sends $m_1 := \mathsf{ct}$ to Alice and outputs $k_B$. Denote this stage by $\mathcal{B}$.

3. Alice computes $k_A \leftarrow \mathsf{Dec}^H(\mathsf{sk}, m_1)$ and outputs $k_A$. Denote this stage by $\mathcal{A}_1$.

Each stage of the key agreement makes at most $d$ queries. Thus breaking this QKA also breaks QPKE. We now construct an eavesdropper Eve that sees $(m_0, m_1)$ and guesses the agreed key w.p. $1 - O(\epsilon)$ by making $O(\mathsf{poly}(n, d, 1/\epsilon))$ queries to $H$. Eve's attack algorithm consists of three steps, as summarized in Figure 4. The rest of the section is a detailed analysis of Eve's attack.

### 4.1 Step 1: Identify $\mathcal{B}$'s heavy queries.

The first step is to identify Bob's heavy queries, i.e., inputs with large query weight. These heavy queries will be kept unchanged when reprogramming the oracle in later steps, in order to ensure that the reprogrammed oracle will be indistinguishable from the real oracle from Bob's perspective.

Specifically, in this step, Eve computes a query record $R_E := \{(i_E, H(i_E))\}$ by repeating the following process $\frac{d^6}{\epsilon^4} \log \frac{d^6}{\epsilon^5}$ times:

1. Randomly choose $t \leftarrow [d]$, simulate $\mathcal{B}^H(m_0)$ to its $t$-th query to the oracle, and measure the input register, obtain outcome $i \in [2^n]$;

2. Classically query $i$ to the oracle and add $(i, H(i))$ to $R_E$.

We claim that, with high probability, Eve can identify all of Bob's heavy queries. Formally,

**Lemma 4.2.** *Let $q_i$ be the query weight of input $i$ when running $\mathcal{B}(m_0)$ on $H$, and $W_B := \{i : q_i \geq \epsilon^4/d^5\}$. Then $\Pr[W_B \not\subseteq R_E] \leq \epsilon$.*

*Proof.* For each $i \in W_B$, it would be measured w.p. at least $\epsilon^4/d^6$ at each repetition. Thus the probability that it is not measured is bounded by

$$\Pr[i \notin R_E] \leq \left(1 - \frac{\epsilon^4}{d^6}\right)^{\frac{d^6}{\epsilon^4} \log \frac{d^6}{\epsilon^5}} \leq \epsilon^5/d^6.$$

Since $\sum_i q_i = d$, we have $|W_B| \leq (\sum_i q_i)/(\epsilon^4/d^5) = d^6/\epsilon^4$ by Markov's inequality. Thus by a union bound, we have $\Pr[W_B \not\subseteq R_E] \leq \sum_{i \in W_B} \Pr[i \notin R_E] \leq \epsilon$. □

### 4.2 Step 2: Sample a fake secret key

The next step is to obtain a fake secret key that is indistinguishable from the real secret key from Bob's perspective. We take an information-theoretic approach: first reduce the mutual information between Alice and Bob's registers conditioned on Eve's registers to a small value, and then apply the reconstruction channel in Theorem 2.10 to sample a fake secret key.

Specifically, consider the time right before $\mathcal{B}$ performs the final measurement. Let $\mathsf{A}$ denote the registers of $\mathcal{A}_0$ and $\mathsf{B}$ denote the registers of $\mathcal{B}$. Eve repetitively runs $\mathcal{B}^H(m_0)$ and stops right before the final measurement for $4dn/\epsilon^2$ times, which yields $4dn/\epsilon^2$ copies $\mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_{4dn/\epsilon^2}$ of $\mathsf{B}$.

**Algorithm (Eve's Attack).**
**Input:** random oracle $H : [N] \to \{0,1\}$ with $N = 2^n$, transcripts $(m_0, m_1)$, parameters $d, \epsilon$.
**Output:** key $k_E$ such that $\Pr[k_E = k_A] = 1 - O(\epsilon)$.

1. **Step 1: Identify $\mathcal{B}$'s heavy queries.**

   Initialize a classical query record $R_E \leftarrow \emptyset$, and repeats the following $\frac{d^6}{\epsilon^4} \log \frac{d^6}{\epsilon^5}$ times:

   (a) Sample $t \leftarrow [d]$, simulate $\mathcal{B}^H(m_0)$ up to its $t$-th oracle query, measure the query register to obtain $i \in [2^n]$.

   (b) Query $i$ to $H$ classically and add $(i, H(i))$ to $R_E$.

2. **Step 2: Sample a fake secret key.**

   Let $\mathsf{A}$ denote the registers of $\mathcal{A}_0$, and $\mathsf{B}$ denote the registers of $\mathcal{B}$ right before $\mathcal{B}$ performs the final measurement.

   (a) Run $\mathcal{B}^H(m_0)$ repeatedly for $4dn/\epsilon^2$ times, stopping just before the final measurement, to obtain copies $\mathsf{B}_1, \ldots, \mathsf{B}_{4dn/\epsilon^2}$ of $\mathsf{B}$.

   (b) Let $j$ be the index[a] such that $I(\mathsf{A} : \mathsf{B}|R_E, \mathsf{B}_1, \ldots, \mathsf{B}_j) \leq \frac{\epsilon^2}{\ln 2}$, and $\mathsf{E} := (R_E, \mathsf{B}_1, \ldots, \mathsf{B}_j)$. Apply the reconstruction channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{A}'$ from Theorem 2.10 to generate a fake copy $\mathsf{A}'$ of $\mathsf{A}$, then measure $\mathsf{A}'$ to obtain a fake secret key $\mathsf{sk}'$.

3. **Step 3: Reprogram the oracle and run $\mathcal{A}_1$.**

   Define polynomials $g(H) := \Pr[\mathcal{A}_0^H \to (\mathsf{sk}', m_0)]$ and $f(H) := g(H^{R_E})$.

   (a) If $g$ is the zero polynomial, abort.

   (b) Otherwise, apply Lemma 3.4 to $f$ with $m = d^2/\epsilon^2$ to obtain a partial assignment $\mu : [N] \to \{0, 1, \star\}$ of size at most $d^4/\epsilon^2$.

   (c) Reprogram the oracle as $\tilde{H}(i) := H^\mu(i) = \begin{cases} \mu(i) & \text{if } i \in \mathsf{supp}(\mu) \\ H(i) & \text{otherwise} \end{cases}$.

   (d) Run $k_E \leftarrow \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)$, and output $k_E$.

[a] Existence of such $j$ is guaranteed by Lemma 3.3; see Section 4.2 for detailed analysis.

Figure 4: Summary of Eve's attack algorithm.

Observe that $\mathsf{B}, \mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_{4dn/\epsilon^2}$ are permutation invariant. By Lemma 3.1 and Lemma 3.3, there exists a $0 \leq j \leq 4dn/\epsilon^2$ such that

$$I(\mathsf{A} : \mathsf{B}|R_E, \mathsf{B}_1, \mathsf{B}_2, \cdots, \mathsf{B}_j) \leq \frac{S(\mathsf{A})}{4dn/\epsilon^2} \leq \frac{2d(n+1)}{4dn/\epsilon^2} \leq \frac{\epsilon^2}{\ln 2}.$$

Note $j$ is determined by the mixed state of $\mathsf{AB}$ and Eve, averaged over the distribution of all possible oracles, as $H$ is already traced out in Lemma 3.3. Thus $j$ can be computed by computationally

unbounded Eve without making oracle queries to $H$[6].

Eve only keeps the $j$ copies of $\mathsf{B}$, so that $\mathsf{E} := (R_E, \mathsf{B}_1, \mathsf{B}_2, \ldots, \mathsf{B}_j)$. By Theorem 2.10, there exists a quantum channel $\mathcal{T} : \mathsf{E} \to \mathsf{E} \otimes \mathsf{A}'$ which generates a fake copy $\mathsf{A}'$ of $\mathsf{A}$ such that

$$TD(\rho_{ABE}, \rho_{A'BE}) \leq \sqrt{\ln 2 \cdot I(\mathsf{A} : \mathsf{B}|\mathsf{E})} < \epsilon,$$

where $\rho_{ABE}$ is the state of system $\mathsf{ABE}$ and $\rho_{A'BE}$ is state of system $\mathsf{A'BE}$. Similar to $j$, the channel $\mathcal{T}$ only depends on the mixed state of $\mathsf{AEB}$, can thus be implemented by computationally unbounded Eve without making queries to $H$. Since $\mathsf{A}'$ contains a register $\mathsf{sk}'$ storing the secret key, Eve applies $\mathcal{T}$ on $\mathsf{E}$ to generate $\mathsf{A}'$, and uses $\mathsf{sk}'$ as the fake secret key.

**Lemma 4.3.** *Let* $\mathsf{View}_{ABE} := (\mathsf{sk}, m_0, k_B, m_1, R_E)$, $\mathsf{View}_{A'BE} := (\mathsf{sk}', m_0, k_B, m_1, R_E)$, $D_{ABE}$ *denote the distribution of* $\mathsf{View}_{ABE}$, *and* $D_{A'BE}$ *denote the distribution of* $\mathsf{View}_{A'BE}$. *We have*

$$\Pr_{\mathsf{View}_{A'BE} \leftarrow D_{A'BE}}[\mathsf{View}_{A'BE} \notin \mathsf{supp}(D_{ABE})] \leq 2\epsilon.$$

*Proof.* $\mathsf{View}_{ABE}$ and $\mathsf{View}_{A'BE}$ are obtained from performing measurement in computational basis on the corresponding registers of state $\rho_{ABE}$ and $\rho_{A'BE}$ respectively. Since $TD(\rho_{ABE}, \rho_{A'BE}) \leq \epsilon$, we have that $TV(D_{ABE}, D_{A'BE}) \leq \epsilon$ by the operational meaning of trace distance. By Lemma 2.3, we have
$\Pr_{\mathsf{View}_{A'BE} \leftarrow D_{A'BE}}[\mathsf{View}_{A'BE} \notin \mathsf{supp}(D_{ABE})] \leq 2\epsilon$. □

The above lemma shows that, with high probability, the tuple $(\mathsf{sk}', m_0, k_B, m_1, R_E)$ corresponds to a valid execution and is therefore compatible with some oracle $H'$. Here, we say that a tuple $(\mathsf{sk}', m_0, k_B, m_1, R_E)$ and an oracle $H'$ are compatible if (i) running the QKA protocol on $H'$ generates the view $(\mathsf{sk}', m_0, k_B, m_1)$ with non-zero probability; and (ii) the list of input-output pairs $R_E$ is consistent with $H'$. If Eve had access to such an oracle $H'$, it could just compute $k_E \leftarrow \mathsf{Dec}^{H'}(\mathsf{sk}', m_1)$ to break the QKA protocol, since the perfect completeness ensures that $k_E = k_B$.

### 4.3 Step 3: Reprogram the oracle and run $\mathcal{A}_1$

In the final step, Eve first obtains a reprogrammed oracle $\tilde{H}$ by modifying $O(d^4/\epsilon^2)$ entries of the original oracle $H$ (the details of the reprogramming will be specified later), and then computes $k_E$ by running the decryption algorithm $\mathcal{A}_1$ on $\tilde{H}$ using $\mathsf{sk}'$. In some cases, $\tilde{H}$ is compatible with the tuple $(\mathsf{sk}', m_0, k_B, m_1, R_E)$, in which case we have $k_E = k_B$. However, this compatibility does not always hold[7]. Nevertheless, we can argue that $(\mathsf{sk}', m_0, k_B, m_1, R_E)$ is compatible with another (possibly unknown) oracle $H'$ that is very close to $\tilde{H}$ and agrees with it on the heavy queries made by $\mathcal{A}_1$. In this case, by Lemma 2.5, we can still conclude that $k_E = k_B$ with high probability.

Here are the details of reprogramming. Let $N := 2^n$ and fix $\mathsf{sk}'$ and $m_0$. Let $g(H)$ denote the probability $\Pr[\mathcal{A}_0^H \to (\mathsf{sk}', m_0)]$ where $H \in \{0,1\}^N$ is treated as the truth table of oracle $H$. Define

[6] $j$ can be explicitly computed by the following (inefficient) procedure: (i) compute the density matrix $\rho$ of $\mathsf{ABE}$, averaged over the uniform distribution of oracles; (ii) For each $j$, compute $\mathrm{CMI}_j := I(\mathsf{A} : \mathsf{B}|R_E, \mathsf{B}_1, \mathsf{B}_2, \cdots, \mathsf{B}_j)_\rho$, which is a function of $\rho$; (iii) output a $j$ such that $\mathrm{CMI}_j \leq \epsilon^2/\ln 2$, of which existence is guaranteed by Lemma 3.3. One can also guess a uniformly random $j$ if we do not require finding the key with probability close to 1.

[7] Unlike [LLLL25], our reprogramming step does not always guarantee under $\tilde{H}$, the algorithm outputs $\mathsf{sk}'$ with non-zero probability.

$f(H) := g(H^{R_E})$ where we abuse $R_E$ as a partial assignment that assigns the $i$-th bit of $H$ as $y$ for all $(i, y) \in R_E$. Since $\mathcal{A}_0$ makes at most $d$ queries, we have $\deg(f) \le 2d$ by Lemma 2.11.

If $g$ is a zero polynomial, then Eve aborts[8]. Otherwise, Eve applies Lemma 3.4 on polynomial $f$ by setting $m = d^2/\epsilon^2$ and obtains a partial assignment $\mu : [N] \to \{0, 1, \star\}$ of size at most $m \cdot (2d)^2 = d^4/\epsilon^2$. We can assume $\mathsf{supp}(\mu) \cap \mathsf{supp}(R_E) = \emptyset$ because changing $x_i$ for $i \in \mathsf{supp}(R_E)$ has no effect on the value of $f(x) = g(x^{R_E})$. Then Eve reprograms the oracle as

$$\tilde{H}(i) := H^{\mu}(i) = \begin{cases} \mu(i) & \text{if } i \in \mathsf{supp}(\mu) \\ H(i) & \text{otherwise} \end{cases}.$$

We have the following lemma. Intuitively, by Lemma 2.5, $\tilde{H}$ is likely to be compatible with Bob's view $(m_1, k_B)$, as it differs from $H$ on only a small fraction of Bob's query weight.

**Lemma 4.4.** *For the reprogrammed oracle $\tilde{H}$ defined above and any quantum algorithm $\mathcal{B}$ making $d$ queries to the oracle,*

$$\Pr_{(k_B, m_1) \leftarrow \mathcal{B}^H(m_0)} \left[ (k_B, m_1) \in \mathsf{supp}\left(\mathcal{B}^{\tilde{H}}(m_0)\right) \middle| \mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE}) \right] \ge 1 - O(\epsilon),$$

*where we slightly abuse the notation $\mathcal{B}^H(m_0)$ for the output distribution of the algorithm $\mathcal{B}$.*

*Proof.* Combining Lemma 4.3 and Lemma 4.2 we have that

$$\begin{aligned} &\Pr[\mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE}) \wedge W_B \subseteq R_E] \\ \ge &1 - \Pr[\mathsf{View}_{A'BE} \notin \mathsf{supp}(D_{ABE})] - \Pr[W_B \not\subseteq R_E] \\ \ge &1 - O(\epsilon). \end{aligned}$$

Now consider when $\mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE})$ and $W_B \subseteq R_E$. Since $\mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE})$, $(\mathsf{sk}', m_0, R_E)$ is valid under some oracle, which implies $f(x)$ is not identically zero. Then Eve will not abort and $\tilde{H}$ is well-defined. We have

$$\begin{aligned} TV\left(\mathcal{B}^{\tilde{H}}(m_0), \mathcal{B}^{H}(m_0)\right) &\le 4 || \, |\psi_d\rangle - |\phi_d\rangle \, || \le 8\sqrt{d} \sqrt{\sum_{i:\, \tilde{H}(i) \neq H(i)} q_i} \\ &\le 8\sqrt{d} \sqrt{\frac{\epsilon^4}{d^5} |\{i : \tilde{H}(i) \neq H(i)\}|} \\ &\le 8\sqrt{d} \sqrt{\frac{\epsilon^4}{d^5} \cdot \frac{d^4}{\epsilon^2}} = O(\epsilon), \end{aligned}$$

where $|\psi_d\rangle$ and $|\psi_d\rangle$ are the states of $\mathcal{B}^{\tilde{H}}(m_0)$ and $\mathcal{B}^{H}(m_0)$ respectively, and $q_i$ is the query weight of input $i$ when running $\mathcal{B}$ on $H$. The first inequality comes from [BBBV97, Theorem 3.1], the second inequality is Lemma 2.5, the third inequality is because $H$ and $\tilde{H}$ only differ on inputs that are outside $W_B$, and the last inequality is because $|\{i : \tilde{H}(i) \neq H(i)\}| \le |\mu| \le d^4/\epsilon^2$.

[8] As we will see, this will never happen unless $\mathsf{View}_{A'BE} \notin \mathsf{supp}(D_{ABE})$.

By Lemma 2.3, we have that

$$\Pr_{(k_B,m_1)\leftarrow\mathcal{B}^H(m_0)}\left[(k_B,m_1)\in\mathsf{supp}\left(\mathcal{B}^{\tilde{H}}(m_0)\right)\middle|\mathsf{View}_{A'BE}\in\mathsf{supp}(D_{ABE})\wedge W_B\subseteq R_E\right]\geq 1-O(\epsilon).$$

The final statement follows from a conditional probability formula and Lemma 4.2. □

### 4.4 Putting things together

Now, we are ready to prove Theorem 4.1.

*Proof of Theorem 4.1.* We will prove that by the 3-step attack algorithm described above, Eve will output $k_E$ such that $\Pr[k_E = k_B] = 1 - O(\epsilon)$. First of all, by Lemma 4.3, we have that

$$\Pr[\mathsf{View}_{A'BE}\in\mathsf{supp}(D_{ABE})]\geq 1-O(\epsilon). \tag{1}$$

Now consider the case when $\mathsf{View}_{A'BE}\in\mathsf{supp}(D_{ABE})$, which implies that $(\mathsf{sk}', m_0, k_B, m_1, R_E)$ will be a valid execution under some oracle. In this case, the function $f$ is not identically zero, and Eve will obtain a partial assignment $\mu$ without aborting. By Lemma 3.4, one of the following cases must hold:

(a) $f(H^\mu)\neq 0$. As $\tilde{H}=H^\mu$, we have $\Pr[(\mathsf{sk}', m_0)\leftarrow\mathcal{A}_0^{\tilde{H}}] = f(H^\mu)$ is non-zero, which means $(\mathsf{sk}', m_0)\in\mathsf{supp}(\mathcal{A}_0^{\tilde{H}})$.

(b) There exist $d^2/\epsilon^2$ pairwise disjoint partial assignments $\mu_1,\ldots,\mu_{d^2/\epsilon^2}$ of size $\leq 2d$ such that $f(H^{\mu_\ell\cdot\mu})\neq 0$ for all $\ell\in[d^2/\epsilon^2]$. We can assume that for all $\ell$, $\mathsf{supp}(\mu_\ell)\cap\mathsf{supp}(R_E)=\emptyset$ as changing $x_i$ for $i\in\mathsf{supp}(R_E)$ has no effect on the value of $f(x)=g(x^{R_E})$. Then observe that $H^{\mu_\ell\cdot\mu}$ is the truth table of the following oracle

$$\tilde{H}_\ell(i) := H^{\mu_\ell\cdot\mu}(i) = \begin{cases}\mu_\ell(i) & \text{if } i\in\mathsf{supp}(\mu_\ell)\setminus\mathsf{supp}(\mu)\\ \tilde{H}(i) & \text{otherwise}\end{cases}.$$

Thus $\Pr[(\mathsf{sk}', m_0)\leftarrow\mathcal{A}_0^{\tilde{H}_\ell}] = f(H^{\mu_\ell\cdot\mu})$ is non-zero, which means $(\mathsf{sk}', m_0)\in\mathsf{supp}(\mathcal{A}_0^{\tilde{H}_\ell})$.

Next, we argue that in both cases, Eve will output $k_E = k_B$ with probability $1 - O(\epsilon)$.

*Case (a)* We argue that $\mathsf{View}_{A'BE} = (\mathsf{sk}', m_0, k_B, m_1, R_E)$ is compatible with $\tilde{H}$ with high probability. Obviously, $\tilde{H}$ is consistent with $R_E$. Moreover,

1. From perspective of $A'$, $(\mathsf{sk}', m_0)\in\mathsf{supp}(\mathcal{A}_0^{\tilde{H}})$ implies that $(\mathsf{sk}', m_0)$ is compatible with $\tilde{H}$.
2. From perspective of Bob, $\mathcal{B}^H(m_0)$ represents a distribution over key-message pairs $(k_B, m_1)$. Now suppose we run the algorithm $\mathcal{B}^{\tilde{H}}(m_0)$ instead. According to Lemma 4.4, with probability at least $1 - O(\epsilon)$, a pair $(k_B, m_1)$ produced by $\mathcal{B}^H(m_0)$ will also lie within the support of $\mathcal{B}^{\tilde{H}}(m_0)$.

So, in particular, $\mathsf{View}_{A'B}$ is a valid execution under $\tilde{H}$ with probability $1 - O(\epsilon)$. Conditioned on that $\mathsf{View}_{A'B}$ is valid under $\tilde{H}$, the perfect completeness implies that $k_E = \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)$ must equal $k_B$. Thus we have $\Pr[k_E = k_B] = 1 - O(\epsilon)$ in Case (a).

*Case (b)* Let $w_i$ be the query weight of input $i$ when running $\mathcal{A}_1(\mathsf{sk}', m_1)$ on $\tilde{H}$. Note that $\sum_i w_i \le d$ because $\mathcal{A}_1$ makes at most $d$ queries to $H$. Since $\mu_1, \ldots, \mu_{d^2/\epsilon^2}$ are disjoint partial assignments, there must exist a $\ell^* \in [d^2/\epsilon^2]$ such that

$$\sum_{i \in \mathsf{supp}(\mu_{\ell^*})} w_i \le \frac{d}{d^2/\epsilon^2} = \frac{\epsilon^2}{d}. \tag{2}$$

For simplicity, let $H'$ denote $\tilde{H}_{\ell^*}$. First, imagine that Eve runs $\mathcal{A}_1(\mathsf{sk}', m_1)$ on $H'$ and obtains a key $k'_E$. Observe that $H'$ and $H$ differ by at most $|\mu_{l^*} \cdot \mu| \le 2d + O(d^4/\epsilon^2) = O(d^4/\epsilon^2)$ positions and $\mathsf{supp}(\mu_{l^*} \cdot \mu) \cap \mathsf{supp}(R_E) = \emptyset$. By the same argument as in Lemma 4.4, we have

$$\Pr_{(k_B, m_1) \leftarrow \mathcal{B}^H(m_0)} \left[ (k_B, m_1) \in \mathsf{supp}\left(\mathcal{B}^{H'}(m_0)\right) \middle| \mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE}) \right] = 1 - O(\epsilon). \tag{3}$$

Then, by the same argument as in Case (a), we have

$$\Pr_{k'_E \leftarrow \mathcal{A}_1^{H'}(\mathsf{sk}', m_1)} \left[ k'_E = k_B \middle| (k_B, m_1) \in \mathsf{supp}\left(\mathcal{B}^{H'}(m_0)\right), \mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE}) \right] = 1. \tag{4}$$

Combining Eqs. (1), (3), and (4), we have $\Pr_{k'_E \leftarrow \mathcal{A}_1^{H'}(\mathsf{sk}', m_1)} \left[ k'_E = k_B \right] = 1 - O(\epsilon)$, which means Eve will find the key with probability $1 - O(\epsilon)$ if it runs $\mathcal{A}_1(\mathsf{sk}', m_1)$ on oracle $H'$. However, Eve knows only the existence of $H'$, but does not know how to access it. The next step is to argue that by running $\mathcal{A}_1(\mathsf{sk}', m_1)$ on oracle $\tilde{H}$ instead, as done in the attack algorithm, Eve can also find the key with high probability. For any fixed real oracle $H$, we have that

$$\begin{aligned} TV\left(\mathcal{A}_1^{H'}(\mathsf{sk}', m_1), \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)\right) &\le 4 || \, |\psi_d\rangle - |\phi_d\rangle \, || \\ &\le 8\sqrt{d} \sqrt{\sum_{i : H'(i) \neq \tilde{H}(i)} w_i} \\ &\le 8\sqrt{d} \sqrt{\sum_{i \in \mu_{\ell^*}} w_i} \le 8\sqrt{d}\sqrt{\frac{\epsilon^2}{d}} = O(\epsilon) \end{aligned}$$

where $|\psi_d\rangle$ and $|\psi_d\rangle$ are the states of $\mathcal{A}_1^{H'}(\mathsf{sk}', m_1)$ and $\mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)$ right before the final measurement, respectively. The first inequality comes from [BBBV97, Theorem 3.1], the second inequality is Lemma 2.5, the third inequality is by definition of $H'$, and the last inequality is Eq. (2).

Since the distribution of $k_E \leftarrow \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)$ and $k'_E \leftarrow \mathcal{A}_1^{H'}(\mathsf{sk}', m_1)$ are $O(\epsilon)$-close for any $H$, replacing $k'_E$ with $k_E$ in Eq. (4) will only cause $O(\epsilon)$ loss of the probability, i.e.,

$$\Pr_{k_E \leftarrow \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)} \left[ k_E = k_B \middle| (k_B, m_1) \in \mathsf{supp}\left(\mathcal{B}^{H'}(m_0)\right), \mathsf{View}_{A'BE} \in \mathsf{supp}(D_{ABE}) \right] = 1 - O(\epsilon). \tag{5}$$

Combining Eqs. (1), (3), and (5), we have $\Pr_{k_E \leftarrow \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', m_1)} [k_E = k_B] = 1 - O(\epsilon)$.

Finally, we analyze the query complexity of Eve's attack algorithm. Step 1 requires at most $(d+1) \cdot d^6 \log(d^6/\epsilon^5)/\epsilon^4 = O(d^7 \log(d/\epsilon)/\epsilon^4)$ queries. Step 2 requires at most $d \cdot 4dn/\epsilon^2 = O\left(d^2 n/\epsilon^2\right)$ queries. Step 3 requires $d$ queries. Thus Eve makes $O\left(d^7 \log(d/\epsilon)/\epsilon^4 + nd^2/\epsilon^2\right)$ queries in total. □

### 4.5 Extending to quantum public key and ciphertext

By further reviewing our proof, we can extend the impossibility result to quantum $m_0$ and $m_1$. Specifically, as long as the public key $|m_0\rangle$ is a pure state that is uniquely determined by the secret key, and Eve can access polynomially many copies of $|m_0\rangle$, the attack algorithm still works, with a few minor modifications detailed below.

- Steps 1 and 2 of the attack require Eve to run $\mathcal{B}^H(|m_0\rangle)$ for polynomial times. As Eve can obtain polynomially many copies of $|m_0\rangle$, these two steps are doable and the related analysis still holds.
- In Step 3, since Eve does not have the full description of the quantum $m_0$, and thus is not capable of identifying the polynomial that represents the probability of outputting $(\mathsf{sk}', m_0)$. However, the key observation is that since $|m_0\rangle$ is uniquely determined by the secret key, we can instead define the polynomial $f$ as the probability of the oracle outputting $\mathsf{sk}'$ alone, i.e.,

  $$g(H) := \Pr[\mathcal{A}_0^H \to \mathsf{sk}'], \quad f(H) := g(H^{R_E}).$$

  Then Eve applies Lemma 3.4 on this $f$, obtains a reprogrammed oracle $\tilde{H}$, and finally outputs $k_E \leftarrow \mathcal{A}_1^{\tilde{H}}(\mathsf{sk}', \rho_{m_1})$.

The proof is almost the same as before, and we only sketch the main ideas here.

1. From the perspective of $\mathsf{A}'$, the reprogramming the oracle using $f$ will guarantee $\tilde{H}$ in Case (a) (and $H'$ in Case (b)) to produce $\mathsf{sk}'$ with non-zero probability. We argue that the fake secret key $\mathsf{sk}'$ will produce the real public key $|m_0\rangle$ with high probability: Since $\mathsf{B}$ is not affected by channel $\mathcal{T}$, we uncompute $\mathcal{B}$ on state $\rho_{A'B}$. As the uncomputation does not increase trace distance, we can see that the state of $(\mathsf{sk}', |m_0\rangle)$ is $\epsilon$-close to that of $(\mathsf{sk}, |m_0\rangle)$. Thus with probability $1-\epsilon$, $\mathsf{sk}'$ will produce $|m_0\rangle$.
2. From the perspective of Bob, the output state of $\mathcal{B}^H(|m_0\rangle)$ is a cq-state $\rho_B = \sum_{k_B} p_{k_B} |k_B\rangle\langle k_B| \otimes \rho_{k_B}$ where $\rho_{k_B}$ is the state of $m_1$ conditioned on $k_B$. Similarly for $\mathcal{B}^{\tilde{H}}(|m_0\rangle)$ in Case (a) (and $\mathcal{B}^{H'}(|m_0\rangle)$ in Case (b)), we express the output state as $\sigma_B = \sum_{k_B} p'_{k_B} |k_B\rangle\langle k_B| \otimes \sigma_{k_B}$. By using Lemma 2.5 as in the proof of Theorem 4.1, we will get $TD(\rho_B, \sigma_B) \le O(\epsilon)$. Let $\rho'_B := \sum_{k_B} p'_{k_B} |k_B\rangle\langle k_B| \otimes \rho_{k_B}$, $D_B := \{p_{k_B}\}$ and $D_{B'} := \{p_{k'_B}\}$. Then we have

  $$\begin{aligned} TD(\rho'_B, \sigma_B) \le & TD(\rho_B, \rho'_B) + TD(\rho_B, \sigma_B) = TV(D_B, D_{B'}) + TD(\rho_B, \sigma_B) \\ \le & 2TD(\rho_B, \sigma_B) \le O(\epsilon), \end{aligned}$$

  where the first inequality is the triangle inequality, the second inequality is because partial trace does not increase trace distance, and the last inequality is $TD(\rho_B, \sigma_B) \le O(\epsilon)$. Since $TD(\rho'_B, \sigma_B) = \mathbb{E}[TD(\rho_{k_B}, \sigma_{k_B})]$, by Markov inequality, we have

  $$\Pr_{k_B \leftarrow D_B}[TD(\rho_{k_B}, \sigma_{k_B}) \le \sqrt{\epsilon}] \ge 1 - O(\sqrt{\epsilon}).$$

By the above argument, with probability $1 - O(\sqrt{\epsilon})$, the tuple $(\mathsf{sk}', |m_0\rangle, k_B, \rho_{k_B}, R_E)$ is $O(\sqrt{\epsilon})$-close to a tuple $(\mathsf{sk}', |m_0\rangle, k_B, \sigma_{k_B}, R_E)$ which is compatible with oracle $\tilde{H}$ in Case (a) (and $H'$ is Case (b)). In Case (a), by perfect completeness, it follows that $k_E = k_B$ with probability $1 - O(\sqrt{\epsilon})$. In Case

(b), firstly for $k'_E \leftarrow \mathcal{A}_1^{H'}(\mathsf{sk}', \sigma_{k_B})$, we have $k'_E = k_B$ with probability $1 - O(\sqrt{\epsilon})$. We then apply Lemma 2.5 to the algorithm $\mathcal{A}_1(\mathsf{sk}', \sigma_{k_B})$ under oracles $\tilde{H}$ and $H'$ to conclude that $k_E = k_B$ with probability $1 - O(\sqrt{\epsilon})$. We remark that although Lemma 2.5 is stated for pure-state inputs, it also applies to the mixed-state input $\sigma_{k_B}$, since we can always assume the input to be the purification of $\sigma_{k_B}$.

Recall the IND-CPA security notion from Definition 2.14. If the public key is a pure state, the adversary algorithm $\mathcal{E}$ can obtain polynomial number of copies of $|\mathsf{pk}\rangle$. Given any two plaintexts $m_0 \neq m_1$, we can create a one-bit key agreement by designating the ciphertext as the second message (choosing between $\rho_{\mathsf{ct}_0}$ and $\rho_{\mathsf{ct}_1}$). Therefore, by executing our modified attack algorithm, we can break the IND-CPA security with an advantage of $1 - O(\sqrt{\epsilon})$.

**Theorem 4.5** (Restate of Theorem 1.4)**.** *For any perfect-complete QPKE with quantum public key in QROM, which makes $d$ queries to the random oracle $H : [2^n] \to \{0,1\}$ during each of* SKGen, Enc *and* Dec*, and no queries during* PKGen*, there exists an adversary Eve that can break the scheme with probability $1 - O(\sqrt{\epsilon})$ by making $O\left(d^7 \log(d/\epsilon)/\epsilon^4 + nd^2/\epsilon^2\right)$ queries to $H$.*

Theorem 1.3 is immediately implied by Theorem 4.5.

## References


[ACC+22] Per Austrin, Hao Chung, Kai-Min Chung, Shiuan Fu, Yao-Ting Lin, and Mohammad Mahmoody. On the impossibility of key agreements from quantum random oracles. In *Annual International Cryptology Conference*, pages 165–194. Springer, 2022. 2, 3

[AGL25] Prabhanjan Ananth, Aditya Gulati, and Yao-Ting Lin. Cryptography in the Common Haar State Model: Feasibility Results and Separations. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography*, pages 94–125, Cham, 2025. Springer Nature Switzerland. 4

[AHY23] Prabhanjan Ananth, Zihan Hu, and Henry Yuen. On the (im) plausibility of public-key quantum money from collision-resistant hash functions. In *International conference on the theory and application of cryptology and information security*, pages 39–72. Springer, 2023. 2

[AL70] Huzihiro Araki and Elliott H Lieb. Entropy inequalities. *Communications in Mathematical Physics*, 18(2):160–170, 1970. 10

[Alo99] Noga Alon. Combinatorial nullstellensatz. *Comb. Probab. Comput.*, 8(1–2):7–29, jan 1999. 7

[BB14] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science*, 560:7–11, 2014. 1

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. 7, 9, 18, 20

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. 11

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 406–435. Springer, 2021. 1

[BGH+23] Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-Key Encryption with Quantum Keys. In Guy Rothblum and Hoeteck Wee, editors, *Theory of Cryptography*, pages 198–227, Cham, 2023. Springer Nature Switzerland. 1, 3, 4

[BGVV23] Samuel Bouaziz, Alex B Grilo, Damien Vergnaud, and Quoc-Huy Vu. Towards the impossibility of quantum public key encryption with classical keys from one-way functions. *Cryptology ePrint Archive*, 2023. 2, 3

[Col23] Andrea Coladangelo. Quantum trapdoor functions from classical one-way functions. *arXiv preprint arXiv:2302.12821*, 2023. 1, 3, 4

[FR15] Omar Fawzi and Renato Renner. Quantum conditional mutual information and approximate markov chains. *Communications in Mathematical Physics*, 340(2):575–611, 2015. 2, 5, 10

[GLSV21] Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 531–561. Springer, 2021. 1

[GMMY24] Eli Goldin, Tomoyuki Morimae, Saachi Mutreja, and Takashi Yamakawa. CountCrypt: Quantum Cryptography between QCMA and PP, 2024. 4

[HY24] Akinori Hosoyamada and Takashi Yamakawa. Finding collisions in a quantum world: quantum black-box separation of collision-resistance and one-wayness. *Journal of Cryptology*, 37(4):35, 2024. 2

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Annual Structure in Complexity Theory Conference (SCT'95)*, SCT '95, page 134, USA, June 1995. IEEE Computer Society. 1

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 44–61, 1989. 1, 3

[KMNY24] Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum public-key encryption with tamper-resilient public keys from one-way functions. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VII*, page 93–125, Berlin, Heidelberg, 2024. Springer-Verlag. 1, 3, 4

[KQT25] William Kretschmer, Luowen Qian, and Avishay Tal. Quantum-Computable One-Way Functions without One-Way Functions. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 189–200, New York, NY, USA, June 2025. Association for Computing Machinery. 4

[LLLL24] Longcheng Li, Qian Li, Xingjian Li, and Qipeng Liu. How (not) to build qpke in minicrypt. In *Annual International Cryptology Conference*. Springer, 2024. 2, 3, 4, 5, 6, 8, 13, 24

[LLLL25] Longcheng Li, Qian Li, Xingjian Li, and Qipeng Liu. Toward the Impossibility of Perfect Complete Quantum PKE from OWFs. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Sciencshoe Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 71:1–71:16, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2, 3, 5, 6, 17

[LLPY23] Xingjian Li, Qipeng Liu, Angelos Pelecanos, and Takashi Yamakawa. Classical vs quantum advice and proofs under classically-accessible oracle. *arXiv preprint arXiv:2303.04298*, 2023. 2

[LMY25] Jiahui Liu, Saachi Mutreja, and Henry Yuen. QMA vs QCMA and pseudorandomness. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1520–1531, 2025. 2

[Mid04] Gatis Midrijanis. Exact quantum query complexity for total boolean functions, 2004. 7, 11

[MW24] Giulio Malavolta and Michael Walter. Robust quantum public-key encryption with applications to quantum key distribution. In *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part VII*, page 126–151, Berlin, Heidelberg, 2024. Springer-Verlag. 1, 3, 4

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition edition, 2010. 8

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999. 1

[Wie83] Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983. 1

[Wil11] Mark M Wilde. From classical to quantum shannon theory. *arXiv preprint arXiv:1106.1445*, 2011. 25

# A Missing Proofs in Section 3.1

For the convenience of readers, we provide the proofs of two lemmas from [LLLL24] that are used in the main text.

## A.1 Proof of Lemma 3.1

We can realize the quantum query unitary $U_H$ via a quantum communication protocol involving two parties: the algorithm $\mathcal{A}$ and Oracle. To execute $U_H$, the protocol proceeds as follows:

1. $\mathcal{A}$ sends both its input register and output register, $n+1$ qubits in total, to Oracle;

2. Oracle applies the unitary $U_H$ on these $n+1$ qubits and then returns them to $\mathcal{A}$.

By the subadditivity of entropy, each such quantum communication can increase the entropy of $\mathcal{A}$'s whole register by at most $2(n+1)$. In addition, applying local unitary $U_i$ does not change the entropy. Since $\mathcal{A}$'s register $\mathsf{A}$ initially contains a pure state (with zero entropy), it follows that $S(\mathsf{A})_\rho \leq 2d(n+1)$ after $d$ such rounds.

## A.2 Proof of Lemma 3.3

The following basic facts below will be used.

**Fact A.1** ([Wil11]). *If $\rho_{\mathsf{AB}}$ is a separable state, then $S(\mathsf{A}|\mathsf{B}) \geq 0$.*

**Fact A.2** (Chain rule). $I(\mathsf{B}_1, \mathsf{B}_2, \cdots, \mathsf{B}_t : \mathsf{A} \mid \mathsf{C}) = \sum_{i=1}^{t} I(\mathsf{B}_i : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \cdots, \mathsf{B}_{i-1})$.

*Proof of Lemma 3.3.* By the chain rule for conditional mutual information (Fact A.2), we have

$$\sum_{i=1}^{t} I(\mathsf{B}_i : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_{i-1}) = I(\mathsf{B}_1, \ldots, \mathsf{B}_t : \mathsf{A} \mid \mathsf{C}). \tag{6}$$

Moreover, we have

$$I(\mathsf{B}_1, \ldots, \mathsf{B}_t : \mathsf{A} \mid \mathsf{C}) = S(\mathsf{A} \mid \mathsf{C}) - S(\mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_t) \leq S(\mathsf{A} \mid \mathsf{C}) \leq S(\mathsf{A}), \tag{7}$$

where the inequalities follow from Fact A.1 and the non-negativity of $I(\mathsf{A} : \mathsf{C}) = S(\mathsf{A}) - S(\mathsf{A} \mid \mathsf{C})$. Combining (6) and (7), it follows that there exists some $i \in [t]$ for which

$$I(\mathsf{B}_i : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_{i-1}) \leq \frac{S(\mathsf{A})}{t}.$$

Finally, by permutation invariance, we have

$$I(\mathsf{B}_i : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_{i-1}) = I(\mathsf{B}_t : \mathsf{A} \mid \mathsf{C}, \mathsf{B}_1, \ldots, \mathsf{B}_{i-1}).$$

This completes the proof. □