Improved Quantum Lifting by Coherent Measure-and-Reprogram

Alexandru Cojocaru[1], Juan Garay[2], Qipeng Liu[3], and Fang Song[4]

[1] School of Informatics, University of Edinburgh, Edinburgh, UK.
a.cojocaru@ed.ac.uk
[2] Department of Computer Science and Engineering, Texas A&M University, College Station, USA.
garay@tamu.edu
[3] Department of Computer Science and Engineering, UC San Diego, La Jolla, USA.
qipengliu@ucsd.edu
[4] Department of Computer Science, Portland State University, Portland, USA.
fang.song@pdx.edu

Abstract. We give a tighter lifting theorem for security games in the quantum random oracle model. At the core of our main result lies a novel measure-and-reprogram framework that we call *coherent reprogramming*. This framework gives a tighter lifting theorem for query complexity problems, that only requires purely classical reasoning. As direct applications of our lifting theorem, we first provide a quantum direct product theorem in the average case — i.e., an enabling tool to determine the hardness of solving multi-instance security games. This allows us to derive in a straightforward manner the hardness of various security games, for example (i) the non-uniform hardness of salted games, (ii) the hardness of specific cryptographic tasks such as the multiple instance version of one-wayness and collision-resistance, and (iii) uniform or non-uniform hardness of many other games.

1 Introduction

Hash functions are a fundamental workhorse in modern cryptography. Efficient constructions such as SHA-2 and SHA-3 are widely used in real-world cryptographic applications. To facilitate the analysis of constructions based on hash functions, Bellare and Rogaway [BR93] proposed a framework known as the random oracle model (ROM). Recent development of quantum computing demands re-examining security against potential quantum attackers. The *quantum random oracle model* (QROM) has since been proposed by Boneh *et al.* [BDF+11] as an extension of ROM by taking into account quantum attackers. Various techniques have been developed for analyzing security in the QROM; however, they are often either *ad-hoc* (for specific scenarios) or too involved to apply.

In this paper we revisit a general tool for lifting security from ROM to QROM by Yamakawa and Zhandry [YZ21]. The lifting theorems are applicable to search games in (Q)ROM between a classical challenger interacting with

an adversary (e.g., think of an adversary that aims to find a preimage of 0 in the random oracle, with the challenger querying the random oracle to verify the adversary's answer). Roughly speaking, the lifting theorems assert that if a search game with a challenger performing a constant number of queries to the random oracle is hard against a classical adversary, then it is also hard against a quantum adversary in the QROM. Specifically, if the challenger performs k queries and if a quantum adversary performs q quantum queries and wins the search game with probability ϵ, then there exists a classical adversary performing only k classical queries winning with probability $\epsilon/(2q+1)^{2k}$.

This tool is particular powerful to establish quantum query lower bounds in the QROM. Let us consider function inversion from above for example; the goal is to find an input x, whose image equals 0. In this case, $k = 1$ and

$$\frac{\epsilon}{(2q+1)^{2k}} = \frac{\epsilon}{(2q+1)^2} \leq \frac{1}{N}.$$

This is because a single query reveals a pre-image of 0 with probability at most $1/N$. Therefore we have $\epsilon \leq (2q+1)^2/N$, which immediately reproduces the tight bound for the famous Grover's search problem [BBBV97]. However, for a k-search problem whose goal is to find k distinct inputs that all map to 0, the bound derived from [YZ21] is $O\left((kq)^2/N\right)^k$ for any quantum algorithm with kq queries, which has a large k^{2k} factor gap from the tight bound $\Theta\left(q^2/N\right)^k$ [5] . Similar weaknesses appear in a variety of problems involving multiple inputs.

In this work, we derive a new tighter lifting theorem for search games. If the challenger performs k queries and the quantum adversary performs q quantum queries and wins the search game with probability ϵ, then there exists a quantum adversary performing only k quantum queries and winning with probability $\epsilon/2^{2k}\binom{q+k}{k}^2$, improving on the previous lifting theorem by Yamakawa and Zhandry. Let us consider a (kq)-quantum-query algorithm for the previous k-search problem. Our bound (in this case, q in the theorem should be kq) gives the tight bound as below:

$$\frac{\epsilon}{2^{2k}\binom{kq+k}{k}^2} \leq \frac{1}{N^k} \quad \longleftrightarrow \quad \epsilon \leq \frac{2^{2k}\binom{kq+k}{k}^2}{N^k} \leq O\left(\frac{(q+1)^2}{N}\right)^k .$$

To achieve this, we develop a new measure-and-reprogram technique which is a key technical contribution of our work. The technique, which we call *coherent reprogramming*, improves on the recent results on adaptively reprogramming QRO on multiple points by Don *et al.* [DFM20, DFMS22] and Liu and Zhandry [LZ19], yielding tighter reprogramming bounds than the existing measure-and-reprogram proofs. As an immediate consequence, we are able to derive tighter

---

[5] We believe this is a folklore result that to our knowledge, this bound follows from a result in [CGK+23] (Theorem 3.1). Moreover, we would like to emphasize that since our main result is a strengthening of the lifting Lemma of [YZ21], we can also show that our result concerning the bound of this problem is stronger than the bound derived from [YZ21].

quantum hardness bounds for many applications, such as direct-product theorems, salted security, and non-uniform security.

1.1 Summary of Our Results

Lifting Theorem for Search Games. Our central result is a new lifting theorem for search games that relates (and upper bounds) the success probability of an arbitrary quantum algorithm to the success probability of a quantum algorithm performing a small number of queries to the RO. More formally:

Theorem 1 (Quantum Lifting Theorem (Informal)). *Let $\mathcal{G}$ be a search game with a classical challenger $\mathcal{C}$ that performs at most k queries to the RO, and let $\mathcal{A}$ be a q-quantum query adversary in the game $\mathcal{G}$ (against the k-classical query challenger $\mathcal{C}$). Then there exists a k-quantum query adversary $\mathcal{B}$ such that:*

$$\Pr[\mathcal{B} \text{ wins } \mathcal{G}] \geq \frac{1}{2^{2k}\binom{q+k}{k}^2} \Pr[\mathcal{A} \text{ wins } \mathcal{G}].$$

Remark 1. Comparing to the lifting theorem in [YZ21], we have a better loss $2^{2k}\binom{q+k}{k}^2$, whereas it is $(2q+1)^k$ in their work. Since the algorithm often makes more queries than the challenger $q \gg k$, it is roughly a $(k!)^2$ save. In [YZ21], they are able to reduce a q-quantum-query algorithm to a k-classical-query algorithm; whereas in this work, we only reduce the number of the queries, with the algorithm $\mathcal{B}$ still making quantum queries. Nonetheless, it does not affect the applications and improvement we have in this work. Our framework handles the case where the challenge is independent of the oracle (similar to the results in [YZ21]). We leave the case of oracle-dependent challenges as an interesting open question.

Coherent Reprogramming. At the core of our main lifting result above lies a new framework for quantum reprogramming which we call *coherent reprogramming*. This new framework has the following advantages:

1. It simplifies the proofs and frameworks of existing quantum reprogramming results;
2. It yields improved tighter reprogramming bounds; and
3. It implies in a straightforward manner several applications in quantum query complexity and cryptography.

In order to present our main coherent reprogramming result, we first need to introduce a few notions. For an oracle H we call $H_{x,y}$ the reprogrammed oracle that behaves almost like the original function H, with the only difference that its value on input x will be y. Similarly, we define the reprogrammed oracle on k inputs $\boldsymbol{x} = (x_1, ..., x_k)$ and k corresponding outputs $\boldsymbol{y} = (y_1, ..., y_k)$, denoted by $H_{\boldsymbol{x},\boldsymbol{y}}$, as the original function H with the only difference that for every input x_i in $\boldsymbol{x}$, the corresponding image will be y_i in $\boldsymbol{y}$.

Theorem 2 (Coherent Reprogramming (Informal)). *Let H, G be two random oracles. Let $\mathcal{A}$ be any q-quantum query algorithm to the oracle H, and let $\boldsymbol{x}_\circ = (x_1, ..., x_k) \in X^k$ be any k-vector of inputs and $\boldsymbol{y}_\circ = (y_1, ..., y_k) = (G(x_1), ..., G(x_k))$. Then there exists a simulator algorithm* Sim *that given oracle access to H, G and $\mathcal{A}$, simulates the output of $\mathcal{A}$ having oracle access to $H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}$ (the reprogrammed version of H) with probability:*

$$\Pr_{H,G} \left[\textsf{Sim } outputs\ correct\ (\boldsymbol{x}, \boldsymbol{y}) \right] \geq \frac{1}{2^{2k} \binom{q+k}{k}^2} \cdot \Pr_{H,G} \left[\mathcal{A}^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}} outputs\ correct\ (\boldsymbol{x}, \boldsymbol{y}) \right].$$

where "correct" is defined with respect to some predicate that can depend on the reprogrammed oracle $H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}$.

Remark 2. Similar to the comparison between Theorem 1 and [YZ21], the second theorem improves the factor $(2q + 1)^k$ in [DFM20] to $2^{2k} \binom{q+k}{k}^2$. Our simulator does not measure and reprogram directly, but does everything coherently (or in superposition).

Next, we show the applications of our lifting theorem in query complexity and cryptography.

Quantum Lifting Theorem with Classical Reasoning. A multi-output k-search game between a challenger and an adversary is defined as follows. The adversary receives k different challenges from the challenger, and at the end of their interaction, the adversary needs to respond with k outputs. If the k outputs (taken together) satisfy some relation R specified by the game, we say the adversary wins the multi-output k-search game . The goal of the *lifting theorem* is to establish the hardness of solving the multi-output k-search game by any general quantum adversary, with only simple classical reasoning. For an arbitrary k-ary relation R, let $\mathcal{S}_k$ be the symmetric group on $[k]$ and we define:

$$p(R) := \Pr[\exists \pi \in \mathcal{S}_k \mid (y_{\pi(1)}, y_{\pi(2)}, ..., y_{\pi(k)}) \in R : (y_1, ..., y_k) \xleftarrow{\$} Y^k].$$

Note that $p(R)$ is a quantity that only depends on the game itself, and can be calculated with only classical reasoning.

Theorem 3 (Quantum Lifting Theorem with Classical Reasoning (Informal)). *For any quantum algorithm $\mathcal{A}$ equipped with q quantum queries to a random oracle $H : X \rightarrow Y$, $\mathcal{A}$'s success probability to solve the multi-output k-search game as specified by the winning relation R, is bounded by:*

$$\Pr[\mathcal{A} \text{ wins multi-output } k\text{-search game }] \leq 2^{2k} \binom{q+k}{k}^2 \cdot p(R).$$

Our lifting theorem translates into the following quantum hardness results for our applications in query complexity and cryptography.

Direct Product Theorem. We give the first direct product theorem (DPT) in the average case (in the QROM). Previously, only worst-case quantum DPTs were

known [She11, LR13] and were proof-method dependent; until recently, Dong et al. [DLW24] shows the first average-case quantum DPTs for some problems in the QROM. While they are non-tight, our DPTs works for all games in the QROM and proof-method independent.

Our direct product theorem establishes the hardness of solving g independent instances (each instance is associated with an independent oracle) of a game $\mathcal{G}$ given a total of $g \cdot q$ quantum queries:

Theorem 4 (Direct Product Theorem). *For any quantum algorithm $\mathcal{A}$ equipped with $g \cdot q$ quantum queries, $\mathcal{A}$'s success probability to solve the Direct Product game $\mathcal{G}^{\otimes g}$ with the underlying $\mathcal{G}$ specified by the winning relation R, is bounded by*

$$\Pr[\mathcal{A} \text{ wins } \mathcal{G}^{\otimes g}] \leq \left(2^{2k} \binom{q+k}{k}^2 p(R) \right)^g .$$

Non-uniform Security of Salting. The above theorem directly implies non-uniform security of salting. Non-uniform attacks allow a malicious party to perform heavy computation offline and attack a protocol much more efficiently, using the information in the offline stage. Salting is a generic method that prevent non-uniform attacks against hash functions. Chung et al. [CGLQ20] shows that "salting generically defeats quantum preprocessing attacks"; they show that if a game in the QROM is $\epsilon(q)$ secure, the salted game with salt space $[K]$ is $\epsilon(q) + \frac{Sq}{K}$ secure against a quantum adversary with S-bit of advice. Their bound is non-tight, since when the underlying game is collision-finding, the tight non-uniform security should be $\epsilon(q) + \frac{S}{K}$. Improving the additive factor is an interesting open question and until recently [DLW24] is able to answer this question affirmatively for a limited collection of games.

Using our direct product theorem, we show:

Theorem 5 (Another "Salting Defeats Quantum Preprocessing"). *For any non-uniform quantum algorithm $\mathcal{A}$ equipped with q quantum queries and S-bit of classical advice, $\mathcal{A}$'s success probability to solve the salted game $\mathcal{G}_s$ with the underlying $\mathcal{G}$ specified by the winning relation R, is bounded by*

$$\Pr[\mathcal{A} \text{ wins } \mathcal{G}_s] \leq 4 \cdot \left(2^{2k} \binom{q+k}{k}^2 p(R) + \frac{S}{K} \right) .$$

Our bound is incomparable to that in [CGLQ20]. We are able to improve the additive term from $\frac{Sq}{K}$ to $\frac{S}{K}$, while only able to give an upper bound for $\epsilon(q)$. Even our bound is non-tight in general, it still confirms (on a high level) that the help from classical advice only comes from the following:

– using S-bit advice to store solutions for S random salts;
– if the challenge salt matches with the random salts (with probability $\frac{S}{K}$), the attack succeeds; otherwise, proceed the attack as if there is no advice.

Non-Uniform Security. By combining our lifting theorem with the results by Chung *et al.* [CGLQ20], we derive the following results concerning the security (hardness) against non-uniform quantum adversaries with classical advice, for a broader class of games.

Lemma 1 (Security against Quantum Non-Uniform Adversaries (Informal)). *Let $\mathcal{G}$ be any classically verifiable search game specified by the winning relation R. Let $R^{\otimes S}$ be the winning relation of the multi-instance game of $\mathcal{G}$. Any quantum non-uniform algorithm $\mathcal{A}$ equipped with q quantum queries and S classical bits of advice, can win $\mathcal{G}$ with probability at most:*

$$\Pr[\mathcal{A} \text{ wins } \mathcal{G}] \leq 4 \cdot 2^{2k} \binom{S(q+k)}{Sk}^{\frac{2}{S}} \cdot p(R^{\otimes S})^{1/S}.$$

To demonstrate the power of our results, we also apply them to the hardness of three concrete cryptographic tasks: the multiple instance versions of one-wayness, collision resistance and search, as described next. Note that the applications we list below are non-exhaustive, given $p(R)$ is easy to define for almost every game.

Hardness of Multi-Image Inversion. Firstly, we can analyze the quantum hardness of inverting k different images of a random oracle $H : [M] \to [N]$.

Our first result establishes the quantum hardness of multi-image inversion, which is a tight bound as already proven in [CGLQ20], but achieved here in a much simpler way, directly from our quantum lifting theorem.

Lemma 2 (Quantum Hardness of Multi-Image Inversion (Informal)). *For any distinct $\boldsymbol{y} = (y_1, ..., y_k) \in [N]^k$ and for any q-quantum query algorithm $\mathcal{A}$ whose aim is to invert all the images in $\boldsymbol{y}$, the success probability of $\mathcal{A}$ is upper bounded by:*

$$\Pr_H[\mathcal{A}(\boldsymbol{y}) \to \boldsymbol{x} = (x_1, ..., x_k) \ : \ H(x_i) = y_i \ \forall i \in [k]] \leq 2^{2k} \binom{q+k}{k}^2 \cdot \frac{k!}{N^k}.$$

Hardness of Multi-Collision Finding. Secondly, we can analyse the quantuanalyzeess of finding k collisions, namely, k inputs that map to the same image of the random oracle $H : [M] \to [N]$. We can also determine upper bounds for solving the salted version of this task, as well as the hardness of finding a collision for quantum algorithms that are also equipped with advice.

Lemma 3 (Quantum Hardness of Multi-Collision Finding and Salted Multi-Collision Finding (Informal)). *For any q-quantum query algorithm $\mathcal{A}$, the probability of solving the k-multi-collision problem is at most:*

$$\Pr_H[\mathcal{A}() \to \boldsymbol{x} = (x_1, ..., x_k) \ : \ H(x_1) = ... = H(x_k)] \leq \frac{1}{N^{k-1}} \left[\frac{2e(q+k)}{k} \right]^{2k}.$$

6

Any quantum algorithm $\mathcal{A}$ equipped with q quantum queries and S-bit of classical advice can win the salted multi-collision finding game with salted space $[K]$ with probability at most:

$$\Pr_{H}[\mathcal{A}() \to \boldsymbol{x} = (x_1, ..., x_k) \ : \ H(x_1) = ... = H(x_k)] \leq \frac{4}{N^{k-1}} \left[\frac{2e(q+k)}{k} \right]^{2k} + \frac{4S}{K}.$$

The above bounds become $O(q^4/N)$ and $O(q^4/N + S/K)$ respectively, for $k = 2$ (the standard collision-finding). Previous work [YZ21] achieves the same uniform bound, but only achieves $O((Sq)^4/N + S/K)$, due to the extra loss in their lifting theorem.

Hardness of Multi-Search. Finally, we also establish a tight bound for finding k distinct inputs that all map to 0 under the random oracle $H : [M] \to [N]$. This is potentially useful in analyzing proofs-of-work in the blockchain context [GKL15].

Lemma 4 (Quantum Hardness of Multi-Search). *For any q-quantum query algorithm $\mathcal{A}$ whose task is to find k different preimages of 0 of the random oracle, the success probability of $\mathcal{A}$ is upper bounded by:*

$$\Pr_{H}[\mathcal{A}() \to \boldsymbol{x} = (x_1, ..., x_k) \ : \ H(x_i) = 0 \ \forall i \in [k]] \leq \left[\frac{4e^2(q+k)^2}{Nk^2} \right]^{k}.$$

1.2 Related Work

The measure-and-reprogram framework was proposed, and subsequently generalized and improved with tighter bounds in the works of [DFM20, LZ19, DFMS22, GHHM21]. A main application of the framework has been in the context of the Fiat-Shamir transformation, with several works establishing its post-quantum security [Cha19, DFMS19, AFK22, AFKR23, GOP+23]. Other cryptographic applications of measure-and-reprogram have been considered in [Kat21, BKS21, ABKK23, JMZ23, DFHS23, KX24]. Finally, applications in query complexity of the framework have been developed in [CGLQ20, YZ21].

2 Preliminaries

Notation. For two vectors $\boldsymbol{x}, \boldsymbol{x}' \in X^k$, we say $\boldsymbol{x} \equiv \boldsymbol{x}'$ if and only if there exists a permutation σ over the indices $\{1, 2, \ldots, k\}$ such that $x_i' = x_{\sigma(i)}$. For a function $H : X \to Y$ and $\boldsymbol{x} \in X^k$, $H(\boldsymbol{x})$ is defined as $(H(x_1), H(x_2), \ldots, H(x_k))$. We say $x \in \boldsymbol{x}$, if $x = x_i$ for some $i \in [k]$.

2.1 Quantum Query Algorithms

We will denote a quantum query algorithm by $\mathcal{A}$. Let q be the total number of quantum queries of $\mathcal{A}$. By $\mathcal{A}^H$ we mean that $\mathcal{A}$ has quantum access to the function H.

A quantum oracle query to H will be applied as the unitary $O_H: O_H \ket{x} \ket{y} \longrightarrow \ket{x} \ket{y \oplus H(x)}$. Without loss of generality, we assume an algorithm will never perform any measurement (until the very end) and thus the internal state is always pure. We use $\ket{\phi_i^H}$ to denote the algorithm $\mathcal{A}$'s internal (pure) state right after the i-th query.

$$\ket{\phi_i^H} = O_H U_i \cdots O_H U_1 \ket{0} .$$

Specifically, we have,

– $\ket{\phi_0^H} = \ket{0}$ is the initial state of $\mathcal{A}$;
– $\ket{\phi_q^H}$ is the final state of $\mathcal{A}$.

Without loss of generality, the algorithm will have three registers $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ at the end of the computation, where $\mathcal{X}$ consists of a list of inputs, $\mathcal{Y}$ consists of a list of outputs corresponding to these inputs and some auxiliary information in $\mathcal{Z}$.

Definition 1 (Reprogrammed Oracle). *Reprogram oracle H to output y on input x, results in the new oracle, defined as:*

$$H_{x,y}(z) = \begin{cases} y, & \text{if } z = x \\ H(z), & \text{otherwise.} \end{cases}$$

We can similarly define a multi-input reprogram oracle $H_{\boldsymbol{x},\boldsymbol{\Theta}}$ for $\boldsymbol{x} \in X^k$ without duplicate entries and $\boldsymbol{\Theta} \in Y^k$:

$$H_{\boldsymbol{x},\boldsymbol{\Theta}}(z) = \begin{cases} \Theta_i, & \text{if } z = x_i \\ H(z), & \text{otherwise.} \end{cases}$$

2.2 Quantum Measure-and-Reprogram Experiment

We recall the measure-and-reprogram experiment and the state-of-the-art results here, first proposed by [DFM20] and later adapted by [YZ21].

Definition 2 (Measure-and-Reprogram Experiment). *Let $\mathcal{A}$ be a q-quantum query algorithm that outputs $\boldsymbol{x} \in X^k$ and $z \in Z$. For a function $H : X \to Y$ and $\boldsymbol{y} = (y_1, ..., y_k) \in Y^k$, define a measure-and-reprogram algorithm $\mathcal{B}[H, \boldsymbol{y}]$:*

1. *For each $j \in [k]$, uniformly pick $(i_j, b_j) \in ([q] \times \{0, 1\}) \cup \{(\bot, \bot)\}$ such that there does not exist $j \neq j'$ such that $i_j = i_{j'} \neq \bot$;*
2. *Run $\mathcal{A}^O$ where the oracle O is initialized to be a quantumly accessible classical oracle that computes H and when $\mathcal{A}$ makes its i-th query, the oracle is simulated as follows:*
   (a) *If $i = i_j$ for some $j \in [k]$, measure $\mathcal{A}$'s query register to obtain x'_j and do either of the following:*
      i. *If $b_j = 0$, reprogram O using (x'_j, y_j) and answer $\mathcal{A}$'s i_j-th query using the reprogrammed oracle;*

      ii. If $b_j = 1$, answer $\mathcal{A}$'s i_j-th query using oracle before reprogramming and then reprogram O using (x'_j, y_j);

    (b) Else, answer $\mathcal{A}$'s i-th query by just using the oracle O without any measurement or reprogramming;

3. *Let $(\boldsymbol{x} = (x_1, ..., x_k), z)$ be $\mathcal{A}$'s output;*
4. *For all $j \in [k]$ such that $i_j = \perp$, set $x'_j = x_j$*
5. *Output $\boldsymbol{x'} := ((x'_1, ..., x'_k), z)$*

We next state the current state-of-the-art quantum measure-and-reprogram result.

Lemma 5 (Quantum Measure-and-Reprogram (adaptation from [DFM20, YZ21])). *For any $H : X \to Y$, for any $\boldsymbol{x}^* = (x_1^*, ..., x_k^*) \in X^k$ without duplicated entries, for all $\boldsymbol{y} = (y_1, ..., y_k)$ and any relation $R \subseteq X^k \times Y^k \times Z$, we have:*

$$\Pr[\boldsymbol{x'} = \boldsymbol{x}^* \wedge (\boldsymbol{x'}, \boldsymbol{y}, z) \in R \mid (\boldsymbol{x'}, z) \leftarrow \mathcal{B}[H, y]]$$
$$\geq \frac{1}{(2q+1)^{2k}} \Pr[\boldsymbol{x} = \boldsymbol{x}^* \wedge (\boldsymbol{x}, \boldsymbol{y}, z) \in R \mid (\boldsymbol{x}, z) \leftarrow \mathcal{A}^{H_{\boldsymbol{x}^*, \boldsymbol{y}}}]$$

where $\mathcal{B}[H, y]$ is the measure-and-reprogram experiment.

2.3 Predicates and Success Probabilities

Definition 3 (Predicate/Verification Projection/Symmetric Predicate). *Let R be a relation on $X^k \times Y^k \times Z$. A predicate $V^H(\boldsymbol{x}, \boldsymbol{y}, z)$ parameterized by an oracle H, returns 1 if and only if $(\boldsymbol{x}, \boldsymbol{y}, z) \in R$ and $H(x_i) = y_i$ for every $i \in \{1, 2, \ldots, k\}$.*

Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be the registers that store $\boldsymbol{x}, \boldsymbol{y}, z$, respectively. We define Π_V^H as the projection corresponding to V^H:

$$\Pi_V^H \ket{\boldsymbol{x}, \boldsymbol{y}, z} = \begin{cases} \ket{\boldsymbol{x}, \boldsymbol{y}, z} & \text{if } V^H(\boldsymbol{x}, \boldsymbol{y}, z) = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Finally, for any predicate V^H, we are able to establish the success probability using the projection Π_V^H.

Definition 4 (Success Probability). *Let $\mathcal{A}$ a quantum query algorithm. Its success probability of outputting $\boldsymbol{x}, \boldsymbol{y}, z$ such that $H(\boldsymbol{x}, \boldsymbol{y}, z) = 1$ is defined by*

$$\Pr\left[\mathcal{A}^H \to (\boldsymbol{x}, \boldsymbol{y}, z) \text{ and } V^H(\boldsymbol{x}, \boldsymbol{y}, z) = 1\right] = \left\| \Pi_V^H \ket{\phi_q^H} \right\|^2.$$

(Recall that $\ket{\phi_q^H}$ is the final state of $\mathcal{A}$.)

Sometimes, we care about the event that $\mathcal{A}$ outputs a particular $\boldsymbol{x}$ and still succeeds. For any $\boldsymbol{x}_\circ$, the following probability denotes that $\mathcal{A}$ outputs $\boldsymbol{x} \equiv \boldsymbol{x}_\circ$ and succeeds:

$$\Pr\left[\mathcal{A}^H \to (\boldsymbol{x}, \boldsymbol{y}, z), \ \ \boldsymbol{x} \equiv \boldsymbol{x}_\circ \text{ and } V^H(\boldsymbol{x}, \boldsymbol{y}, z) = 1\right] = \left\| G_{\boldsymbol{x}_\circ} \Pi_V^H \ket{\phi_q^H} \right\|^2,$$

where $G_{\boldsymbol{x}_\circ}$ is defined as the projection that checks whether $\mathcal{A}$ consists of $\boldsymbol{x} \equiv \boldsymbol{x}_\circ$.

3 Coherent Measure-and-Reprogram

In this section, we give our main theorem: the coherent measure-and-reprogram theorem. A main difference between our theorem and the previous measure-and-reprogram theorem [DFM20] is that our simulator needs to make quantum queries, instead of classical queries, which is potentially required by the coherent nature of our simulator and gives tighter reprogramming bounds for many applications. While this makes the simulator slightly more complicated, it yields improved bounds on the various applications that we mention in the next section.

3.1 Main Theorem

We give our main theorem below.

Theorem 6. *Let $H, G : \{0,1\}^m \to \{0,1\}^n$ be two functions $X \to Y$. Let k be a positive integer (can be a computable function in both n and m). There exists a black-box quantum algorithm $\mathsf{Sim}^{H,G,\mathcal{A}}$, satisfying the properties below. Let V^H be any predicate defined over $X^k \times Y^k \times Z$. Let $\mathcal{A}$ be any q-quantum query algorithm to the oracle H. Then for any $\boldsymbol{x}_\mathsf{o} \in X^k$ without duplicate entries and $\boldsymbol{y}_\mathsf{o} = G(\boldsymbol{x}_\mathsf{o})$, we have,*

$$\Pr_{H,G}\left[\mathsf{Sim}^{H,G,\mathcal{A}} \to (\boldsymbol{x}, \boldsymbol{y}, z) \text{ and } \boldsymbol{x} \equiv \boldsymbol{x}_\mathsf{o} \text{ and } V^{H_{\boldsymbol{x}_\mathsf{o}, \boldsymbol{y}_\mathsf{o}}}(\boldsymbol{x}, \boldsymbol{y}, z) = 1\right]$$

$$\geq \frac{1}{2^{2k}\binom{q+k}{k}^2} \cdot \Pr_{H,G}\left[\mathcal{A}^{H_{\boldsymbol{x}_\mathsf{o}, \boldsymbol{y}_\mathsf{o}}} \to (\boldsymbol{x}, z) \text{ and } \boldsymbol{x} \equiv \boldsymbol{x}_\mathsf{o} \text{ and } V^{H_{\boldsymbol{x}_\mathsf{o}, \boldsymbol{y}_\mathsf{o}}}(\boldsymbol{x}, H_{\boldsymbol{x}_\mathsf{o}, \boldsymbol{y}_\mathsf{o}}(\boldsymbol{y}), z) = 1\right].$$

Furthermore, Sim makes exactly k quantum queries to G and has a running time polynomial in n, m, k and the running time of $\mathcal{A}$.

Before formally defining our simulator, we introduce one more notation: controlled reprogrammed oracle queries. That is, an oracle query will be reprogrammed by a list of input and output pairs in a control register.

Definition 5 (Controlled Reprogrammed Oracle Query). *For every $x \in X, y \in Y$, every $\ell > 0$ and $\boldsymbol{x} \in X^\ell$ without duplicated entries, $\Theta \in Y^\ell$, controlled reprogrammed oracle O_H^{ctrl} acts as below.*

$$O_H^{\mathsf{ctrl}} |x\rangle |y\rangle |\boldsymbol{x}, \boldsymbol{\Theta}\rangle = \left(O_{H_{\boldsymbol{x}, \boldsymbol{\Theta}}} |x\rangle |y\rangle\right) |\boldsymbol{x}, \boldsymbol{\Theta}\rangle$$

Its behavior on $|\boldsymbol{x}\rangle$ with duplicated entries can be arbitrarily defined as long as unitarity is maintained, as this case will never occur in the simulator or our analysis.

We define our simulator used in Theorem 6, as follows:

Definition 6 (Coherent Measure-and-Reprogram Experiment). *Let $\mathcal{A}$ be a $(q + k)$-quantum query algorithm that outputs $\boldsymbol{x} = (x_1, \ldots, x_k) \in X^k, \boldsymbol{y} \in Y^k$ and $z \in Z$. We assume $\boldsymbol{y}$ is always computed by $H(\boldsymbol{x})$, using the last k queries. For a function $H : X \to Y$ and $G : X \to Y$, define a measure-and-reprogram algorithm $B[H, G]$:*

1. *Pick a uniformly random subset $\boldsymbol{v}$ of $[q + k]$, of length k. We have $1 < v_1 < \cdots < v_k \leq q + k$. Pick $\boldsymbol{b} \in \{0, 1\}^k$ uniformly at random.*
2. *Run $\mathcal{A}$ with an additional control register $\mathcal{R}$, initialized as empty $|\emptyset\rangle$. Define the following operation U that updates the control register: for x that is not in L,*

$$U |x\rangle |L\rangle_\mathcal{R} \leftarrow |x\rangle |L \cup (x, G(x))\rangle_\mathcal{R}.$$

   Here L is the set of input and output pairs. Since we will only work with basis states $|x\rangle |L\rangle$ whose x is not in L, U clearly can be implemented by a unitary (by assuming that the list is initialized as empty).
3. *When $\mathcal{A}$ makes its i-th query,*
   (a) *If $i = v_j$ for some $j \in [k]$, do either of the following:*
      i. *If $b_j = 0$, <u>update $\mathcal{R}$ using the input register and G</u>, and make the i-th query to H controlled by $\mathcal{R}$ (see O_H^{ctrl} above);*
      ii. *If $b_j = 1$, make the i-th query to H controlled by $\mathcal{R}$ and <u>update $\mathcal{R}$ using the input register and G</u>.*
      iii. *Before updating the control register, it <u>checks coherently that the input register is not contained in the control register</u>; otherwise, it aborts.*
   (b) *Else, answer $\mathcal{A}$'s i-th query controlled by $\mathcal{R}$;*
4. *Let $(\boldsymbol{x}, \boldsymbol{y}, z)$ be $\mathcal{A}$'s output;*
5. *<u>Measure $\mathcal{R}$ register to obtain $L = (\boldsymbol{x}', \boldsymbol{\Theta}')$.</u>*
6. *<u>Output $(\boldsymbol{x}, \boldsymbol{y}, z)$ if $\boldsymbol{x}' \equiv \boldsymbol{x}$</u>; otherwise, abort.*

At a high level, our simulator resembles that in Definition 2; instead of measuring $\mathcal{A}$'s queries, we put it into a separate register (a.k.a., measure the queries coherently). With the "controlled reprogrammed oracle query", we are still able to progressively reprogram the oracle and run the algorithm with (an) updated oracle(s). The ability of coherently measuring and reprogramming, makes all the improvement (mentioned in the later sections) possible.

Proof of Theorem 6. Before we start with the proof, we first recall and introduce some notations. Fix any $\boldsymbol{x} \in X^k$ without duplicate entries and $\boldsymbol{\Theta} \in Y^k$. Recall that in Section 2.1, $\left|\phi_q^{H_{\boldsymbol{x},\boldsymbol{\Theta}}}\right\rangle$ is the state of the algorithm $\mathcal{A}$ after making all its queries to $H_{\boldsymbol{x},\boldsymbol{\Theta}}$. More precisely, it is:

$$\left|\phi_q^{H_{\boldsymbol{x},\boldsymbol{\Theta}}}\right\rangle = O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} U_q \cdots O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} U_1 |0\rangle.$$

In the next step, we decompose this quantum state, so that each component corresponds to one of the cases in the quantum simulator Definition 6.

11

The first query. We start by considering the state up to the first query: $O_{H_{x,\Theta}} U_1 |0\rangle$. We insert an additional identity operator and have,

$$O_{H_{x,\Theta}} U_1 |0\rangle = O_{H_{x,\Theta}} I U_1 |0\rangle$$

$$\stackrel{(i)}{=} O_{H_{x,\Theta}} \left(I - \sum_{x_j} |x_j\rangle\langle x_j| + \sum_{x_j} |x_j\rangle\langle x_j| \right) U_1 |0\rangle$$

$$= O_{H_{x,\Theta}} \left(I - \sum_{x_j} |x_j\rangle\langle x_j| \right) U_1 |0\rangle + O_{H_{x,\Theta}} \left(\sum_{x_j} |x_j\rangle\langle x_j| \right) U_1 |0\rangle$$

$$\stackrel{(ii)}{=} O_H \left(I - \sum_{x_j} |x_j\rangle\langle x_j| \right) U_1 |0\rangle + \sum_{x_j} O_{H_{x_j,\Theta_j}} |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$= \underbrace{O_H U_1 |0\rangle}_{(1)} - \sum_{x_j} \underbrace{O_H |x_j\rangle\langle x_j| U_1 |0\rangle}_{(2)} + \sum_{x_j} \underbrace{O_{H_{x_j,\Theta_j}} |x_j\rangle\langle x_j| U_1 |0\rangle}_{(3)}$$

Above, x_j is enumerated over all entries in x.

Line (i) follows easily. Line (ii) is due to the fact that, if the query input is not in x, $H_{x,\Theta}$ is functionally equivalent to H; similarly, if the query input is x_j, $H_{x,\Theta}$ is functionally equivalent to H_{x_j,Θ_j}.

Next, we look at the three terms (1), (2), (3):

(1) $O_H U_1 |0\rangle$ corresponds to the case that no measurement happens for the first query.

(2) $O_H |x_j\rangle\langle x_j| U_1 |0\rangle$ corresponds to the case that measurement is made at the first query and the query input is x_j; the oracle is not reprogrammed immediately. In other words, the case $(v_1, b_1) = (1, 1)$ in the simulator.

(3) $O_{H_{x_j,\Theta_j}} |x_j\rangle\langle x_j| U_1 |0\rangle$ corresponds to the case that measurement is made at the first query and the query input is x_j; the oracle is reprogrammed immediately and used for the first query. In other words, the case $(v_1, b_1) = (1, 0)$ in the simulator.

The second query. We do the same: insert an additional identity operator. To make the presentation clearer, we focus only on one term $O_H |x_j\rangle\langle x_j| U_1 |0\rangle$; the other cases are simpler.

12

$$O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$\overset{(1)}{=} O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} \left(I - \sum_{x_k \neq x_j} |x_k\rangle\langle x_k| + \sum_{x_k \neq x_j} |x_k\rangle\langle x_k| \right) U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$\overset{(2)}{=} O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} \left(I - \sum_{x_k \neq x_j} |x_k\rangle\langle x_k| \right) U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$+ O_{H_{\boldsymbol{x},\boldsymbol{\Theta}}} \left(\sum_{x_k \neq x_j} |x_k\rangle\langle x_k| \right) U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$\overset{(3)}{=} O_{H_{x_j,\Theta_j}} \left(I - \sum_{x_k \neq x_j} |x_k\rangle\langle x_k| \right) U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$+ \left(\sum_{x_k \neq x_j} O_{H_{(x_j,x_k),(\Theta_j,\Theta_k)}} |x_k\rangle\langle x_k| \right) U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle$$

$$= \underbrace{O_{H_{x_j,\Theta_j}} U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle}_{(1)} - \sum_{x_k \neq x_j} \underbrace{O_{H_{x_j,\Theta_j}} |x_k\rangle\langle x_k| U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle}_{(2)}$$

$$+ \sum_{x_k \neq x_j} \underbrace{O_{H_{(x_j,x_k),(\Theta_j,\Theta_k)}} |x_k\rangle\langle x_k| U_2 O_H |x_j\rangle\langle x_j| U_1 |0\rangle}_{(3)}$$

We explain the equations line by line:

1. This one is straightforward by realizing the summation inside the bracket is an identity operator.
2. This one follows from the distributive property.
3. This is the most important one.
   - For the first term, we realize that the oracle will only be applied to inputs that are not in $\boldsymbol{x}$, or are equal to x_j. Thus, $H_{\boldsymbol{x},\boldsymbol{\Theta}}$ is functionally equivalent to H_{x_j,Θ_j}.
   - For the second term, the oracle will only be applied to inputs that are equal to x_k. Thus, $H_{\boldsymbol{x},\boldsymbol{\Theta}}$ is functionally equivalent to $H_{(x_j,x_k),(\Theta_j,\Theta_k)}$[6].

(1) corresponds to the case that no measurement happens for the second query, but since the first query is measure-and-reprogrammed, the second query is made with the oracle H_{x_j,Θ_j}. In other words, the case $(v_1, b_1) = (1, 1)$.
(2) corresponds to the case that measurement is made at the second query and the query input is x_k; the oracle is not reprogrammed immediately. In other words, the case $(v_1, b_1) = (1, 1)$ and $(v_2, b_2) = (2, 1)$ in the simulator.

---

[6] It is also equivalent to H_{x_k,Θ_k}. However, due to our description of the simulator, $H_{(x_j,x_k),(\Theta_j,\Theta_k)}$ is more natural to work with.

(3) corresponds to the case that measurement is made at the second query and the query input is x_k; the oracle is reprogrammed immediately. In other words, the case $(v_1, b_1) = (1, 1)$ and $(v_2, b_2) = (2, 0)$ in the simulator.

Generalization to all queries — state decomposition. By repeating the same state decomposition up to the first q queries (instead of all $q + k$ queries), we will end up a collection of subnormalized states, who sum up to the original state $\left|\phi_q^{H_x,\Theta}\right\rangle$. These states are parameterized by when the measurement happens (an ordered vector $\boldsymbol{v}$ such that $1 \leq v_1 \cdots \leq v_t \leq q$), whether these queries are made before or after each reprogramming ($\boldsymbol{b} \in \{0,1\}^t$), and $t \in \{0, \ldots, q\}$; in the following we will denote these states by $|\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle$. For example, assuming $\boldsymbol{b} = \boldsymbol{0}$ (all reprogramming happens immediately), we have,

$$|\phi_{\boldsymbol{v},\boldsymbol{0}}\rangle = \sum_{\sigma \in S_t^k} O_{H_{\boldsymbol{x}_\sigma, \Theta_\sigma}} U_q \cdots O_{H_{\boldsymbol{x}_\sigma, \Theta_\sigma}} U_{v_t+1} \underbrace{O_{H_{\boldsymbol{x}_\sigma, \Theta_\sigma}} |x_{\sigma_t}\rangle\langle x_{\sigma_t}| \cdots U_{v_{t-1}+1}}_{\text{stage (t)}}$$

$$\cdots$$

$$\cdot \underbrace{O_{H_{(x_{\sigma_1}, x_{\sigma_2}),(\Theta_{\sigma_1}, \Theta_{\sigma_2})}} |x_{\sigma_2}\rangle\langle x_{\sigma_2}| U_{v_2} \cdots O_{H_{x_{\sigma_1}, \Theta_{\sigma_1}}} U_{v_1+1}}_{\text{stage (2)}}$$

$$\cdot \underbrace{O_{H_{x_{\sigma_1}, \Theta_{\sigma_1}}} |x_{\sigma_1}\rangle\langle x_{\sigma_1}| U_{v_1} O_H \cdots O_H U_1 |0\rangle}_{\text{stage (1)}}$$

Here S_t^k denotes all ordered list of length t, with elements in $\{1, \ldots, k\}$ without duplication; $\boldsymbol{x}_\sigma$ denotes $(x_{\sigma_1}, \ldots, x_{\sigma_t})$ and $\boldsymbol{\Theta}_\sigma$ denotes $(\Theta_{\sigma_1}, \ldots, \Theta_{\sigma_t})$. We can similarly define $|\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle$ for all other $\boldsymbol{b} \in \{0,1\}^t$, the only difference here is the oracle may not be immediately reprogrammed at the end of each stage. More generally, for each $\boldsymbol{v}$ of length t and $\boldsymbol{b} \in \{0,1\}^t$, we define

$$|\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle = \sum_{\sigma \in S_t^k} |\phi_{\boldsymbol{v},\boldsymbol{b},\sigma}\rangle,$$

where $|\phi_{\boldsymbol{v},\boldsymbol{b},\sigma}\rangle$ is the state that is measured-and-reprogrammed according to $\boldsymbol{v}, \boldsymbol{b}$ with the order σ, similar to that in the definition of $|\phi_{\boldsymbol{v},\boldsymbol{0}}\rangle$. Thus, we have:

$$\left|\phi_q^{H_x,\Theta}\right\rangle = \sum_{\boldsymbol{v},\boldsymbol{b}} |\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle,$$

Adding the extra k queries. We assume the algorithm $\mathcal{A}$, after the first q queries, already prepares the output $\boldsymbol{x}, z$. We will force $\mathcal{A}$ making the last k queries, to generate $\boldsymbol{y} = H(\boldsymbol{x})$. Recall the definitions $\Pi_V^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}}$ and $G_{\boldsymbol{x}_\circ}$ in Definition 4. By setting $\boldsymbol{x} = \boldsymbol{x}_\circ$ and $\boldsymbol{\Theta} = \boldsymbol{y}_\circ$ in the above analysis, the probability on the RHS in the theorem we are proving is equal to:

$$\Pr_{H,G} \left[\mathcal{A}^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}} \to (\boldsymbol{x}, z) \text{ and } \boldsymbol{x} \equiv \boldsymbol{x}_\circ \text{ and } V^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}}(\boldsymbol{x}, H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}(\boldsymbol{x}), z) = 1 \right]$$

$$= \left\| G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}} \left|\phi_{q+k}^{H_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}}\right\rangle \right\|^2.$$

14

Since $G_{\boldsymbol{x}_\circ}$ and $\Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}}$ commute (as they are both projections over computational basis), we can assume $G_{\boldsymbol{x}_\circ}$ is applied to the state first. Even further, as the computation of $\boldsymbol{y} = H(\boldsymbol{x})$ and the projection $G_{\boldsymbol{x}_\circ}$ also commute, we can assume $G_{\boldsymbol{x}_\circ}$ applies to the state right before the last k queries, which are used to compute $\boldsymbol{y}$. Therefore, for every $|\phi_{\boldsymbol{v},\boldsymbol{b},\sigma}\rangle$, even if $t < k$ (the length of $\boldsymbol{v}$), we can measure-and-(immediately)-reprogram exactly $k - t$ locations of the last k queries, and making the random oracle exactly reprogrammed to $H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}$.

Thus, we have:

$$\left| \phi_{q+k}^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} \right\rangle = \sum_{\substack{\boldsymbol{v},\boldsymbol{b} \\ |\boldsymbol{v}|=k}} |\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle, \tag{1}$$

where the RHS has (at most) $2^k \binom{q+k}{k}$ terms.

By Equation (1), Cauchy-Schwartz and the triangle inequality, we have:

$$\left\| G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} \left| \phi_{q+k}^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} \right\rangle \right\|^2 \leq 2^k \binom{q+k}{k} \sum_{\substack{\boldsymbol{v},\boldsymbol{b} \\ |\boldsymbol{v}|=t}} \left\| G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} |\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle \right\|^2. \tag{2}$$

Finally, to prove the theorem statement, we relate each individual term on the RHS with the behaviors of our simulator B.

Relating each term with our simulator B. Next, we prove that each term $\left\| G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} |\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle \right\|^2$ is upper bounded by the probability that when the simulator B picks $\boldsymbol{v},\boldsymbol{b}$, it succeeds and outputs $\boldsymbol{x} \equiv \boldsymbol{x}_\circ$, which we denote by $p_{\boldsymbol{x}_\circ,\boldsymbol{v},\boldsymbol{b}}$.

Since the simulator B ensures that (1) no duplicated elements ever in the control register, (2) at the end, the control register only consists of inputs that are outputted by A (which will be $\boldsymbol{x}_\circ$, enforced by $G_{\boldsymbol{x}_\circ}$), we have that $p_{\boldsymbol{x}_\circ,\boldsymbol{v},\boldsymbol{b}}$ is the squared norm of the state $\left(G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} \otimes I_{\mathcal{R}} \right) |\psi_{\boldsymbol{v},\boldsymbol{b}}\rangle$, with the state $|\psi_{\boldsymbol{v},\boldsymbol{b}}\rangle$ being:

$$|\psi_{\boldsymbol{v},\boldsymbol{b}}\rangle = \sum_{\sigma \in S_k^k} |\phi_{\boldsymbol{v},\boldsymbol{b},\sigma}\rangle \otimes |\mathsf{set}\,\{(x_{\circ,\sigma_1}, y_{\circ,\sigma_1}), \ldots, (x_{\circ,\sigma_k}, y_{\circ,\sigma_k})\}\rangle_{\mathcal{R}}.$$

The only difference between $|\psi_{\boldsymbol{v},\boldsymbol{b}}\rangle$ and $|\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle$ is the extra control register! However, we realize that in this case, when σ is a permutation of $[k]$, the control register is unentangled, making $p_{\boldsymbol{x}_\circ,\boldsymbol{v},\boldsymbol{b}}$ is equal to $\left\| G_{\boldsymbol{x}_\circ} \Pi_V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}} |\phi_{\boldsymbol{v},\boldsymbol{b}}\rangle \right\|^2$. This is because the set will simply be $\mathsf{set}\{(x_{\circ,1}, y_{\circ,1}), \ldots, (x_{\circ,k}, y_{\circ,k})\}$, regardless of what σ is.

Finally, we have the L.H.S. is equal to

$$\Pr_{H,G}\left[\mathsf{Sim}^{H,G,A} \to (\boldsymbol{x}, \boldsymbol{y}, z) \text{ and } \boldsymbol{x} \equiv \boldsymbol{x}_\circ \text{ and } V^{H_{\boldsymbol{x}_\circ,\boldsymbol{y}_\circ}}(\boldsymbol{x}, \boldsymbol{y}, z) = 1 \right]$$

$$= \frac{1}{2^k \binom{q+k}{k}} \sum_{\substack{\boldsymbol{v},\boldsymbol{b} \\ |\boldsymbol{v}|=k}} p_{\boldsymbol{x}_\circ,\boldsymbol{v},\boldsymbol{b}}$$

Thus, combining Equation (2) and the equation above, we have:

$$\text{R.H.S.} = \left\| G_{\boldsymbol{x_\circ}} \Pi_V^{H_{\boldsymbol{x_\circ}, \boldsymbol{y_\circ}}} \left| \phi_q^{H_{\boldsymbol{x_\circ}, \boldsymbol{y_\circ}}} \right\rangle \right\|^2 \leq 2^k \binom{q+k}{k} \sum_{\substack{\boldsymbol{v}, \boldsymbol{b} \\ |\boldsymbol{v}|=k}} p_{\boldsymbol{x_\circ}, \boldsymbol{v}, \boldsymbol{b}}$$

$$\leq \left(2^k \binom{q+k}{k} \right)^2 \cdot \frac{1}{2^k \binom{q+k}{k}} \sum_{\substack{\boldsymbol{v}, \boldsymbol{b} \\ |\boldsymbol{v}|=k}} p_{\boldsymbol{x_\circ}, \boldsymbol{v}, \boldsymbol{b}}$$

$$= \text{L.H.S.}$$

where L.H.S. and R.H.S. denote the left/right-hand side term in the theorem statement. Therefore, we conclude the proof. $\square$

Lemma 6 (Coherent Measure-and-Reprogram results in Uniform Images). *Consider the Coherent Measure-and-Reprogram Experiment in Definition 6, but where we choose G to be uniformly random Then, for the measure-and-reprogram algorithm $\mathcal{B}$, the measurement $L = (\boldsymbol{x'}, \boldsymbol{\Theta'})$ of the $\mathcal{R}$ register (in Step 5) will result in uniformly random images $\boldsymbol{\Theta'}$.*

Proof. We will proceed with a proof by induction over the number of quantum queries of $\mathcal{A}$. In this proof, we will denote by n the total number of queries ($n = q + k$). For $n = 1$, let $v_j = n = 1$. Then, if $b_j = 0$, after updating the register $\mathcal{R}$ using the unitary U (in step 2), the register $\mathcal{R}$ will contain superposition of L sets consisting of a single pair $(x, G(x))$. Then, we perform the query to H controlled by $\mathcal{R}$ using O_H^{ctrl}, which is a query to the reprogrammed H on single points x, modifying accordingly the image register ($y \to y \oplus H_{x, G(x)}$), but which does not affect the $\mathcal{R}$ register. As G is random oracle, measuring $\mathcal{R}$ will result in a uniform image $\theta' = G(x)$, for some $x \in X$. If $b_j = 1$, we first query using O_H^{ctrl}, which is a query to the original H as L is empty. Then, we update $\mathcal{R}$ using unitary U, resulting in $\mathcal{R}$ containing superposition of L sets consisting of a single pair $(x, G(x))$. As before, measuring $\mathcal{R}$ will result in a uniform image $\theta' = G(x)$, for some $x \in X$. We emphasize that although Definition 6 defines G as an arbitrary function, in the statement of this Lemma, we consider uniformly random G instead of an arbitrary G.

For the inductive step, suppose that up to query $n-1$, the register $\mathcal{R}$ consists of sets L' with uniform images. Let $\mathcal{A}$ make its n-th query. If there does not exist any v_j equal to n then algorithm $\mathcal{B}$ answer $\mathcal{A}$'s query controlled by $\mathcal{R}$, reprogramming the oracle with the inputs and outputs pairs in L', but importantly $\mathcal{R}$ remains unchanged, hence $\mathcal{R}$ contains only uniform images by our inductive hypothesis. Otherwise, suppose there exists j^* such that $v_{j^*} = n$. Then if $b_{j^*} = 0$, we are first going to add in L the pair $(x, G(x))$ if x is not already in L, i.e. $L = L' \cup \{(x, G(x))\}$, otherwise $L = L'$. We are then going to make the controlled query O_H^{ctrl} to the reprogrammed oracle O_{H_L}, which does not affect the register $\mathcal{R}$. Hence by measuring $\mathcal{R}$ results in either $(x', \theta') \in L'$, which by hypothesis contains uniform image θ' or in $(x, G(x))$, which given that G is a

random oracle, also results in a uniform image. Similarly, if $b_{j*} = 1$ we are first going to make the controlled query O_H^{ctrl} to the reprogrammed oracle $O_{H'_L}$, then we will update the register $\mathcal{R}$ using the unitary U, which as before will either result in either $L = L' \cup \{(x, G(x))\}$ or $L = L'$. In both cases, by measuring $\mathcal{R}$ we will get a uniform image by using the uniformity of G and the inductive hypothesis. $\qquad \square$

4 Applications

4.1 Query Complexity

We will begin by first introducing the family of (security) games for which we will establish their quantum query complexity, namely the hardness of a quantum adversary to win such games.

Definition 7 (Multi-Output k-Search Game (Single-Instance)). *Let the random oracle $H : [M] \to [N]$, a distribution over challenges π_H and a winning relation $R_{H,ch}$ defined over Y^k.*
Then we define the multi-output k-search game $\mathcal{G}$ as follows:

1. *Challenger samples randomness ch and sends it to a quantum algorithm $\mathcal{A}$ having (quantum) oracle access to H;*
2. *Adversary $\mathcal{A}$ gets oracle access to H and outputs $\boldsymbol{x} := (x_1, ..., x_k), z$;*
3. *Challenger queries $\boldsymbol{x}$ to the random oracle, resulting in $\boldsymbol{y} := (y_1 = H(x_1), ..., y_k = H(x_k))$ and checks if they satisfy the winning relation:*
   $b := (x_1, \ldots, x_k, y_1, \ldots, y_k, z) \in R_{H,ch}$;
4. *If $b = 1$, $\mathcal{A}$ wins the $\mathcal{G}$ game.*

We will denote by $\epsilon_{\mathcal{G}}(q)$ the maximum probability over all q-quantum algorithms $\mathcal{A}$ of winning the multi-output k-search game $\mathcal{G}$.

Our main result is a quantum lifting theorem in the average case, relating the success probability of an arbitrary quantum algorithm to win a multi-output k-search game with the probability of success of a quantum algorithm equipped with exactly k quantum queries.

Theorem 7 (Lifting for Multi-Output k-Search Games). *Let $\mathcal{G}$ be a multi-output k-search game (as defined in Def. 7). Let $\mathcal{A}$ be a q-quantum query adversary in the game $\mathcal{G}$ (against the k-classical query challenger $\mathcal{C}$). Then there exists a k-quantum query adversary $\mathcal{B}$ against the game such that:*

$$\Pr[\mathcal{B}^{|H\rangle} \text{ wins } \mathcal{G}] \geq \frac{1}{2^{2k} \binom{q+k}{k}^2} \Pr[\mathcal{A}^{|H\rangle} \text{ wins } \mathcal{G}].$$

Proof. We will show that our Coherent Reprogramming result in Theorem 6 implies the lifting theorem. We will now show how to instantiate the coherent reprogramming theorem. Let $\boldsymbol{x}_\circ$ be uniformly sampled from X^k. Let $H', G' :$

17

$X \to Y$ be two uniform random oracles. Then, it is clear that, as $\boldsymbol{y}_\circ = G'(\boldsymbol{x}_\circ)$ is also uniform over Y^k, the reprogrammed function $H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ} : X \to Y$ is a uniform random function; this is due to the fact that, as stated in Theorem 6 (invoked here), the tuple $\boldsymbol{x}_\circ$ has distinct values for its element. We will instantiate the random oracle in the game $\mathcal{G}$ as the function $H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}$. Assume that in the game $\mathcal{G}$ after receiving the challenge and after performing its q quantum queries to $H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}$, the adversary $\mathcal{A}$ returns to the Challenger the outcome $\boldsymbol{x}$. Then, the Challenger queries $\boldsymbol{x}$ to $H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}$ resulting in $\boldsymbol{y}$ and checks if $\boldsymbol{y}$ satisfies the winning relation $R_{H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}, ch}$. Define $V^{H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}}$ as the predicate that outputs 1 if $\boldsymbol{y} \in R_{H'_{\boldsymbol{x}_\circ, \boldsymbol{y}_\circ}, ch}$ and 0 else. In this way, we observe that the probability that $\mathcal{A}$ wins the game $\mathcal{G}$ is exactly the RHS of Theorem 6. As a result, by Theorem 6, there must exist an efficient quantum simulator $\mathsf{Sim}^{H', G', \mathcal{A}}$ performing k quantum queries that also wins the game $\mathcal{G}$. Hence, it suffices to instantiate $\mathcal{B}$ as the simulator Sim.

$\square$

Let $L_{\mathcal{C}}$ represent the set of (classical) queries that a challenger performs during a multi-output k-search game $\mathcal{G}$ (Def. 7). For a quantum query adversary $\mathcal{B}$ against $\mathcal{G}$, we will denote by $L_{\mathcal{B}}$ the result of measuring its input and output query registers. Now, for the query complexity applications we will need the following stronger lifting theorem, which intuitively additionally guarantees the existence of an algorithm against $\mathcal{G}$ such that at the end of the game, measuring its input and output registers gives us exactly the set of queries of the challenger.

Theorem 8 (Lifting for Search Game with Uniform Images). *Let $\mathcal{G}$ be a multi-output k-search game (as defined in Def. 7). Let $\mathcal{A}$ be a q-quantum query adversary in the game $\mathcal{G}$ (against the k-classical query challenger $\mathcal{C}$). Then there exists a k-quantum query adversary $\mathcal{B}$ such that $L_{\mathcal{B}}$ is uniform, satisfying:*

$$\Pr[\mathcal{B}^{|H\rangle} \text{ wins } \mathcal{G} \text{ and } L_{\mathcal{C}} = L_{\mathcal{B}}] \geq \frac{1}{2^{2k} \binom{q+k}{k}^2} \Pr[\mathcal{A}^{|H\rangle} \text{ wins } \mathcal{G}].$$

Proof. The simulator algorithm $\mathcal{B}$ will follow the outline of the algorithm in the proof of Theorem 7, with the only difference that $\mathcal{B}$ will perform an additional step at the end. Namely, after interaction with Challenger $\mathcal{C}$, compute list of queries of $\mathcal{C}$ as $L_{\mathcal{C}}$. If any query in $L_{\mathcal{C}}$ has not yet been queried by $\mathcal{B}$, $\mathcal{B}$ will query them to oracle H. The uniformity of $L_{\mathcal{B}}$ follows directly from Lemma 6. $\square$

4.2 A New Quantum Lifting Theorem and Direct Product Theorem for Image Relations

Our first quantum lifting result (in Theorem 7) gives a first bound on the quantum hardness of solving any multi-output k-search game $\mathcal{G}$ by relating it to the probability of $\mathcal{G}$ being solved by a quantum algorithm with a small number of quantum queries. In this section we can derive a stronger quantum lifting theorem for the class of relations that only depend on images.

Theorem 9 (Quantum Lifting Theorem for Image Relations). *For any quantum algorithm $\mathcal{A}$ equipped with q quantum queries, $\mathcal{A}$'s success probability to solve the multi-output k-search game specified by the winning relation R, is bounded by:*

$\Pr[\mathcal{A}^{|H\rangle} \text{ wins multi-output } k\text{-search game}] \leq$

$$2^{2k}\binom{q+k}{k}^2 \Pr[\exists \text{ perm } \pi \mid (y_{\pi(1)}, y_{\pi(2)}, ..., y_{\pi(k)}) \in R : (y_1, ..., y_k) \xleftarrow{\$} Y^k].$$

For simplicity, in the rest of the section, we define $p(R)$ as:

$$p(R) = \Pr[\exists \text{ perm } \pi \mid (y_{\pi(1)}, y_{\pi(2)}, ..., y_{\pi(k)}) \in R : (y_1, ..., y_k) \xleftarrow{\$} Y^k].$$

Proof. Let $\mathcal{G}$ be a multi-output k-search game and assume a q-quantum adversary $\mathcal{A}$ sends to the Challenger the answer $\boldsymbol{x} = (x_1, ..., x_k)$. Challenger $\mathcal{C}$ will accept if and only if $\boldsymbol{y} := (H(x_1), ..., H(x_k)) \in R_{H,ch}$ and if x_i, x_j are pairwise distinct. By Theorem 8 we know there exists a quantum algorithm $\mathcal{B}$ making k quantum queries to H winning the game such that $L_\mathcal{B} = L_\mathcal{C}$ with success probability at least the success probability of $\mathcal{A}$ multiplied by a factor of $2^{2k}\binom{q+k}{k}^2$. The condition $L_\mathcal{B} = L_\mathcal{C}$ implies that $\mathcal{C}$ will verify as the images of $\mathcal{B}$'s answer exactly a permutation of the recorded information in $L_\mathcal{B}$. Therefore, due to the property of Theorem 8 that $L_\mathcal{B}$ will be uniformly over Y^k, $\mathcal{B}$'s winning probability will be lower bounded by the probability that there exists a permutation such that for uniformly sampled images from Y^k, the permuted images will belong to our target relation:

$$\Pr[\mathcal{A}^{|H\rangle} \text{ wins multi-output } k\text{-search game}] \leq 2^{2k}\binom{q+k}{k}^2 p(R).$$

$\square$

Next, we show a Direct Product Theorem for Image Relations.

Definition 8 (Direct Product). *Let $\mathcal{G}$ be a multi-output k-search game specified by the winning relation R, with respect to a random oracle $[M] \to [N]$. Define the following Direct Product $\mathcal{G}^{\otimes g}$:*

- *Let H be a random oracle $[g] \times [M] \to [N]$, and H_i denotes $H(i, \cdot)$;*
- *Challenger samples ch_i as in $\mathcal{G}$ for $i \in \{1, \ldots, g\}$.*
- *Adversary $\mathcal{A}$ gets oracle access to H and outputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_g, z_1, \ldots, z_g$ such that each input in $\boldsymbol{x}_i$ start with i.*
- *Challenger computes $b_i := (\boldsymbol{x}_i, H(\boldsymbol{x}_i), z_i) \in R_{H_i, ch_i}$;*
- *If all b_i equal to 1, $\mathcal{A}$ wins the $\mathcal{G}^{\otimes g}$ game.*

Theorem 10 (Direct Product Theorem for Image Relations). *For any quantum algorithm $\mathcal{A}$ equipped with gq quantum queries, $\mathcal{A}$'s success probability to solve the Direct Product $\mathcal{G}^{\otimes g}$ with the underlying $\mathcal{G}$ specified by the winning relation R, is bounded by*

$$\Pr[\mathcal{A}^{|H\rangle} \text{ wins } G^{\otimes g}] \leq \left(2^{2k}\binom{q+k}{k}^2 p(R)\right)^g.$$

Proof. Let $\mathcal{G}$ be a multi-output k-search game and assume a gq-quantum adversary $\mathcal{A}$ sends to the Challenger the answer $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_g, z_1, \ldots, z_g$. By Theorem 8 we know there exists a quantum algorithm $\mathcal{B}$ making gk quantum queries to H winning the game such that $L_\mathcal{B} = L_\mathcal{C}$ with success probability at least the success probability of $\mathcal{A}$ multiplied by a factor of $2^{2gk} \binom{gq+gk}{gk}^2$. The condition $L_\mathcal{B} = L_\mathcal{C}$ implies that $\mathcal{C}$ will verify as the images of $\mathcal{B}$'s answer exactly a permutation of the recorded information in $L_\mathcal{B}$. Moreover, for every image y, its associated input x only belongs to one of the oracles $H(i, \cdot)$; thus, it can only contribute to one of the relation checks R_{H_i, ch_i}. Thus, the permutation of the recorded information can only permute images with respect to the same oracle H_i.

Therefore, due to the property of Theorem 8 that $L_\mathcal{B}$ will be uniformly over Y^{gk}, $\mathcal{B}$'s winning probability will be lower bounded by the probability that there exists a permutation such that for uniformly sampled images from Y^{gk}, the permuted images will belong to our target relation:

$$\Pr[\mathcal{A}^{|H\rangle} \text{ wins } \mathcal{G}^{\otimes g}]$$

$$\leq 2^{2gk} \binom{gq + gk}{gk}^2 \Pr[\exists \pi_1, \ldots, \pi_g \in \mathcal{S}_k \mid (y_{i,\pi_i(1)}, ..., y_{i,\pi_i(k)}) \in R : (y_{i,1}, ..., y_{i,k}) \xleftarrow{\$} Y^k]$$

$$\leq \left(2^{2k} \binom{q + k}{k}^2 \Pr[\exists \pi \in \mathcal{S}_k \mid (y_{\pi(1)}, y_{\pi(2)}, ..., y_{\pi(k)}) \in R : (y_1, ..., y_k) \xleftarrow{\$} Y^k] \right)^g$$

$$\leq \left(2^{2k} \binom{q + k}{k}^2 p(R) \right)^g .$$

$\square$

In the following section, we will show some of the query complexity and cryptographic applications of our quantum lifting theorems and Direct Product Theorem.

4.2.1 Application 1: Non-uniform Security

Definition 9 (Advice Algorithms). *We define an advice (non-uniform) algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ equipped with q queries and advice of length S as follows:*

1. *$\mathcal{A}_1^H \to |adv\rangle$: an unbounded algorithm $\mathcal{A}_1$ outputs the advice $|adv\rangle$ consisting of S qubits;*
2. *$\mathcal{A}_2^H(|adv\rangle, ch) \to x$: q-quantum algorithm $\mathcal{A}_2$ takes as input the quantum advice $|adv\rangle$ and a challenge ch, outputs answer x;*

We define $\epsilon_\mathcal{G}^C(q, S)$ as the maximum winning probability over any advice adversary $\mathcal{A}$ equipped with q quantum queries and S classical bits of advice against the classically-verifiable search game $\mathcal{G}$.

We also consider multi-instance games, similar to Direct Product, except all the instances share the same oracle.

Definition 10 (Multi-Instance Game). *Let $\mathcal{G}$ be a multi-output k-search game specified by the winning relation R, with respect to a random oracle $[M] \to [N]$. Define the following Direct Product $\mathcal{G}_{\mathsf{MIS}}^{\otimes g}$:*

- *Let H be a random oracle $[M] \to [N]$;*
- *Challenger samples ch_i as in $\mathcal{G}$ for $i \in \{1, \dots, g\}$;*
- *Adversary $\mathcal{A}$ gets oracle access to H and outputs $\boldsymbol{x}_1, \dots, \boldsymbol{x}_g, z_1, \dots, z_g$;*
- *Challenger computes $b_i := (\boldsymbol{x}_i, H(\boldsymbol{x}_i), z_i) \in R_{H,ch_i}$;*
- *If all b_i equal to 1, $\mathcal{A}$ wins the $\mathcal{G}_{\mathsf{MIS}}^{\otimes g}$ game.*

From above, we can define $R_{\mathsf{MIS}}^{\otimes g}$ as the winning relation for $\mathcal{G}_{\mathsf{MIS}}^{\otimes g}$.

Lemma 7 (Multi-Output Implies Non-Uniform Classical Advice ([CGLQ20])).
Let $\mathcal{G}$ be a search game (as defined in Def. 7). If the maximum winning probability for any quantum algorithm equipped with q quantum queries against $\mathcal{G}_{\mathsf{MIS}}^{\otimes g}$ is $\epsilon_{\mathcal{G}_{\mathsf{MIS}}^{\otimes g}}(q)$, then the maximum probability of any non-uniform adversary equipped with q quantum queries and S-length classical advice against the original game $\mathcal{G}$ is at most:

$$\epsilon_{\mathcal{G}}^{C}(q, S) \leq 4 \cdot \left[\epsilon_{\mathcal{G}_{\mathsf{MIS}}^{\otimes S}}(Sq) \right]^{\frac{1}{S}}$$

By combining these results with our Quantum Lifting Theorem, we derive the security against advice (non-uniform) quantum algorithms.

Lemma 8 (Security against Advice Quantum Adversaries). *Let $\mathcal{G}$ be any multi-output k-search game specified by the winning relation R. Let $\mathcal{G}_{\mathsf{MIS}}^{\otimes g}$ be the multi-instance game and $R_{\mathsf{MIS}}^{\otimes g}$ be the relation. Any non-uniform algorithm $\mathcal{A}$ equipped with q quantum queries and S classical bits of advice can win the game $\mathcal{G}$ with probability at most:*

$$\epsilon_{\mathcal{G}}^{C}(q, S) \leq 4 \cdot 2^{2k} \binom{S(q+k)}{Sk}^{\frac{2}{S}} \cdot p(R_{\mathsf{MIS}}^{\otimes S}).$$

Proof. By combining our (strong) quantum lifting theorem (in Theorem 9) with the two advice results (Lemma 7). $\square$

4.2.2 Application 2: Salting Against Non-uniform Adversaries

Definition 11 (Salted Game). *Let $\mathcal{G}$ be a search game (as defined in Def. 7) specified by a random oracle $H : [M] \to [N]$, a distribution over challenges π_H and a winning relation $R_{H,ch}$ defined over Y. Then we define the salted version of $\mathcal{G}$ as the game $\mathcal{G}_s$ with salted space $[K]$ defined as follows:*

1. *The random oracle function is defined as: $G = (H_1, ..., H_K)$ for K random functions $H_i : [M] \to [N]$;*
2. *For any such G, the challenge $ch := (i, ch_i)$ is produced by first sampling uniformly at random $i \in [K]$ and then sampling ch_i according to π_{H_i};*

3. The winning relation is defined as $R_{G,ch} := R_{H_i,ch_i}$;

We will denote by $\epsilon_{\mathcal{G}_s}(q)$ the maximum probability over all q-quantum algorithms $\mathcal{A}$ of winning the salted game $\mathcal{G}_s$.

Lemma 9 (Security of Salted Game against Classical Advice). *Let $\mathcal{G}$ be a multi-output k-search game (as defined in Def. 7), specified by a relation R. Let $\mathcal{G}_s$ be the salted game, with salt space $[K]$. Then we have,*

$$\epsilon_{\mathcal{G}_s}^C(q, S) \leq 4 \cdot \frac{S}{K} + 4 \cdot 2^{2k} \binom{q+k}{k}^2 p(R).$$

Proof. By Lemma 7, the non-uniform security is related to the multi-instance game $\mathcal{G}_{s,\mathsf{MIS}}^{\otimes g}$, with salt space $[K]$. The security of the multi-instance game is closely related to the Direct Product, for salted games, as shown in [DLW24] (in the proof of Theorem 4.1). More precisely,

$$\epsilon_{\mathcal{G}_{s,\mathsf{MIS}}^{\otimes g}}(gq)^{1/g} \leq \epsilon_{\mathcal{G}_s^{\otimes g}}(gq)^{1/g} + \frac{g}{K}.$$

Intuitively, the only difference between the multi-instance game and the Direct Product is that, the same salt can be sampled with duplication. The extra factor $\frac{g}{K}$ captures the fact that the salt can be duplicated. Combining with Theorem 10, we have:

$$\begin{aligned}
\epsilon_{\mathcal{G}_s}^C(q, S) &\leq 4 \left(\epsilon_{\mathcal{G}_{s,\mathsf{MIS}}^{\otimes S}}(Sq) \right)^{1/S} \\
&\leq 4 \left(\epsilon_{\mathcal{G}_s^{\otimes S}}(Sq)^{1/S} + \frac{S}{K} \right) \\
&\leq 4 \cdot \frac{S}{K} + 4 \cdot 2^{2k} \binom{q+k}{k}^2 p(R).
\end{aligned}$$

$\square$

4.2.3 Application 3: Multi-Image Inversion

Our first result establishes the quantum hardness of multi-image inversion, which is a tight bound as already proven in [CGLQ20], but achieved here in a much simpler way, directly from our quantum lifting theorem.

Lemma 10 (Quantum Hardness of Multi-Image Inversion).
For any $\boldsymbol{y} = (y_1, ..., y_k) \in Y^k = [N]^k$ (without duplicates) and for any q-quantum query algorithm $\mathcal{A}$ whose task is to invert all the images in $\boldsymbol{y}$, the success probability of $\mathcal{A}$ is upper bounded by:

$$\Pr_H[\mathcal{A}^{|H\rangle}(\boldsymbol{y}) \to \boldsymbol{x} = (x_1, ..., x_k) \ : \ H(x_i) = y_i \ \forall i \in [k]]$$
$$\leq \left[\frac{4e(q+k)^2}{Nk} \right]^k$$

Proof. We will show this using our strong quantum lifting theorem for image relations (Theorem 9). Define R as the relation over $[N]^k$, with $H : [M] \to [N]$ such that: $R = \{y_1, ..., y_k\}$. Then for each permutation π, we have $\Pr[(y_{\pi(1)}, ..., y_{\pi(k)}) \in R \mid (y_1, ..., y_k) \leftarrow [N]^k] = \frac{1}{N^k}$. Using that the number of permutations π is $k!$ leads to:

$$\Pr_H[\mathcal{A}^{|H\rangle}(\boldsymbol{y}) \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_i) = y_i \; \forall i \in [k]] \leq 2^{2k} \binom{q+k}{k}^2 \cdot \frac{k!}{N^k}$$

Using first the inequality $\binom{q+k}{k} \leq \frac{(q+k)^k}{k!}$ and then the Stirling approximation $k! \geq \left(\frac{k}{e}\right)^k$, we get:

$$2^{2k} \binom{q+k}{k}^2 \cdot \frac{k!}{N^k} \leq \left(\frac{4}{N}\right)^k \cdot k! \cdot \left[\frac{(q+k)^k}{k!}\right]^2$$

$$\leq \left(\frac{4}{N}\right)^k \cdot (q+k)^{2k} \cdot \left(\frac{e}{k}\right)^k = \left[\frac{4e(q+k)^2}{Nk}\right]^k$$

$\square$

4.2.4   Application 4: Multi-Collision Finding and Multi-Search

Next, we can determine the quantum hardness of the multi-collision problem, namely finding k different inputs that map to the same output of the random oracle.

Lemma 11 (Quantum Hardness of Multi-Collision Finding and Salted Multi-Collision Finding).
For any q-quantum query algorithm $\mathcal{A}$, we have the upper bound for solving the k-multi-collision problem:

$$\Pr_H[\mathcal{A}^{|H\rangle}() \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_1) = ... = H(x_k)] \leq \frac{1}{N^{k-1}} \left[\frac{2e(q+k)}{k}\right]^{2k}$$

Any quantum algorithm $\mathcal{A}$ equipped with q quantum queries and S-bit of classical advice can win the salted multi-collision finding game with salted space $[K]$ with probability at most:

$$\Pr_H[\mathcal{A}() \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_1) = ... = H(x_k)] \leq \frac{4}{N^{k-1}} \left[\frac{2e(q+k)}{k}\right]^{2k} + \frac{4S}{K}.$$

Proof. We will show this using our strong quantum lifting theorem for image relations (Theorem 9). Define $R := \{y, ..., y\}_y$ the relation over $[N]^k$, where $H : [M] \to [N]$. Then for each permutation π, we have $\Pr[(y_{\pi(1)}, ..., y_{\pi(k)}) \in R \mid (y_1, ..., y_k) \leftarrow [N]^k] = \frac{1}{N^{k-1}}$. As R is permutation invariant, this implies that:

$$\Pr_H[\mathcal{A}^{|H\rangle}() \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_1) = ... = H(x_k)] \leq 2^{2k} \binom{q+k}{k}^2 \cdot \frac{1}{N^{k-1}}$$

23

Using first the inequality $\binom{q+k}{k} \leq \frac{(q+k)^k}{k!}$ and then the Stirling approximation:

$$2^{2k}\binom{q+k}{k}^2 \cdot \frac{1}{N^{k-1}} \leq N \cdot \left(\frac{4}{N}\right)^k \cdot \left[\frac{(q+k)^k}{k!}\right]^2$$

$$\leq N\left(\frac{4}{N}\right)^k \cdot (q+k)^{2k} \cdot \left(\frac{e}{k}\right)^{2k} = \frac{1}{N^{k-1}}\left[\frac{2e(q+k)}{k}\right]^{2k}$$

Finally, the security of salted multi-collision against non-uniform quantum adversaries equipped with S bits of advice follows by combining this quantum hardness bound of multi-collision with Lemma 9. $\square$

Finally, we consider another search application, namely the task of determining k different inputs that all map to 0 under the random oracle. One of the main motivations behind this problem is its relation to the notion of proof-of-work in the blockchain context [GKL15].

Lemma 12 (Quantum Hardness of Multi-Search). *For any q-quantum query algorithm $\mathcal{A}$ whose task is to find different preimages of 0 of a random oracle H, the success probability of $\mathcal{A}$ is upper bounded by:*

$$\Pr_H[\mathcal{A}^{|H\rangle}() \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_i) = 0 \; \forall i \in [k]] \leq \left[\frac{4e^2(q+k)^2}{Nk^2}\right]^k$$

Note that this bound is asymptotically tight, as an algorithm with q queries can use q/k queries to find each pre-image (Grover's algorithm), resulting in a probability of $\Theta\left(\left\{(\frac{q}{k})^2/N\right\}^k\right)$.

Proof. We will show this using our strong quantum lifting theorem for image relations (Theorem 9). Define $R := \{0, ..., 0\}$ the relation over $[N]^k$, where $H : [M] \to [N]$. Then for each permutation π, we have $\Pr[(y_{\pi(1)}, ..., y_{\pi(k)}) \in R \mid (y_1, ..., y_k) \leftarrow [N]^k] = \frac{1}{N^k}$. As R is permutation invariant, this implies that:

$$\Pr_H[\mathcal{A}^{|H\rangle}() \to \boldsymbol{x} = (x_1, ..., x_k) \; : \; H(x_i) = 0 \; \forall i \in [k]] \leq 2^{2k} \cdot \binom{q+k}{k}^2 \frac{1}{N^k}$$

Using first the inequality $\binom{q+k}{k} \leq \frac{(q+k)^k}{k!}$ and then the Stirling approximation:

$$2^{2k}\binom{q+k}{k}^2 \cdot \frac{1}{N^k} \leq \left(\frac{4}{N}\right)^k \cdot \left[\frac{(q+k)^k}{k!}\right]^2$$

$$\leq \left(\frac{4}{N}\right)^k \cdot (q+k)^{2k} \cdot \left(\frac{e}{k}\right)^{2k} = \left[\frac{4e^2(q+k)^2}{Nk^2}\right]^k$$

$\square$

Acknowledgements

References

ABKK23.   Amit Agarwal, James Bartusek, Dakshita Khurana, and Nishant Kumar. A new framework for quantum oblivious transfer. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 363–394, Cham, 2023. Springer Nature Switzerland.

AFK22.   Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography*, pages 113–142, Cham, 2022. Springer Nature Switzerland.

AFKR23.   Thomas Attema, Serge Fehr, Michael Klooß, and Nicolas Resch. The fiat–shamir transformation of $(\gamma_1, \ldots, \gamma_\mu)$-special-sound interactive proofs. Cryptology ePrint Archive, Paper 2023/1945, 2023. https://eprint.iacr.org/2023/1945.

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.

BDF+11.   Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

BKS21.   Nir Bitansky, Michael Kellner, and Omri Shmueli. Post-quantum resettably-sound zero knowledge. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 62–89, Cham, 2021. Springer International Publishing.

BR93.   Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

CGK+23.   Alexandru Cojocaru, Juan Garay, Aggelos Kiayias, Fang Song, and Petros Wallden. Quantum Multi-Solution Bernoulli Search with Applications to Bitcoin's Post-Quantum Security. *Quantum*, 7:944, March 2023.

CGLQ20.   Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. Tight quantum time-space tradeoffs for function inversion. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 673–684. IEEE, 2020.

Cha19.   André Chailloux. Tight quantum security of the fiat-shamir transform for commit-and-open identification schemes with applications to post-quantum signature schemes. Cryptology ePrint Archive, Paper 2019/699, 2019. https://eprint.iacr.org/2019/699.

DFHS23.   Jelle Don, Serge Fehr, Yu-Hsuan Huang, and Patrick Struck. On the (in)security of the buff transform. Cryptology ePrint Archive, Paper 2023/1634, 2023. https://eprint.iacr.org/2023/1634.

DFM20.   Jelle Don, Serge Fehr, and Christian Majenz. *The Measure-and-Reprogram Technique 2.0: Multi-round Fiat-Shamir and More*, page 602–631. Springer International Publishing, 2020.

DFMS19.   Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 356–383, Cham, 2019. Springer International Publishing.

DFMS22.   Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022*, pages 677–706, Cham, 2022. Springer International Publishing.

DLW24.   Fangqi Dong, Qipeng Liu, and Kewen Wu. Tight characterizations for pre-processing against cryptographic salting. In *Annual International Cryptology Conference*. Springer, 2024.

GHHM21.   Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the qrom. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 637–667, Cham, 2021. Springer International Publishing.

GKL15.   Juan A Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. *Journal of the ACM*, 2015.

GOP+23.   Chaya Ganesh, Claudio Orlandi, Mahak Pancholi, Akira Takahashi, and Daniel Tschudi. Fiat-shamir bulletproofs are non-malleable (in the random oracle model). Cryptology ePrint Archive, Paper 2023/147, 2023. https://eprint.iacr.org/2023/147.

JMZ23.   Haodong Jiang, Zhi Ma, and Zhenfeng Zhang. Post-quantum security of key encapsulation mechanism against cca attacks with a single decapsulation query. In Jian Guo and Ron Steinfeld, editors, *Advances in Cryptology – ASIACRYPT 2023*, pages 434–468, Singapore, 2023. Springer Nature Singapore.

Kat21.   Shuichi Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to qrom secure nizks. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021*, pages 580–610, Cham, 2021. Springer International Publishing.

KX24.   Haruhisa Kosuge and Keita Xagawa. Probabilistic hash-and-sign with retry in the quantum random oracle model. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography – PKC 2024*, pages 259–288, Cham, 2024. Springer Nature Switzerland.

LR13.   Troy Lee and Jérémie Roland. A strong direct product theorem for quantum query complexity. *computational complexity*, 22:429–462, 2013.

LZ19.   Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 326–355, Cham, 2019. Springer International Publishing.

She11.   Alexander A Sherstov. Strong direct product theorems for quantum communication and query complexity. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 41–50, 2011.

YZ21.   Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 568–597, Cham, 2021. Springer International Publishing.