On the Two-sided Permutation Inversion Problem

Gorjan Alagic[1], Chen Bai[2], Alexander Poremba[3], and Kaiyan Shi[4]

[1]QuICS, University of Maryland, and NIST
[2]Dept. of Electrical and Computer Engineering, University of Maryland
[3]Computing and Mathematical Sciences, California Institute of Technology
[4]Dept. of Computer Science, University of Maryland

Abstract

In the permutation inversion problem, the task is to find the preimage of some challenge value, given oracle access to the permutation. This is a fundamental problem in query complexity, and appears in many contexts, particularly cryptography. In this work, we examine the setting in which the oracle allows for quantum queries to both the forward and the inverse direction of the permutation—except that the challenge value cannot be submitted to the latter. Within that setting, we consider two options for the inversion algorithm: whether it can get quantum advice about the permutation, and whether it must produce the entire preimage (search) or only the first bit (decision). We prove several theorems connecting the hardness of the resulting variations of the inversion problem, and establish a number of lower bounds. Our results indicate that, perhaps surprisingly, the inversion problem does not become significantly easier when the adversary is granted oracle access to the inverse, provided it cannot query the challenge itself.

1 Introduction

1.1 The permutation inversion problem

The permutation inversion problem is defined as follows: given a permutation $\pi : [N] \to [N]$ and an image $y \in [N]$, output the correct preimage $x := \pi^{-1}(y)$. In the decision version of the problem, it is sufficient to output only the first bit of x. If the algorithm can only access π by making classical queries, then making $T = \Omega(N)$ queries is necessary and sufficient for both problems. If quantum queries are allowed, then Grover's algorithm can be used to solve both problems with $T = O(\sqrt{N})$ queries [Gro96, Amb02], which is worst-case asymptotically optimal [BBBV97, Amb02, Nay10].

In this work, we consider the permutation inversion problem in a setting where the algorithm is granted both forward and inverse quantum query access to the permutation π. In order to make the problem nontrivial, we modify the inverse oracle so that it outputs a reject symbol when queried on the challenge image y. We call this the *two-sided permutation inversion problem*. This variant appears naturally in the context of chosen-ciphertext security for encryption schemes based on (pseudorandom) permutations [KL20], as well as in the context of sponge hashing (SHA3) [GJMG11]. We consider several variants:

1. *(Auxiliary information.)* With this option enabled, the inversion algorithm now consists of two phases. The first phase is given a full description of π (e.g., as a table) and allowed to prepare an arbitrary quantum state ρ_π consisting of S qubits. This state is called *auxiliary*

information or *advice*. The second phase of the inversion algorithm is granted only the state ρ_π and query access to π, and asked to invert an image y. The two phases of the algorithm can also share an arbitrarily long uniformly random string, referred to as *shared randomness*. The complexity of the algorithm is measured in terms of the number of qubits S of the advice state (generated by the first phase) and the total number of queries T (made during the second phase.)

2. *(Adaptive restriction of challenge distribution.)* In this case, the inversion algorithm again consists of two phases. The first phase is again given a full description of π, and allowed to output a string $\mu \in \{0,1\}^m$ for $m < n$, where $n = \sqrt{N}$. The second phase is then granted query access to π and asked to invert an image y which is sampled uniformly at random from the set of all strings whose last m bits equal μ.

3. *(Search vs Decision.)* Here the two options simply determine whether the inversion algorithm is tasked with producing the entire preimage $x = \pi^{-1}(y)$ of the challenge y (search version), or only the first bit x_0 (decision version.)

If the algorithm is solving the search problem, we refer to it as a search permutation inverter, or SPI. If it is solving the decision problem, we refer to it as a decision permutation inverter, or DPI. If an SPI uses S qubits of advice and T queries to succeed with probability at least ϵ in the search inversion experiment, we say it is a (S, T, ϵ)-SPI. If a DPI uses S qubits of advice and T queries to succeed with probability at least $1/2 + \delta$ in the decision inversion experiment, we say it is a (S, T, δ)-DPI. If the algorithm is allowed to adaptively restrict the challenge distribution, we say it is adaptive and denote it by aSPI or aDPI, as appropriate.

In this work, we are mainly interested in the *average-case* setting. This means that both the permutation π and the challenge image y are selected uniformly at random. Moreover, the success probability is taken over all the randomness in the inversion experiment, i.e., over the selection of π and y along with all internal randomness and measurements of the inversion algorithm.

In Section 2, we present technical preliminaries, including the swapping lemma and quantum random access codes (QRAC), for subsequent proof. In Section 3, we introduce several definitions of the permutation inversion problem, with both auxiliary information and adaptive restriction of challenge distribution. Within Section 4, we show methods for amplifying the success probability of inversion in the non-adaptive case. Subsequently, in Section 5, we illustrate two reductions: from search-to-decision with auxiliary information and from unstructured search-to-decision without auxiliary information. These reductions are then utilized to derive lower bounds, as shown in Section 6. Finally, in Section 7, we propose a novel security notion, called one-way-QCCRA2, and establish the security of two common schemes under this notion, subject to specific conditions.

1.2 Related work

Previous works have considered the quantum-query *function* inversion problem [HXY19, CLQ19, CGLQ20, DKRS23, Liu23]. A number of papers gave lower bounds and time-space tradeoffs for the (one-sided) quantum-query permutation inversion problem, with and without advice [Amb02, Nay10, Ros21, NABT14, HXY19, CLQ19, FK15, BY23]. The relevant highlights among these are summarized in Table 1.

We remark that some of these previous works [CX21, CLQ19, NABT14] do not fully address the average-case setting. Specifically, they deal with inverters that are "restricted" in the following manner. First, the inverter is said to "invert y for π" if it succeeds in the inversion experiment for the specific pair (π, y) with probability at least $2/3$. Second, the inverter is said to "invert a δ-fraction of inputs" if $\Pr_{\pi,y}[\text{the inverter inverts } y \text{ for } \pi] \geq \delta$. This type of inverter is clearly captured by our notion above: it is an $(S, T, 2\delta/3)$-SPI. However, there are successful inverters of interest that are captured by our definition but are not restricted. For example, in a cryptographic context, one would definitely be concerned about adversaries that can invert every (π, y) with a probability of exactly $1/n$. Such an adversary is clearly a $(S, T, 1/n)$-SPI, but is not a restricted inverter for any value of δ. Other works also consider the general average-case (e.g., [CGLQ20, Liu23, HXY19]) but without two-way oracle access. Note that the lower bound for restricted adversaries described in [NABT14, CLQ19] can be translated to the more general lower bound in a black box way by applying our amplification procedure described in Lemma 4.2.

	[NABT14]	[CLQ19]	[HXY19]	Ours
Advice	classical	quantum	quantum	quantum
Access Type	one-sided	one-sided	one-sided	two-sided
Inverter	restricted	restricted	general	general
T-S trade-off	$ST^2 = \widetilde{\Omega}(N)$	$ST^2 = \widetilde{\Omega}(\epsilon N)$	$ST^2 = \widetilde{\Omega}(\epsilon^3 N)$	$ST^2 = \widetilde{\Omega}(\epsilon^3 N)$

Table 1: Summary of previous work on permutation inversion with advice. Success probability is denoted by ϵ. Note that $\epsilon = O(1)$ in [NABT14].

To our knowledge, the two-way variant of the inversion problem has only been considered in one other work. Specifically, [CX21] gives a lower bound of $T = \Omega(N^{1/5})$ to invert a random injective function (with two-way access and no advice) with a non-negligible success probability.

Another novelty of our work is that we give lower bounds and time-space tradeoffs for the decision problem (rather than just search). While prior work [CGLQ20] also considered the general decision game, their generic framework crucially relies on compressed oracles [Zha19] which are only known to support random *functions*. Consequently, their techniques cannot readily be applied in the context of permutation inversion due to a lack of "compressed permutation oracles".

We remark that the notion of two-way quantum accessibility to a random permutation has been considered in other works; for example, [ABKM22, ABK$^+$22] studied the hardness of detecting certain modifications to the permutation in this model. By contrast, we are concerned with the problem of finding the inverse of a random image.

2 Technical preliminaries

2.1 Swapping Lemma

Let $\mathcal{A}^f$ be a quantum algorithm with quantum oracle access to a function $f : \mathcal{X} \to \mathcal{Y}$, for some finite sets $\mathcal{X}$ and $\mathcal{Y}$. Let $\mathcal{S} \subseteq \mathcal{X}$ be a subset. Then, the total query magnitude of $\mathcal{A}^f$ on the set $\mathcal{S}$ is defined as $q(\mathcal{A}^f, \mathcal{S}) = \sum_{t=0}^{T-1} \|\Pi_{\mathcal{S}} |\psi_t\rangle \|^2$, where $|\psi_t\rangle$ represents the state of $\mathcal{A}$ just before the $(t+1)^{\text{st}}$ query and $\Pi_{\mathcal{S}}$ is the projector onto $\mathcal{S}$ acting on the query register of $\mathcal{A}$. We use the following simple fact: for any subset $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{A}$ making at most T queries, it holds that $q(\mathcal{A}^f, \mathcal{S}) \leq T$. The following lemma controls the ability of a query algorithm to distinguish two oracles, in terms of the total query magnitude to locations at which the oracles take differing values.

Lemma 2.1 (Swapping Lemma, [Vaz98])**.** *Let $f, g : \mathcal{X} \to \mathcal{Y}$ be functions with $f(x) = g(x)$ for all $x \notin \mathcal{S}$, where $\mathcal{S} \subseteq \mathcal{X}$. Let $|\Psi_f\rangle$ and $|\Psi_g\rangle$ denote the final states of a quantum algorithm $\mathcal{A}$ with quantum oracle access to the functions f and g, respectively. Then,*

$$\| \, |\Psi_f\rangle - |\Psi_g\rangle \, \| \leq \sqrt{T \cdot q(\mathcal{A}^f, \mathcal{S})},$$

where $\| \, |\Psi_f\rangle - |\Psi_g\rangle \, \|$ denotes the Euclidean distance and where T is an upper bound on the number of quantum oracle queries made by $\mathcal{A}$.

2.2 Lower bounds for quantum random access codes

Quantum random access codes [Wie83, ANTV99, ALMO08] are a means of encoding classical bits into (potentially fewer) qubits. We use the following variant from [CLQ19].

Definition 2.2 (Quantum random access codes with variable length)**.** *Let N be an integer and let $\mathcal{F}_N = \{f : [N] \to \mathcal{X}_N\}$ be an ensemble of functions over some finite set $\mathcal{X}_N$. A quantum random access code with variable length* (QRAC-VL) *for $\mathcal{F}_N$ is a pair* (Enc, Dec) *consisting of a quantum encoding algorithm* Enc *and a quantum decoding algorithm* Dec*:*

- $\mathsf{Enc}(f; R)$*: The encoding algorithm takes as input a function $f \in \mathcal{F}_N$ together with a set of random coins $R \in \{0,1\}^*$, and outputs a quantum state ρ on $\ell = \ell(f)$ many qubits (where ℓ may depend on f).*
- $\mathsf{Dec}(\rho, x; R)$*: The decoding algorithm takes as input a state ρ, an element $x \in [N]$ and random coins $R \in \{0,1\}^*$ (same randomness used for the encoding), and seeks to output $f(x)$.*

The performance of a QRAC-VL *is characterized by parameters L and δ. Let $L := \mathop{\mathbb{E}}_f[\ell(f)]$ be the average length of the encoding over the uniform distribution on $f \in \mathcal{F}_N$, and let*

$$\delta = \Pr_{f,x,R} [\mathsf{Dec}(\mathsf{Enc}(f; R), x; R) = f(x)]$$

be the probability that the scheme correctly reconstructs the image of the function, where $f \in \mathcal{F}_N$, $x \in [N]$ and R are all chosen uniformly at random.

We use the following information-theoretic lower bound on the expected length of any QRAC-VL scheme for permutations, which is a consequence of [CLQ19, Theorem 5].

Theorem 2.3 ([CLQ19], Corollary 1)**.** *For any* QRAC-VL *for the set of permutations $\mathcal{S}_N$ (of the set $[N]$) with $\delta = 1 - k/N$ for some $k = \Omega(1/N)$, we have*

$$L \geq \log N! - O(k \log N)\,.$$

3 The permutation inversion problem

We begin by formalizing the search version of the permutation inversion problem. We let $[N] = \{1, ..., N\}$; typically we choose $N = 2^n$ for some positive integer n. For $f : \mathcal{X} \to \mathcal{Y}$ a function from a set $\mathcal{X}$ to an additive group $\mathcal{Y}$ (typically just bitstrings), the quantum oracle $\mathcal{O}_f$ is the unitary operator $\mathcal{O}_f : |x\rangle |y\rangle \to |x\rangle |y \oplus f(x)\rangle$. We use $\mathcal{A}^{\mathcal{O}_f}$ (or sometimes simply $\mathcal{A}^f$) to denote that algorithm $\mathcal{A}$ has quantum oracle access to f.

Definition 3.1. *Let $m, n \in \mathbb{N}$ and $M = 2^m,\ N = 2^n$. An adaptive search-version permutation inverter (aSPI) is a pair $\mathsf{aS} = (\mathsf{aS}_0, \mathsf{aS}_1)$ of quantum algorithms, where*

- aS_0 *is an algorithm that receives as input a truth table for a permutation over $[N]$ and a random string r, and outputs a quantum state as well as a classical string $\mu \in \{0,1\}^m$ with $0 \le m < n$;*
- aS_1 *is an oracle algorithm that receives a quantum state, a classical string $\mu \in \{0,1\}^m$, an image $y \in [N]$, and a random string r, and outputs $x \in \{0,1\}^{n-m}$.*

Note that m is a parameter of the adaptivity, i.e. the length of the adaptive string.

We will consider the execution of an aSPI aS in the following experiment,

1. *(sample coins)* a uniformly random permutation $\pi : [N] \to [N]$ and a uniformly random string $r \leftarrow \{0,1\}^*$ are sampled;
2. *(prepare advice)* aS_0 is run, producing a pair consisting of a quantum state and a string $(\rho_{\pi,r,\mu}, \mu) \leftarrow \mathsf{aS}_0(\pi, r)$;
3. *(sample instance)* a random image $y \in [N]$ is generated by first sampling a random string $x \leftarrow \{0,1\}^{n-m}$ and then letting $y = \pi(x\|\mu)$;
4. *(invert)* aS_1 is run with the oracles below, and produces a candidate preimage x^*.

$$\mathcal{O}_\pi : |w\rangle\,|z\rangle \to |w\rangle\,|z \oplus \pi(w)\rangle \qquad \mathcal{O}_{\pi^{-1}_{\perp y}} : |w\rangle\,|z\rangle \to |w\rangle\,|z \oplus \pi^{-1}_{\perp y}(w)\rangle\,, \tag{1}$$

   where $\pi^{-1}_{\perp y} : [N] \times \{0,1\} \to [N] \times \{0,1\}$ is defined by

$$\pi^{-1}_{\perp y}(w\|b) = \begin{cases} \pi^{-1}(w)\|0 & \text{if } b = 0 \text{ and } w \neq y \\ 1^{\lceil \log N \rceil}\|1 & \text{otherwise.} \end{cases}$$

   To keep the notation simple, we write this entire process as $x^* \leftarrow \mathsf{aS}_1^{\pi_{\perp y}}(\rho_{\pi,r,\mu}, \mu, y, r)$. We will use $\pi_{\perp y}$ to denote simultaneous access to the two oracles in (1) throughout the paper.
5. *(check)* If $\pi(x^*\|\mu) = y$, output 1; otherwise output 0.

Note that the two oracles allow for the evaluation of the permutation π in both the forward and inverse directions. To disallow trivial solutions, the oracle outputs a fixed "reject" element $1^{\lceil \log N \rceil}\|1 \in [N] \times \{0,1\}$ if queried on y in the inverse direction.

Definition 3.2. *An (S, T, ϵ)-aSPI is a search-version adaptive permutation inverter $\mathsf{aS} = (\mathsf{aS}_0, \mathsf{aS}_1)$ satisfying all of the following:*

1. $\Pr\left[\pi^{-1}(y) \leftarrow \mathsf{aS}_1^{\pi_{\perp y}}(\rho, \mu, y, r)\ :\ (\rho, \mu) \leftarrow \mathsf{aS}_0(\pi, r),\ y = \pi(x\|\mu)\right] \ge \epsilon$, *where the probability is taken over $\pi \leftarrow \mathcal{S}_N$, $r \leftarrow \{0,1\}^*$ and $x \leftarrow \{0,1\}^{n-m}$, along with all internal randomness and measurements of* aS*;*
2. *$S = S(\mathsf{aS})$ is an upper bound on the number of qubits of ρ in the above.*

3. $T = T(\mathsf{aS})$ *is an upper bound on the number of oracle queries made by* aS_1.

We emphasize that the running time of aS and the length of the shared randomness r are only required to be finite. We will assume that both S and T depend only on the parameter N; in particular, they will not vary with π, y, r, or any measurements.

Definition 3.3. *A search-version permutation inverter (SPI)* $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ *is defined as an aSPI with* $m = 0$. *An* (S, T, ϵ)*-SPI is an* (S, T, ϵ)*-aSPI with* $m = 0$.

Decision version. The decision version of the permutation inversion problem is defined similarly to the search version above. An adaptive decision-version permutation inverter (aDPI) is denoted $\mathsf{aD} = (\mathsf{aD}_0, \mathsf{aD}_1)$, and outputs one bit b rather than a full candidate preimage. In the "check" phase of the experiment, the single-bit output b of aD_1 is compared to the first bit $\pi^{-1}(y)|_0$ of the preimage of the challenge y. Success probability is now measured in terms of the advantage over the random guessing probability of 1/2.

Definition 3.4. *A* (S, T, δ)*-aDPI is a decision-version adaptive permutation inverter* $\mathsf{aD} = (\mathsf{aD}_0, \mathsf{aD}_1)$ *satisfying all of the following:*

1. $\Pr\left[\pi^{-1}(y)|_0 \leftarrow \mathsf{aD}_1^{\pi_{\perp y}}(\rho, \mu, y, r) \;:\; (\rho, \mu) \leftarrow \mathsf{aD}_0(\pi, r),\; y = \pi(x\|\mu)\right] \geq \frac{1}{2} + \delta$, *where the probability is taken over* $\pi \leftarrow \mathcal{S}_N$, $r \leftarrow \{0,1\}^*$ *and* $x \leftarrow \{0,1\}^{n-m}$, *along with all internal randomness and measurements of* aD. *Here* $\pi^{-1}(y)|_0$ *denotes the first bit of* $\pi^{-1}(y)$

2. $S = S(\mathsf{aS})$ *is an upper bound on the number of qubits of* ρ *in the above.*

3. $T = T(\mathsf{aS})$ *is an upper bound on the number of oracle queries made by* aS_1.

Definition 3.5. *A decision-version permutation inverter (DPI)* $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$ *is defined as an aDPI with* $m = 0$. *An* (S, T, δ)*-DPI is an* (S, T, δ)*-aDPI with* $m = 0$.

4 Amplification

In this section, we show how to amplify the success probability of search and decision inverters, in the non-adaptive (i.e., $m = 0$) case. The construction for the search case is shown in Protocol 1.

Protocol 1 (ℓ-time repetition of (S, T, ϵ)-SPI)**.** *Given an* (S, T, ϵ)*-SPI* $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ *and an integer* $\ell > 0$, *define a SPI* $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ *as follows.*

1. *(Advice Preparation)* $\mathsf{S}[\ell]_0$ *proceeds as follows:*
   (a) *receives as input a random permutation* $\pi : [N] \to [N]$ *and randomness* $r \leftarrow \{0,1\}^*$ *and parses the string* r *into* 2ℓ *substrings* $r = r_0\|...\|r_{\ell-1}\|r_\ell\|...\|r_{2\ell-1}$ *(with lengths as needed for the next step).*
   (b) *uses* $r_0, ..., r_{\ell-1}$ *to generate* ℓ *permutation pairs* $\{\sigma_{1,i}, \sigma_{2,i}\}_{i=0}^{\ell-1}$ *in* $\mathcal{S}_N$, *and then runs* $\mathsf{S}_0(\sigma_{1,i} \circ \pi \circ \sigma_{2,i}, r_{i+\ell})$ *to get a quantum state* $\rho_i := \rho_{\sigma_{1,i}\circ\pi\circ\sigma_{2,i}, r_{i+\ell}}$ *for all* $i \in [0, \ell-1]$. *Finally,* $\mathsf{S}[\ell]_0$ *outputs the quantum state* $\bigotimes_{i=0}^{\ell-1} \rho_i$.

2. *(Oracle Algorithm)* $\mathsf{S}[\ell]_1^{\pi_{\perp y}}$ *proceeds as follows:*

(a) *receives* $\bigotimes_{i=0}^{\ell-1} \rho_i$*, randomness* r *and an image* $y \in [N]$ *as input.*

(b) *parses* $r = r_0\|...\|r_{\ell-1}\|r_\ell\|...\|r_{2\ell-1}$ *and uses the coins* $r_0\|...\|r_{\ell-1}$ *to reconstruct the permutations* $\{\sigma_{1,i}, \sigma_{2,i}\}_{i=0}^{\ell-1}$ *in* $\mathcal{S}_N$*.*

(c) *runs the following routine for all* $i \in [0, \ell-1]$*:*

   i. *run* S_1 *with oracle access to* $(\sigma_{1,i} \circ \pi \circ \sigma_{2,i})_{\perp \sigma_{1,i}(y)}$*, which implements the permutation* $\sigma_{1,i} \circ \pi \circ \sigma_{2,i}$ *and its inverse (with output* $\perp$ *on input* $\sigma_{1,i}(y)$*).* [a]

   ii. *get back* $x_i \leftarrow \mathsf{S}_1^{(\sigma_{1,i} \circ \pi \circ \sigma_{2,i})_{\perp \sigma_{1,i}(y)}}(\rho_i, \sigma_{1,i}(y), r_{i+\ell})$*.*

(d) *queries the oracle* $\pi_{\perp y}$ *(in the forward direction) on each* $\sigma_{2,i}(x_i)$ *to see if* $\pi(\sigma_{2,i}(x_i)) = y$*. If such an* $\sigma_{2,i}(x_i)$ *is found, outputs it; otherwise outputs* 0*.*

[a] How to construct this quantum oracle is described in Appendix B.1.

In the adaptive case, a difficulty arises with the above approach. To amplify the probability, we randomize the permutation in each iteration and $\mathsf{aS}[\ell]_0$ produces corresponding advice for each randomized permutation. In the adaptive case, $\mathsf{aS}[\ell]_0$ needs to output an adaptive string μ which is used to produce the image y. However, running aS_0 for each randomized permutation will, in general, result in a different μ in each execution, and it is unclear how one can use these to generate a single μ' in the amplified algorithm. We remark that other works considered different approaches to amplification, e.g., via quantum rewinding [HXY19] and the gentle measurement lemma [CGLQ20].

Lemma 4.1 (Amplification, search)**.** *Let* S *be a* (S, T, ϵ)*-SPI for some* $\epsilon > 0$*. Then* $\mathsf{S}[\ell]$ *is a* $(\ell S, \ell(T+1), 1-(1-\epsilon)^\ell)$*-SPI.*

Proof. We consider the execution of the ℓ-time repetition of (S, T, ϵ)-SPI, denoted by SPI $\mathsf{S}[\ell]$, in the search permutation inversion experiment defined in Protocol 1. By construction, $\mathsf{S}[\ell]$ runs ℓ-many SPI procedures $(\mathsf{S}_0, \mathsf{S}_1)$. Since S is assumed to be an (S, T, ϵ)-SPI, let $\pi_i = \sigma_{1,i} \circ \pi \circ \sigma_{2,i}$, for each iteration $i \in [0, \ell-1]$ it follows that

$$
\begin{aligned}
&\Pr\left[(\pi_i)^{-1}(\sigma_{1,i}(y)) \leftarrow \mathsf{S}_1^{(\pi_i)_{\perp \sigma_{1,i}(y)}}\left(\rho_i, \sigma_{1,i}(y), r_{i+\ell}\right) : \rho_i \leftarrow \mathsf{S}_0(\pi_i, r_{i+\ell})\right] \\
&\equiv \Pr\left[(\sigma_{2,i})^{-1} \circ \pi^{-1}(y) \leftarrow \mathsf{S}_1^{(\pi \circ \sigma_{2,i})_{\perp y}}\left(\rho_{\pi \circ \sigma_{2,i}, r_{i+\ell}}, y, r_{i+\ell}\right) : \rho_{\pi \circ \sigma_{2,i}, r_{i+\ell}} \leftarrow \mathsf{S}_0(\pi \circ \sigma_{2,i}, r_{r+\ell})\right] \\
&\equiv \Pr\left[\pi^{-1}(y) \leftarrow \mathsf{S}_1^{\pi_{\perp y}}\left(\rho_{\pi, r_{i+\ell}}, y, r_{i+\ell}\right) : \rho_{\pi, r_{i+\ell}} \leftarrow \mathsf{S}_0(\pi, r_{r+\ell})\right] \ \geq \ \epsilon,
\end{aligned}
$$

where the probability is taken over $\pi \leftarrow \mathcal{S}_N$ and $r \leftarrow \{0,1\}^*$ (which is used to sample permutations σ_i), along with all internal measurements of S.

Essentially, for all $i \in [0, \ell-1]$, the goal of the i-th trial is to find the preimage x_i such that $\sigma_{2,i}(x_i) = \pi^{-1}(y)$. Since all $\{\sigma_{2,i}\}$ are independently randomly generated, the elements $\sigma_{2,i}(x_i)$ are independent for each i in the range $[0, \ell-1]$. Therefore, all ℓ trails are mutually independent. Therefore, we get that

$$
\begin{aligned}
&\Pr\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \\
&= 1 - \Pr\left[\bigcap_{i=0}^{\ell-1}\left[(\pi \circ \sigma_{2,i})^{-1}(y) \not\leftarrow \mathsf{S}_1^{(\pi_i)_{\perp \sigma_{1,i}(y)}}\left(\rho_i, \sigma_{1,i}(y), r_{i+\ell}\right) : \rho_i \leftarrow \mathsf{S}_0(\pi_i, r_{i+\ell})\right]\right] \\
&= 1 - \prod_{i=0}^{\ell-1} \Pr\left[(\pi \circ \sigma_{2,i})^{-1}(y) \not\leftarrow \mathsf{S}_1^{(\pi_i)_{\perp \sigma_{1,i}(y)}}\left(\rho_i, \sigma_{1,i}(y), r_{i+\ell}\right) : \rho_i \leftarrow \mathsf{S}_0(\pi_i, r_{i+\ell})\right] \\
&\geq 1 - (1-\epsilon)^{\ell}.
\end{aligned}
$$

Given that the SPI $(\mathsf{S}_0, \mathsf{S}_1)$ requires space S and T queries, we have that $(\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ requires space $S(\mathsf{S}[\ell]) = \ell \cdot S$ and query number $T(\mathsf{S}[\ell]) = \ell \cdot (T+1)$, as both algorithms need to run either S_0 or S_1 ℓ-many times as subroutines. This proves the claim. □

We also need a variant of the above to compute the search lower bound.

Lemma 4.2. *Let* S *be a* (S, T, ϵ)*-SPI for some* $\epsilon > 0$*. Then, we can construct an SPI* $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ *using* $S(\mathsf{S}[\ell])$ *qubits of advice and making* $T(\mathsf{S}[\ell])$ *queries, with*

$$
S(\mathsf{S}[\ell]) = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil \cdot S \quad \text{and} \quad T(\mathsf{S}[\ell]) = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil \cdot (T+1)
$$

such that

$$
\Pr_{\pi, y}\left[\Pr_r\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \frac{1}{5}.
$$

The proof is analogous to Lemma 4.1 and is given in Appendix B.2.

We also consider amplification for the decision version; the construction is essentially the same, except that the final "check" step is replaced by outputting the majority bit.

Lemma 4.3 (Amplification, decision)**.** *Let* D *be a* (S, T, δ)*-DPI for some* $\delta > 0$*. Then* $\mathsf{D}[\ell]$ *is a* $(\ell S, \ell T, 1/2 - \exp\left(-\delta^2/(1+2\delta) \cdot \ell\right))$*-DPI.*

The proof is analogous to the search version and given in Appendix B.3.

5 Reductions

We give two reductions related to the inversion problem: a search-to-decision reduction (for the case of advice), and a reduction from unstructured search to the decision inversion problem (for the case of no advice).

5.1 A search-to-decision reduction

To construct a search inverter from a decision inverter, we take the following approach. We first amplify the decision inverter so that it correctly computes the first bit of the preimage with certainty. We then repeat this amplified inverter n times (once for each bit position) but randomize the instance in such a way that the j-th bit of the preimage is permuted to the first position. We then output the string of resulting bits as the candidate preimage.

Theorem 5.1. *Let* D *be a* (S, T, δ)*-DPI. Then for any* $\ell \in \mathbb{N}$*, we can construct a* $(n\ell S, n\ell T, \eta)$*-SPI with*

$$\eta \geq 1 - n \cdot \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right), \qquad \textit{where } n = \lceil \log N \rceil.$$

Proof. Given an δ-DPI $(\mathsf{D}_0, \mathsf{D}_1)$ with storage size S and query size T, we can construct a η'-DPI $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ with storage size ℓS and query size ℓT through ℓ-time repetition. By Lemma 4.3, we have that $\eta' \geq \frac{1}{2} - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right)$. Note that the algorithm $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ runs $(\mathsf{D}_0, \mathsf{D}_1)$ as a subroutine. In the following, we represent elements in $[N]$ using a binary decomposition of length $\lceil \log N \rceil$. To state our search-to-decision reduction, we introduce a generalized swap operation, denoted by $\mathsf{swap}_{a,b}$, which acts as follows for any quantum state of m qubits:

$$\mathsf{swap}_{a,b} |w\rangle = \mathsf{swap}_{a,b} |w_{m-1} \dots w_b \dots w_a \dots w_1 w_0\rangle = |w_{m-1} \dots w_a \dots w_b \dots w_1 w_0\rangle$$

Note that $\mathsf{swap}_{k,k}$ is equal to the identity, i.e. $\mathsf{swap}_{k,k} |x\rangle = |x\rangle$ for $x \in [N]$ and $k \in [0, \lceil \log N \rceil - 1]$. We construct a SPI $(\mathsf{S}_0, \mathsf{S}_1)$ as follows.

1. The algorithm S_0 proceeds as follows:
   (a) S_0 receives a random permutation $\pi : [N] \to [N]$ and a random string $r \leftarrow \{0,1\}^*$ as inputs. We parse r into $\lceil \log N \rceil$ individual substrings, i.e. $r = r_0 \| \dots \| r_{\lceil \log N \rceil - 1}$; the length of each substring is clear in context.
   (b) S_0 runs the algorithm $\mathsf{D}[\ell]_0(\pi \circ \mathsf{swap}_{0,j}, r_j)$ to obtain quantum advice $\rho_{\pi \circ \mathsf{swap}_{0,j}, r_j}$ for each $j \in [0, \lceil \log N \rceil - 1]$. Finally, S_0 outputs a quantum state $\rho = \bigotimes_{j=0}^{\lceil \log N \rceil - 1} \rho_{\pi \circ \mathsf{swap}_{0,j}, r_j}$. (Note: We let $\rho_j = \rho_{\pi \circ \mathsf{swap}_{0,j}, r_j}$ for the rest of the proof.)
2. The oracle algorithm $\mathsf{S}_1^{\mathcal{O}_\pi, \mathcal{O}_{\pi^{-1}_{\perp y}}}$ proceeds as follows:[1]
   (a) S_1 receives $\bigotimes_{j=0}^{n-1} \rho_j$, a random string $r := r_0 \| \dots \| r_{n-1}$ and an image $y \in [N]$.
   (b) S_1 then runs the following routine for each $j \in [0, \lceil \log N \rceil - 1]$:
      i. Run $\mathsf{D}[\ell]_1$ with oracle access to $\mathcal{O}_{\pi \circ \mathsf{swap}_{0,j}}$ and $\mathcal{O}_{(\pi \circ \mathsf{swap}_{0,j})^{-1}_{\perp y}}$, where

$$\mathcal{O}_{\pi \circ \mathsf{swap}_{0,j}}(|w\rangle_1 |z\rangle_2) = \left(\mathsf{swap}_{0,j} \otimes I\right) \mathcal{O}_\pi \left(\mathsf{swap}_{0,j} \otimes I\right) |w\rangle_1 |z\rangle_2$$
$$\mathcal{O}_{(\pi \circ \mathsf{swap}_{0,j})^{-1}_{\perp y}}(|w\rangle_1 |z\rangle_2) = (I \otimes \mathsf{swap}_{0,j}) \mathcal{O}_{\pi^{-1}_{\perp y}} |w\rangle_1 |z\rangle_2$$

      ii. Let $b_j \leftarrow \mathsf{D}[\ell]_1^{(\pi \circ \mathsf{swap}_{0,j})_{\perp y}}(\rho_j, y, r_j)$ denote the output.
   (c) S_1 outputs $x^* \in [N]$ with the binary decomposition $x^* = \sum_{j=0}^{\lceil \log N \rceil - 1} 2^j \cdot b_j$.

We now argue that the probability that $\mathcal{D}[\ell]_1$ correctly recovers the pre-image bits b_i and b_j is independent for each $i \neq j$. From Lemma 4.3, we know that $\mathsf{D}[\ell]_1$ runs D_1 as a subroutine, i.e. it decides the first bit of the pre-image of y by running D_1 (in Lemma 4.3) ℓ times with different random coins. It actually needs to recall D_1 for amplification and for each iteration in this amplification $k \in [0, \ell - 1]$, where the actual modified permutation under use is $\sigma_{i,k} \circ \pi \circ \mathsf{swap}_{0,i}$ and image is

[1] Here, we borrow the notation for $\mathcal{O}_\pi$ and $\mathcal{O}_{\pi^{-1}_{\perp y}}$ from the experiment described in Section 3.

$\sigma_{i,k}(y)$. Similarly for term j, $\sigma_{j,k} \circ \pi \circ \mathsf{swap}_{0,j}$ and $\sigma_{j,k}(y)$ is used as the permutation and image. Since the random coins (r_i and r_j), which are used to modify the target permutation π, are independently random, those random permutations ($\sigma_{i,k}$ and $\sigma_{j,k}$) generated from random coins are independently random and so do those modified composition permutations, images, and advice states.

Analyzing the success probability of $(\mathsf{S}_0, \mathsf{S}_1)$, we find that

$$\begin{aligned}
&\Pr\left[\pi^{-1}(y) \leftarrow \mathsf{S}_1^{\pi_{\perp y}}(\rho, y, r) \ : \ \rho \leftarrow \mathsf{S}_0(\pi, r)\right] \\
&= \Pr\left[\bigwedge_{j=0}^{\lceil \log N \rceil - 1} \pi^{-1}(y)|_j \leftarrow \mathsf{D}[\ell]_1^{(\pi \circ \mathsf{swap}_{0,j})_{\perp y}}(\rho_j, y, r_j)\right] \\
&\geq \left(1 - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right)\right)^{\lceil \log N \rceil} \geq 1 - \lceil \log N \rceil \cdot \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right).
\end{aligned}$$

where the last line follows from Bernoulli's inequality. Finally, we compute the resources needed for $(\mathsf{S}_0, \mathsf{S}_1)$. By Lemma 4.3, $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ requires space ℓS and query size ℓT. For $j \in [0, \lceil \log N \rceil - 1]$, S_0 stores $\mathsf{D}[\ell]_0$'s outputs and thus S requires storage size $\lceil \log N \rceil \ell S$. Similarly, S_1 runs $\mathsf{D}[\ell]_1$ to obtain b_j and thus it requires $\lceil \log N \rceil \ell T$ queries in total. $\square$

Comparison to O2H lemma. The one-way to hiding (O2H) lemma [AHU19] also presents a natural reduction from search to decision in the context of general quantum oracle algorithms. However, it is quite limited in our setting. For example, given a decision inverter capable of computing the first bit of $\pi^{-1}(y)$ with certainty after q queries, the O2H lemma yields a search inverter that can invert y with success probability $\frac{1}{4q^2}$ after $\approx q$ queries. By comparison, our amplification technique achieves an inversion of y with a success probability of 1 with nq queries, which is significantly better in the relevant setting of $q \gg n$. However, in applications where only one copy of the advice is available for the amplified algorithm, O2H still works while our amplification technique fails.

5.2 A reduction from unstructured search

Second, we generalize the method used in [Nay10] to give a lower bound for adaptive decision inversion without advice. Unlike in Nayak's original reduction, here we grant two-way access to the permutation. Recall that, in the unique search problem, one is granted quantum oracle access to a function $f : [N] \to \{0,1\}$ which is promised to satisfy either $|f^{-1}(1)| = 0$ or $|f^{-1}(1)| = 1$; the goal is to decide which is the case. The problem is formally defined below.

Definition 5.2. *($\mathsf{UNIQUESEARCH}_n$) Given a function $f : \{0,1\}^n \to \{0,1\}$, such that f maps at most one element to 1, output YES if $f^{-1}(1)$ is non-empty and NO otherwise.*

Definition 5.3. *(Distributional error) Suppose an algorithm solves a decision problem with error probability at most p_0 for NO instances and p_1 for YES instances. Then we say this algorithm has distributional error (p_0, p_1).*

We now establish a reduction from unstructured search to adaptive decision inversion.

Theorem 5.4. *If there exists a $(0, T, \delta)$-aDPI, then there exists a quantum algorithm that solves $\mathsf{UNIQUESEARCH}_{n-m-1}$ with at most $2T$ queries and distributional error $\left(\frac{1}{2} - \delta, \frac{1}{2}\right)$.*

Proof. Our proof is similar to that of Nayak [Nay10]: given a $(0, T, \delta)$-aDPI $\mathcal{A}$, we construct another algorithm $\mathcal{B}$ which solves the $\mathsf{UNIQUESEARCH}_{n-m-1}$ problem.

Let $N = 2^n$. For any uniform image $t \in [N]$, define the NO and YES instances sets (corresponding to the image t) of the decision permutation inversion problem with size N:

$$\pi_{t,0} = \{\pi : \pi \text{ is a permutation on } [N], \text{ the first bit of } \pi^{-1}(t) \text{ is } 0\},$$
$$\pi_{t,1} = \{\pi : \pi \text{ is a permutation on } [N], \text{ the first bit of } \pi^{-1}(t) \text{ is } 1\}.$$

Note that for a random permutation π, whether $\pi \in \pi_{t,0}$ or $\pi_{t,1}$ simply depends on the choice of t. Since t is uniform, $\Pr[\pi \in \pi_{t,0}] = \Pr[\pi \in \pi_{t,1}] = 1/2$. We also consider functions $h : [N] \to [N]$ with a unique collision at t. One of the colliding pairs should have the first bit 0, and the other one should have the first bit 1. Moreover, the last m bits of the colloding pair is μ. Formally speaking, $h(0\|i\|\mu) = h(1\|j\|\mu) = t$, where $i, j \in \{0,1\}^{n-m-1}$. Let $Q_{t,\mu}$ denote the set of all such functions.

Furthermore, given a permutation π on $[N]$, consider functions in $Q_{t,\mu}$ that differ from π at exactly one point. These are functions h with a unique collision and the collision is at t. If $\pi \in \pi_{t,0}$, $\pi(0\|i\|\mu) = h(0\|i\|\mu) = t$ and $1\|j\|\mu$ is the unique point where π and h differ; if $\pi \in \pi_{t,1}$, $\pi(1\|j\|\mu) = h(1\|j\|\mu) = t$ and $0\|i\|\mu$ is the unique point where π and h differ. Let $Q_{\pi,t,\mu}$ denote the set of such functions h and clearly $Q_{\pi,t,\mu} \subseteq Q_{t,\mu}$. Note that if we pick a random permutation π in $\{\pi_N\}$ and choose a uniform random $h \in Q_{\pi,t,\mu}$, h is also uniform in $Q_{t,\mu}$. Next, we construct an algorithm $\mathcal{B}$ that tries to solve $\mathsf{UNIQUESEARCH}_{n-m-1}$ as follows, with quantum oracle access to f:

1. $\mathcal{B}$ first samples some randomness $r \in \{0,1\}^*$, a uniform random string $s \in \{0,1\}^{n-m}$ and a permutation $\pi \in \{\pi_N\}$.

2. $\mathcal{B}$ then runs $\mathcal{A}$ with quantum orale access to π, π^{-1} until it receives a string $\mu \in \{0,1\}^m$ from $\mathcal{A}$.

3. Let $t = \pi(s\|\mu)$, and then it follows that if $s|_0 = 0$, $\pi \in \pi_{t,0}$, and otherwise $\pi \in \pi_{t,1}$.

4. $\mathcal{B}$ then constructs a function $h_{f,\pi,t,\mu}$ and $h^{-1*}_{f,\pi,t,\mu}$ as follows. If $\pi \in \pi_{t,0}$, for any $i \in \{0,1\}$ and $j \in \{0,1\}^{n-m-1}$,

$$h_{f,\pi,t,\mu}(i\|j\|u) = \begin{cases} t & \text{if } i = 1 \text{ and } f(j) = 1, u = \mu, \\ \pi(i\|j\|u) & \text{otherwise.} \end{cases} \tag{2}$$

If $\pi \in \pi_{t,1}$, for any $i \in \{0,1\}$ and $j \in \{0,1\}^{n-m-1}$,

$$h_{f,\pi,t,\mu}(i\|j\|\mu) = \begin{cases} t & \text{if } i = 0 \text{ and } f(j) = 1, u = \mu, \\ \pi(i\|j\|u) & \text{otherwise.} \end{cases} \tag{3}$$

No matter what instance sets π belongs to, the corresponding "inverse" function is defined as

$$h^{-1*}_{f,\pi,t,\mu}(k||b) = \begin{cases} \pi^{-1}(k)\|0 & \text{if } b = 0 \text{ and } k \neq t, \\ 1\|1 & \text{otherwise.} \end{cases} \tag{4}$$

5. $\mathcal{B}$ then sends t, μ and r to $\mathcal{A}$, runs it with quantum oracle access to $h_{f,\pi,t,\mu}$ and $h^{-1*}_{f,\pi,t,\mu}$, and finally gets back b'. For simplicity, we write this process as $b' \leftarrow \mathcal{A}^{h_{\perp t}}(t, \mu, r)$. [2]

[2]Note that those functions are defined classically above, and its allowance for quantum oracle access is discussed in Appendix C, which gives $2q$ queries in the theorem statement.

6. $\mathcal{B}$ outputs b' if $\pi \in \pi_{t,0}$, and $1-b'$ if $\pi \in \pi_{t,1}$.

Let δ_1 be the error probability of $\mathcal{A}$ in the YES case and δ_0 be that in the NO case of $(0,T,\delta)$-DPI. Since s is uniform random and then $\Pr[\pi \in \pi_{t,0}] = \Pr[\pi \in \pi_{t,1}] = 1/2$, it follows that

$$\Pr[\text{error of } \mathcal{A}] = 1 - \left(\frac{1}{2} + \delta\right) = \frac{1}{2}(\delta_0 + \delta_1) \Rightarrow \delta = \frac{1}{2} - \frac{1}{2}(\delta_0 + \delta_1).$$

We now analyze the error probability of $\mathcal{B}$ in the YES and NO cases. In the NO case, $f^{-1}(1)$ is empty, so no matter whether $\pi \in \pi_{t,0}$ or $\pi \in \pi_{t,1}$, $h_{f,\pi,t,\mu} = \pi$. It follows that $\mathcal{A}^{h_{\perp t}}(t,r) = \mathcal{A}^{\pi_{\perp t}}(t,r)$. Therefore,

$$\begin{aligned}
\Pr[\text{error of } \mathcal{B} \text{ in NO case}] &= \Pr\left[1 \leftarrow \mathcal{B}^{\mathcal{O}_f}(\cdot)\right] \\
&= \Pr\left[1 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | \pi \in \pi_{t,0}\right] \Pr[\pi \in \pi_{t,0}] \\
&\quad + \Pr\left[0 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | \pi \in \pi_{t,1}\right] \Pr[\pi \in \pi_{t,1}] \\
&= \frac{1}{2}\left(\Pr[1 \leftarrow \mathcal{A}^{\pi_{\perp t}}(t,r) | \pi \in \pi_{t,0}] + \Pr[0 \leftarrow \mathcal{A}^{\pi_{\perp t}}(t,r) | \pi \in \pi_{t,1}]\right) \\
&= \frac{1}{2}\left(\Pr[\text{error of } \mathcal{A} \text{ in NO case}] + \Pr[\text{error of } \mathcal{A} \text{ in YES case}]\right) \\
&= \frac{1}{2}\left(\delta_0 + \delta_1\right) = \frac{1}{2} - \delta.
\end{aligned}$$

In the YES case, $f^{-1}(1)$ is not empty, so function $h_{f,\pi,t,\mu}$ has a unique collision at t, with one of the colliding pair having first bit 0 and the other one having first bit 1, no matter $\pi \in \pi_{t,0}$ or $\pi_{t,1}$. As f is a black-box function, the place j where $f(j) = 1$ is uniform and so $h_{f,\pi,t,\mu}$ is uniform in $Q_{\pi,t,\mu}$. By arguments at the beginning of this proof, as π is uniform, the function is also uniform in $Q_{t,\mu}$. Let $p := \Pr_{h_{f,\pi,t,\mu} \leftarrow Q_{t,\mu}}[0 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r)]$. Therefore,

$$\begin{aligned}
\Pr[\text{error of } \mathcal{B} \text{ in YES case}] &= \Pr\left[0 \leftarrow \mathcal{B}^{f}(\cdot)\right] \\
&= \Pr\left[0 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | \pi \in \pi_{t,0}\right] \Pr[\pi \in \pi_{t,0}] \\
&\quad + \Pr\left[1 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | \pi \in \pi_{t,1}\right] \Pr[\pi \in \pi_{t,1}] \\
&= \frac{1}{2}\Big(\Pr\left[0 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | h_{f,\pi,t,\mu} \xleftarrow{\$} Q_{t,\mu}\right] \\
&\quad + \Pr\left[1 \leftarrow \mathcal{A}^{h_{\perp t}}(t,r) | h_{f,\pi,t,\mu} \xleftarrow{\$} Q_{t,\mu}\right]\Big) \\
&= \frac{1}{2}\left(p + (1-p)\right) = \frac{1}{2}.
\end{aligned}$$

where the third equality comes from the fact stated above: no matter $\pi \in \pi_{t,0}$ or $\pi \in \pi_{t,1}$, the corresponding h is uniform in $Q_{t,\mu}$ and then can be viewed as uniform randomly generated from $Q_{t,\mu}$. Since $\mathcal{A}$ is granted with oracle access to h, both conditions can be changed to $h_{f,\pi,t,\mu} \xleftarrow{\$} Q_{t,\mu}$. Note that given h, even if $\mathcal{A}$ can notice that it is not a permutation and then acts arbitrarily, this can only influence the probability of two terms individually, i.e. the value of p and $1-p$. But as we only care about their summation, we do not need to handle the consequence of $\mathcal{A}$ noticing the difference, including the probability of oracle distinguishability. $\square$

6 Lower bounds

6.1 Search version

We now give lower bounds for the search version of the permutation inversion problem over $[N]$. We begin with a lower bound for a restricted class of inverters (and its formal definition); these inverters succeed on an ϵ-fraction of inputs with constant probability (say, 2/3.). The proof uses a similar approach as in previous works on one-sided permutation inversion with advice [NABT14, CLQ19, HXY19].

Theorem 6.1. *Let $N \in \mathbb{N}$. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be a $(S, T, 2\epsilon/3)$-SPI that satisfies*

$$\Pr_{\pi,y}\left[\Pr_{r}\left[\pi^{-1}(y) \leftarrow \mathsf{S}_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \epsilon.$$

We call those inverters restricted inverters. Suppose that $\epsilon = \omega(1/N)$, $T = o(\epsilon\sqrt{N})$ and $S \geq 1$. Then, for sufficiently large N we have $ST^2 \geq \widetilde{\Omega}(\epsilon N)$.

Proof. To prove the claim, we construct a QRAC-VL scheme that encodes the function π^{-1} and then derive the desired space-time trade-off via Theorem 2.3. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an $2\epsilon/3$-SPI that succeeds on a ϵ-fraction of inputs with probability at least 2/3. In other words, S satisfies

$$\Pr_{\pi,y}\left[\Pr_{r}\left[\pi^{-1}(y) \leftarrow \mathsf{S}_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \epsilon.$$

By the averaging argument in Lemma A.3 with parameter $\theta = 1/2$, it follows that there exists a large subset $\mathcal{X} \subseteq \mathcal{S}_N$ of permutations with size at least $N!/2$ such that for any permutation $\pi \in \mathcal{X}$, we have that

$$\Pr_{y}\left[\Pr_{r}\left[\pi^{-1}(y) \leftarrow \mathsf{S}_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \frac{\epsilon}{2}.$$

For a given permutation $\pi \in \mathcal{X}$ we let $\mathcal{I}$ be the set of indices $x \in [N]$ such that S correctly inverts $\pi(x)$ with probability at least 2/3 over the choice of r. By the definition of the set $\mathcal{X}$, we have that $|\mathcal{I}| \geq \epsilon/2 \cdot N$. Our QRAC-VL scheme $(\mathsf{Enc}, \mathsf{Dec})$ for encoding permutations is described in detail in Protocol 2. Below, we introduce some additional notations that will be relevant to the scheme. For convenience, we model the two-way accessible oracle given to S_1 in terms of a single oracle for the *merged* function of the form [3]

$$\pi_{\perp y}(w, a) \overset{\text{def}}{=} \begin{cases} \pi(w) & \text{if } a = 0 \\ \pi^{-1}(w) & \text{if } w \neq y \wedge a = 1 \\ \perp & \text{if } w = y \wedge a = 1. \end{cases}$$

Let $c, \gamma \in (0, 1)$ be parameters. As part of the encoding, we use the shared randomness $R \in \{0, 1\}^*$ to sample a subset $\mathcal{R} \subseteq [N]$ such that each element of $[N]$ is contained in $\mathcal{R}$ with probability $\gamma/T(\mathsf{S})^2$. Moreover, we define the following two disjoint subsets of $[N] \times \{0, 1\}$:

$$\Sigma_0^{\mathcal{R}} = \mathcal{R} \setminus \{x\} \times \{0\}$$
$$\Sigma_1^{\mathcal{R}} = \pi(\mathcal{R}) \setminus \{\pi(x)\} \times \{1\}.$$

Let $\mathcal{G} \subseteq \mathcal{I}$ be the set of $x \in [N]$ which satisfy the following two properties:

[3]The (reversible) quantum oracle implementation is similar to the one in Definition 3.3. We use the function $\pi_{\perp y}$ for ease of presentation since the same proof carries over with minor modifications in the quantum oracle case.

1. The element x is contained in the set $\mathcal{R}$, i.e.

$$x \in \mathcal{R}; \tag{5}$$

2. The total query magnitude of $\mathsf{S}_1^{\pi_{\perp y}}$ with input $(\mathsf{S}_0(\pi, r), y, r)$ on the set $\Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}$ is bounded by $c/T(\mathsf{S})$. In other words, we have

$$q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \;\leq\; c/T(\mathsf{S}). \tag{6}$$

Claim 1. Let $\mathcal{G} \subseteq [N]$ be the set of x which satisfy the conditions in (5) and (6). Then, there exist constants $\gamma, c \in (0,1)$ such that

$$\Pr_{\mathcal{R}}\left[|\mathcal{G}| \geq \frac{\epsilon\gamma N}{4\,T(\mathsf{S})^2}\left(1 - \frac{5\gamma^2}{c}\right)\right] \geq 0.8.$$

In other words, we have $|\mathcal{G}| = \Omega(\epsilon N/T(\mathsf{S})^2)$ with high probability.

Proof. (of the claim) Let $\mathcal{H} = \mathcal{R} \cap \mathcal{I}$ denote the set of $x \in \mathcal{R}$ for which S correctly inverts $\pi(x)$ with probability at least $2/3$ over the choice of r. By the definition of the set $\mathcal{R}$, it follows that $|\mathcal{H}|$ has a binomial distribution. Therefore, in expectation, we have that $|\mathcal{H}| = \gamma|\mathcal{I}|/T(\mathsf{S})^2$. Using the multiplicative Chernoff bound in Lemma A.1 and the fact that $T(\mathsf{S}) = o(\epsilon\sqrt{N})$, we get

$$\Pr_{\mathcal{R}}\left[|\mathcal{H}| \geq \frac{\gamma|\mathcal{I}|}{2\,T(\mathsf{S})^2}\right] \geq 0.9, \tag{7}$$

for all sufficiently large N. Because each query made by S_1 has unit length and because S_1 makes at most $T(\mathsf{S})$ queries, it follows that

$$q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0,1\}) \leq T(\mathsf{S}). \tag{8}$$

We obtain the following upper bound for the average total query magnitude:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}})\right] \\
&= \mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}}) + q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_1^{\mathcal{R}})\right] && (\Sigma_0^{\mathcal{R}}, \Sigma_1^{\mathcal{R}} \text{ are disjoint}) \\
&= \mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}})\right] + \mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_1^{\mathcal{R}})\right] && (\text{linearity of expectation}) \\
&= \mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \mathcal{R} \setminus \{x\} \times \{0\})\right] + \mathop{\mathbb{E}}_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \pi(\mathcal{R}) \setminus \{\pi(x)\} \times \{1\})\right] \\
&= \frac{\gamma}{T(\mathsf{S})^2} \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{x\} \times \{0\}) + \frac{\gamma}{T(\mathsf{S})^2} \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, \pi([N]) \setminus \{\pi(x)\} \times \{1\}) \\
&= \frac{\gamma}{T(\mathsf{S})^2} \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{x\} \times \{0\}) \\
&\quad + \frac{\gamma}{T(\mathsf{S})^2} \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{\pi(x)\} \times \{1\}) && (\pi \text{ is a permutation}) \\
&\leq \frac{\gamma}{T(\mathsf{S})^2} \cdot \left[q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0\}) + q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{1\})\right] && (\text{supersets}) \\
&= \frac{\gamma}{T(\mathsf{S})^2} \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0,1\}) \;\leq\; \frac{\gamma}{T(\mathsf{S})}. && (\text{by the inequality in (8)})
\end{aligned}$$

Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \geq \frac{c}{T(\mathsf{S})}\right] \leq \frac{T(\mathsf{S})}{c} \cdot \frac{\gamma}{T(\mathsf{S})} = \frac{\gamma}{c}. \tag{9}$$

Let us now denote by $\mathcal{J}$ the subset of $x \in \mathcal{I}$ that satisfy Eq. (5) but not Eq. (6). Note that Eq. (5) and Eq. (6) are independent for each $x \in \mathcal{I}$, since Eq. (5) is about whether $x \in \mathcal{R}$ and Eq. (6) only concerns the intersection of $\mathcal{R}$ and $[N] \setminus \{x\}$, as well as $\pi(\mathcal{R})$ and $\pi([N]) \setminus \{\pi(x)\}$. Therefore, by (9), the probability that $x \in \mathcal{I}$ satisfies $x \in \mathcal{J}$ is at most $\gamma^2/(cT(\mathsf{S})^2)$. Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}}\left[|\mathcal{J}| \leq \frac{10|\mathcal{I}|\gamma^2}{cT(\mathsf{S})^2}\right] \geq 0.9. \tag{10}$$

Using (7) and (10), we get with probability at least 0.8 over the the choice of $\mathcal{R}$,

$$|\mathcal{G}| = |\mathcal{H}| - |\mathcal{J}| \geq \frac{|\mathcal{I}|\gamma}{2T(\mathsf{S})^2} - \frac{10|\mathcal{I}|\gamma^2}{cT(\mathsf{S})^2} \geq \frac{\epsilon\gamma N}{4T(\mathsf{S})^2}\left(1 - \frac{5\gamma^2}{c}\right),$$

given that γ is a sufficiently small positive constant. □

Protocol 2 (Quantum Random Access Code For Inverting Permutations)**.**
Let $c, \gamma \in (0,1)$ be parameters. Consider the following (variable-length) quantum random-access code given by $\mathsf{QRAC\text{-}VL} = (\mathsf{Enc}, \mathsf{Dec})$ *defined as follows:*

- $\mathsf{Enc}(\pi^{-1}; R)$*: On input $\pi^{-1} \in \mathcal{S}_N$ and randomness $R \in \{0,1\}^*$, first use R to extract random coins r and then proceed as follows:*

  Case 1: *$\pi \notin \mathcal{X}$ or $|\mathcal{G}| < \frac{\epsilon\gamma N}{4T(\mathsf{S})^2}\left(1 - \frac{5\gamma^2}{c}\right)$. Use the classical flag* `case` $= 1$ *(taking one additional bit) and output the entire permutation table of π^{-1}.*

  Case 2: *$|\mathcal{G}| \geq \frac{\epsilon\gamma N}{4T(\mathsf{S})^2}\left(1 - \frac{5\gamma^2}{c}\right)$. Use the classical flag* `case` $= 2$ *(taking one additional bit) and output the following*

  1. *The size of $\mathcal{G}$, encoded using $\log N$ bits;*
  2. *the set $\mathcal{G} \subseteq \mathcal{R}$, encoded using $\log \binom{|\mathcal{R}|}{|\mathcal{G}|}$ bits;*
  3. *The permutation π restricted to inputs outside of $\mathcal{G}$, encoded using $\log(N!/|\mathcal{G}|!)$ bits;*
  4. *Quantum advice used by the algorithm repeated ρ times with $\alpha^{\otimes\rho}$, for $\alpha \leftarrow \mathsf{S}_0(\pi, r)$ for some ρ that we will decide later. (We can compute this as the encoder can preprocess multiple copies of the same advice. Note that this is the only part of our encoding that is not classical.)*

- $\mathsf{Dec}(\beta, y; R)$*: On input encoding β, image $y \in [N]$ and randomness $R \in \{0,1\}^*$, first use R to extract random coins r and then proceed as follows:*

  Case 1: *This corresponds to the flag* `case` $= 1$*. Search the permutation table for π^{-1} and output x such that $\pi^{-1}(y) = x$.*

Case 2: *This corresponds to the flag* $\texttt{case} = 2$*. Recover* $\mathcal{G}$ *and* $\pi(x)$ *for every* $x \notin \mathcal{G}$*. If* $y = \pi(x)$ *for some* $x \notin \mathcal{G}$*, output* $x = \pi^{-1}(y)$*. Otherwise, parse* $\alpha_1, \alpha_2, \dots, \alpha_\rho$ *and run* $\mathsf{S}_1^{\bar{\pi}_{\perp y}}(\alpha_i, y, r)$ *for each* $i \in [\rho]$ *and output their majority vote, where we let* [a]

$$\bar{\pi}_{\perp y}(w, a) = \begin{cases} y & \text{if } w \in \mathcal{G} \wedge a = 0 \\ \pi(w) & \text{if } w \notin \mathcal{G} \wedge a = 0 \\ \pi^{-1}(w) & \text{if } w \notin \pi(\mathcal{G}) \wedge a = 1 \\ \perp & \text{if } w \in \pi(\mathcal{G}) \wedge a = 1. \end{cases}$$

[a]The (reversible) quantum oracle implementation for $\bar{\pi}_{\perp y}$ is provided in Appendix D.

Let us now analyze the performance of our $\mathsf{QRAC\text{-}VL}$ scheme $(\mathsf{Enc}, \mathsf{Dec})$ in Protocol 2. Let $|\Psi_{\pi_{\perp y}}\rangle$ and $|\Psi_{\bar{\pi}_{\perp y}}\rangle$ denote the final states of S_1 when it is given the oracles $\pi_{\perp y}$ and $\bar{\pi}_{\perp y}$, respectively. By Lemma 2.1 and the properties of the total query magnitude:

$$\begin{aligned} \| \, |\Psi_{\pi_{\perp y}}\rangle - |\Psi_{\bar{\pi}_{\perp y}}\rangle \, \| &\leq \sqrt{T(\mathsf{S}) \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, \mathcal{G} \setminus \{x\} \times \{0\}) \cup (\pi(\mathcal{G}) \setminus \{\pi(x)\} \times \{1\})} \\ &\leq \sqrt{T(\mathsf{S}) \cdot q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}})} \\ &\leq \sqrt{T(\mathsf{S}) \cdot \frac{c}{T(\mathsf{S})}} = \sqrt{c}. \end{aligned}$$

Since $x \in \mathcal{I}$, it follows from the definition of $\mathcal{I}$ that measuring $|\Psi_{\pi_{\perp y}}\rangle$ results in x with probability at least $2/3$. Given a small enough positive constant c, we can ensure that measuring $|\Psi_{\bar{\pi}_{\perp y}}\rangle$ will result in x with probability at least 0.6. We now examine the length of our encoding. With probability $1 - \epsilon/2$, we have $\pi \notin \mathcal{X}$; with probability $\epsilon(1 - 0.8)/2$, we have $\pi \in \mathcal{X}$ but $\mathcal{G}$ is small, i.e.,

$$|\mathcal{G}| < \frac{\epsilon \gamma N}{4\, T(\mathsf{S})^2} \left(1 - \frac{5\gamma^2}{c}\right).$$

Therefore, except with probability $1 - 0.4\epsilon$, our encoding will result in the flag $\texttt{case} = 1$, where the encoding consists of $1 + \log N!$ classical bits and the decoder succeeds with probability 1. With probability 0.4ϵ , our encoding has the flag $\texttt{case} = 2$, and the size equals

$$1 + \log N + \log \binom{|\mathcal{R}|}{|\mathcal{G}|} + \log(N!/|\mathcal{G}|!) + \rho S(\mathsf{S}).$$

By the assumption that $T(\mathsf{S}) = o(\epsilon\sqrt{N})$, we have

$$\begin{aligned} \log \binom{|\mathcal{R}|}{|\mathcal{G}|} &= \log \left(\frac{|\mathcal{R}|(|\mathcal{R}| - 1) \dots (|\mathcal{R}| - |\mathcal{G}| + 1)}{|\mathcal{G}|(|\mathcal{G}| - 1) \dots 1} \right) \\ &= O\left(\log \left(\frac{|\mathcal{R}||\mathcal{R}| \dots |\mathcal{R}|}{|\mathcal{G}||\mathcal{G}| \dots |\mathcal{G}|} \right) \right) \\ &= O(|\mathcal{G}| \log(|\mathcal{R}|/|\mathcal{G}|)) \\ &= O(|\mathcal{G}| \log 1/\epsilon) \\ &= o(|\mathcal{G}| \log |\mathcal{G}|), \end{aligned}$$

and we can rewrite the size of the encoding as

$$\log N + o(|\mathcal{G}| \log |\mathcal{G}|) + \log N! - \log |\mathcal{G}|! + \rho S(\mathsf{S}).$$

In the case when the decoder is queried on an input that is already known, that is $y \notin \pi(\mathcal{G})$ (which occurs with probability $1 - |\mathcal{G}|/N$), the decoder recovers the correct pre-image with probability 1. Otherwise, the analysis is the following: with just one copy of the advice, the decoder recovers the correct pre-image with probability $2/3$, and hence with ρ many copies, the decoder can take the majority vote and recover the correct pre-image with probability $1-\exp(-\Omega(\rho))$. The latter follows from the Chernoff bound in Lemma A.1. Overall, the average encoding length is

$$0.4\epsilon \cdot (\log N + o(|\mathcal{G}| \log |\mathcal{G}|) - \log |\mathcal{G}|! + \rho S(\mathsf{S})) + \log N!$$

where the average success probability is $1 - |\mathcal{G}|/N \cdot \exp(-\Omega(\rho))$. By setting $\rho = \Omega(\log(N/\epsilon)) = \Omega(\log N)$, the average success probability amounts to $1 - O(1/N^2)$. Therefore, using the lower bound in Theorem 2.3, we have

$$\begin{aligned}
\log N! + 0.4\epsilon \cdot (\log N + o(|\mathcal{G}| \log |\mathcal{G}|) - \log |\mathcal{G}|! + \rho S(\mathsf{S})) &\geq \log N! - O\left(\frac{1}{N} \log N\right) \\
\log N + o(|\mathcal{G}| \log |\mathcal{G}|) - \log |\mathcal{G}|! + \rho S(\mathsf{S}) &\geq -O\left(\log N\right) \\
\rho S(\mathsf{S}) + O\left(\log N\right) &\geq \log |\mathcal{G}|! - o(|\mathcal{G}| \log |\mathcal{G}|) \\
S(\mathsf{S}) \log N &\geq \Omega(\log |\mathcal{G}|! - o(|\mathcal{G}| \log |\mathcal{G}|))
\end{aligned}$$

where the second and the last equality comes from the fact that $\epsilon = \omega(1/N)$ and $\rho = \Omega(\log N)$, respectively. Since $\log |\mathcal{G}|! = O(|\mathcal{G}| \log |\mathcal{G}|)$, it follows that

$$\begin{aligned}
S(\mathsf{S}) \log N &\geq \Omega(O(|\mathcal{G}| \log |\mathcal{G}|) - o(|\mathcal{G}| \log |\mathcal{G}|)) \\
S(\mathsf{S}) \log N &\geq \Omega(|\mathcal{G}| \log |\mathcal{G}|).
\end{aligned}$$

As we are conditioning on the event that $\mathcal{G}$ is large, plugging in the lower bound on $|\mathcal{G}|$, we have that, for sufficiently large N, $S(\mathsf{S}) \geq \widetilde{\Omega}(|\mathcal{G}|)$, and thus

$$S(\mathsf{S}) \cdot T(\mathsf{S})^2 \geq \widetilde{\Omega}(\epsilon N).$$

This gives the desired space-time trade-off. □

We remark that the search inverter we consider in Theorem 6.1 succeeds on more than just a constant number of inputs, that is $\epsilon = \omega(1/N)$, and beats the time complexity of $T = \Omega(\sqrt{\epsilon N})$ which is required for unstructured search using Grover's algorithm. [Gro96, DH08, Zha19]. Next, we remove the restriction on the inverter by applying amplification (specifically, Corollary 4.2.) This yields a lower bound in the full average-case version of the search inversion problem.

Theorem 6.2. *Let S be a (S,T,ϵ)-SPI for some $\epsilon > 0$. Suppose that $\epsilon = \omega(1/N)$, $T = o(\epsilon^2\sqrt{N})$, and $S \geq 1$. Then, for sufficiently large N we have*

$$S(\mathsf{S}) \cdot T(\mathsf{S})^2 \geq \widetilde{\Omega}(\epsilon^3 N).$$

Proof. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an (S, T, ϵ)-SPI, for some $\epsilon > 0$. Using Corollary 4.2, we can construct an SPI $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ with space and time complexities

$$S(\mathsf{S}[\ell]) = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil \cdot S(\mathsf{S}) \quad \text{and} \quad T(\mathsf{S}[\ell]) = \left(\left\lceil \frac{\ln(10)}{\epsilon} \right\rceil + 1 \right) \cdot T(\mathsf{S})$$

such that

$$\Pr_{\pi,y}\left[\Pr_r\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\mathsf{S}[\ell]_0(\pi, r), y, r)\right] \geq \frac{2}{3}\right] \geq \frac{1}{5}.$$

From Theorem 6.1 it follows that for sufficiently large $N \geq 1$,

$$S(\mathsf{S}[\ell]) \cdot T(\mathsf{S}[\ell])^2 \geq \widetilde{\Omega}(N).$$

Plugging in the expressions for $S(\mathsf{S}[\ell])$ and $T(\mathsf{S}[\ell])$, we get that with assumption

$$\epsilon = \omega(1/N), \qquad T(\mathsf{S}) = o(\epsilon^2\sqrt{N}) \quad \text{and} \quad S(\mathsf{S}) \geq 1,$$

the trade-off between space and time complexities is

$$S(\mathsf{S}) \cdot T(\mathsf{S})^2 \geq \widetilde{\Omega}(\epsilon^3 N).$$

□

Note that we incur a loss (ϵ^3 versus ϵ) in our search lower bound due to the fact that we need to amplify the *restricted* search inverter in Theorem 6.1. This results in a multiplicative overhead of $\Theta(1/\epsilon)$ in terms of space and time complexity, as compared to the restricted inverter. We remark that a similar loss as a result of amplification is also inherent in [HXY19].

6.2 Decision version

6.2.1 Space-time tradeoff, no adaptive sampling

The search lower bound of Theorem 6.2, when combined with the search-to-decision reduction of Theorem 5.1, yields a lower bound for the decision version.

Corollary 6.3. *Let D be a (S, T, δ)-DPI for some $\delta > 0$. Suppose that $\delta = \omega(1/N)$ and $T = \tilde{o}\left(\delta^2\sqrt{N}\right)$ and $S \geq 1$. Then, for sufficiently large N we have*

$$S(\mathsf{D}) \cdot T(\mathsf{D})^2 \gtrapprox \widetilde{\Omega}\left(\delta^6 N\right).$$

Proof. Let $N = 2^n$. Given a $(S(\mathsf{D}), T(\mathsf{D}), \delta)$-DPI $= (\mathsf{D}_0, \mathsf{D}_1)$ where D_0 outputs S-qubit state and D_1 makes T queries, one can construct an $(S(\mathsf{S}), T(\mathsf{S}), \eta)$-SPI $= (\mathsf{S}_0, \mathsf{S}_1)$ by Theorem 5.1 with $\eta \geq 1 - \mathsf{negl}(n)$, and with space and time complexities

$$S(\mathsf{S}) = n\ell S(\mathsf{D}) \quad \text{and} \quad T(\mathsf{S}) = n\ell T(\mathsf{D})$$

where $\ell = \Omega\left(\frac{n(1+2\delta)}{\delta^2}\right)$. It directly follows from Theorem 6.2 that with conditions

$$\delta = \omega(1/N), \qquad S(\mathsf{D}) \geq 1,$$
$$T(\mathsf{D}) = \frac{1}{n\ell} \cdot o(\eta\sqrt{N}) = o\left(\frac{\delta^2}{n^2(1+2\delta)}\sqrt{N}\right) = \tilde{o}\left(\delta^2\sqrt{N}\right),$$

S satisfies the space-time trade-off lower bound

$$n^3 \left(\frac{n(1+2\delta)}{\delta^2}\right)^3 S(\mathsf{D}) \cdot T(\mathsf{D})^2 \geq \widetilde{\Omega}(\eta^3 N) \approx \widetilde{\Omega}(N)$$
$$S(\mathsf{D}) \cdot T(\mathsf{D})^2 \gtrapprox \widetilde{\Omega}\left(\delta^6 N\right)$$

for sufficiently large N. □

Similar to the search lower bound from before, we incur a loss that amounts to a factor δ^6. This results from our specific approach which is based on the search-to-decision reduction in Theorem 5.1. We believe that our lower bound could potentially be improved even further.

6.2.2 Time lower bound, adaptive sampling

In the case of an adaptive decision inverter without advice, we can get a tight bound by means of the reduction from the unique search problem (Theorem 5.4), combined with well-known lower bounds on the average-case unique search problem.

Theorem 6.4. *Let* D *be a* $(0, T, \delta)$*-aDPI. Then* $T^2 \geq \Omega(\delta N/M)$.

Proof. Since D is a $(0, T, \delta)$-DPI, by the lower bound of unique search problem [Gro96, Zal99, Nay10, Zha19], we get a $2T$-query algorithm for $\mathsf{UNIQUESEARCH}_{n-1}$ with distributional error $(\frac{1}{2} - \delta, \frac{1}{2})$. Since the YES and NO cases are uniformly distributed, we can write the overall error probability as $\frac{1}{2}\left(\frac{1}{2} - \delta\right) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} - \frac{\delta}{2}$. Then by the lower bound of unique search, we have

$$1 - \left(\frac{1}{2} - \frac{\delta}{2}\right) \leq \frac{1}{2} + O\left(\frac{(2T)^2}{2^{n-m}}\right)$$
$$T^2 \geq \Omega(\delta \cdot 2^{n-m})$$
$$T^2 \geq \Omega\left(\frac{\delta N}{M}\right).$$

We note that with non-adaptive D, i.e. $m = 0$, the above bound reduces to query lower bound $T^2 \geq \Omega\left(\delta N\right)$. □

7 Applications

In this section, we give a plausible security model for symmetric-key encryption and a scheme whose security in that model is based on the hardness of our adaptive two-sided permutation inversion problem. Recall that a symmetric-key encryption scheme consists of three algorithms:

- (key generation) Gen: given randomness s and security parameter n; outputs key $k := \mathsf{Gen}(1^n; s)$;
- (encryption) Enc: given key k, plaintext m, and randomness r; outputs ciphertext $c := \mathsf{Enc}_k(m; r)$;
- (decryption) Dec: given key k, ciphertext c; outputs plaintext $m := \mathsf{Dec}_k(c)$.

When the key randomness is to be selected uniformly, we suppress it and simply write $\mathsf{Gen}(1^n)$.

Consider the following security definition.

Definition 7.1. *(*OW-QCCRA2*) Let* SKE $= (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a private-key encryption scheme. We say that* SKE *is* OW-QCCRA2 *if the advantage for any quantum polynomial-time adversary $\mathcal{A}$ in the following OW-QCCRA2 experiment is at most negligible:*

1. *A key k is generated by running $\mathsf{Gen}(1^n; s)$;*
2. *$\mathcal{A}$ gets quantum oracle access to $\mathsf{Enc}_k(\,\cdot\,;\,\cdot\,)$ and $\mathsf{Dec}_k(\cdot)$, and then outputs a $(m-1)$-bit string μ and a quantum state ρ with size S. Let $t(n)$ be the number of quantum queries that $\mathcal{A}$ makes in this phase.*
3. *Uniform $b \in \{0,1\}$ and $r \in \{0,1\}^{n-1}$ are chosen, and a challenge ciphertext $c = \mathsf{Enc}_k(b\|\mu; r)$ is computed and given to $\mathcal{A}$;*
4. *$\mathcal{A}$ gets quantum oracle access to $\mathsf{Enc}_k(\,\cdot\,;\,\cdot\,)$ and $\mathsf{Dec}_k^{\perp c}(\cdot)$, and eventually outputs a bit b'. Let $\ell(n)$ be the number of quantum queries that $\mathcal{A}$ makes in this phase.*
5. *The experiment outputs 1, if $b' = b$, and 0 otherwise.*

We remark that, unlike in most definitions of security, here the adversary is allowed to choose both inputs to the encryption oracle: the plaintext as well as the randomness. To generate the challenge ciphertext, the coin r needs to be chosen truly randomly; otherwise, the scheme will degenerate into a deterministic one that cannot be secure. Moreover, we do not yet make any restriction on the computational power of $\mathcal{A}$, or on the functions t and ℓ.

Next, we define two simple encryption schemes.

RP Scheme. Consider the following (inefficient) scheme that uses uniformly random permutations.

- Gen is given 1^n and outputs a description k of a uniformly random permutation π on $\{0,1\}^{2n}$;
- Enc is given k, $m \in \{0,1\}^n$ and $r \in \{0,1\}^n$, and outputs $c := \pi(m\|r)$;
- Dec is given k and $c \in \{0,1\}^{2n}$, and outputs the first n bits of $\pi^{-1}(c)$.

Definition 7.2. *(ϵ-Qsecure PRP)[KL20, Zha16] Let $P_k : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$ be a permutation family. We call P_k a ϵ-Qsecure PRP if for any efficient quantum adversary $\mathcal{A}$ who makes q quantum queries, there exist a negligible function $\epsilon(\lambda)$ such that*

$$\left|\Pr\left[\mathcal{A}^{P_k(\cdot),P_k^{-1}(\cdot)}(1^n) = 1\right] - \Pr\left[\mathcal{A}^{\pi(\cdot),\pi^{-1}(\cdot)}(1^n) = 1\right]\right| \le \epsilon \cdot \mathrm{poly}(q)\,,$$

where $\pi : \{0,1\}^n \to \{0,1\}^n$ is a truly random permutation.

PRP Scheme. Let $\{P_k : \{0,1\}^{2n} \mapsto \{0,1\}^{2n}\}$ be a family of ϵ-Qsecure PRPs and consider the following scheme:

- Gen takes as input a security parameter 1^n and returns a key $k \in \{0,1\}^n$ for P_k;
- Enc is given key $k \in \{0,1\}^n$, $m \in \{0,1\}^n$ and $r \in \{0,1\}^n$, and outputs $c := P_k(m\|r)$;

- Dec is given key $k \in \{0,1\}^n$ and $c \in \{0,1\}^{2n}$, and outputs the first n bits of $P_k^{-1}(c)$.

Of course, any practical scheme should be efficient, and indeed we can show that the PRP scheme is $\mathsf{OW\text{-}QCCRA2}$ in two special cases: when there is no advice, i.e., $S = 0$ (we call this $\mathsf{OW\text{-}QCCRA2\text{-}v1}$) and when there is no adaptivity, i.e., $|\mu| = 0$ (we call this $\mathsf{OW\text{-}QCCRA2\text{-}v2}$). We are able to prove the following theorems.

Theorem 7.3. *The* PRP *scheme is* $\mathsf{OW\text{-}QCCRA2\text{-}v1}$. *In other words, for any quantum adversary $\mathcal{A}$ who makes $t(n)$ quantum queries in the pre-challenge phase and $\ell(n)$ quantum queries in the post-challenge phase, it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{OW\text{-}QCCRA2\text{-}v1}}_{\mathcal{A},\mathrm{PRP}}(1^n) = 1\right] \le \frac{1}{2} + \delta + \epsilon \cdot T(n).$$

Here, $\delta \le O(\frac{\ell^2 2^{n-1}}{2^{2n}})$, $T(n) = t(n) + \ell(n)$ and ϵ is a negligible function.

Proof. Given an adversary $\mathcal{A}$ that attacks the RP scheme in the $\mathsf{OW\text{-}QCCRA2}$ experiment described in Definition 7.1 with $S = 0$, we can construct a $(0, T, \delta)$-aDPI $\mathsf{aD} = (\mathsf{aD}_0, \mathsf{aD}_1)$ in the decision inversion experiment, which takes place as follows:

1. **(sample instance and coins)** a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is sampled;
2. **(prepare advice)** aD_0 is given the whole permutation table of π. Then it constructs oracles $\mathsf{Enc}(\cdot;\cdot) = \pi(\cdot\|\cdot)$ and $\mathsf{Dec}(\cdot) = \pi^{-1}(\cdot)$ and gives $\mathcal{A}$ quantum oracle access. aD_0 will get back a $(n-1)$-bit output string μ and then output it. Suppose $\mathcal{A}$ makes $t(n)$ quantum queries.
3. **(invert)** An instance $c = \pi(b\|\mu\|r)$ is computed, with $b \in \{0,1\}$ and $r \in \{0,1\}^n$ are sampled. aD_1 is run with c, auxiliary string μ and quantum oracle access $\mathcal{O}_\pi$ and $\mathcal{O}_{\pi^{-1}_{\perp y}}$. It then directly passes c and two oracles to $\mathcal{A}$ and gets back a bit b' and outputs it. Suppose $\mathcal{A}$ makes $\ell(n)$ quantum queries.
4. **(check)** If $b' = b$, output 1; otherwise output 0.

It trivially follows that

$$\Pr\left[\mathsf{Exp}^{\mathsf{OW\text{-}QCCRA2\text{-}v1}}_{\mathcal{A},\mathrm{RP}}(1^n) = 1\right] \le \Pr[\mathsf{DecisionInvert}_{\mathsf{aD}} = 1].$$

By assumption we have that, for all efficient quantum adversary $\mathcal{A}$, there exists a negligible ϵ such that

$$\left|\Pr\left[\mathcal{A}^{P_k(\cdot),P_k^{-1}(\cdot)}(1^n) = 1\right] - \Pr\left[\mathcal{A}^{\pi(\cdot),\pi^{-1}(\cdot)}(1^n) = 1\right]\right| \le \epsilon \cdot \mathrm{poly}(t(n) + \ell(n)),$$

Therefore

$$\begin{aligned}\Pr\left[\mathsf{Exp}^{\mathsf{OW\text{-}QCCRA2\text{-}v1}}_{\mathcal{A},\mathrm{PRP}}(1^n) = 1\right] &\le \Pr\left[\mathsf{Exp}^{\mathsf{OW\text{-}QCCRA2\text{-}v1}}_{\mathcal{A},\mathrm{RP}}(1^n) = 1\right] + \epsilon \cdot T(n)\\ &\le \Pr[\mathsf{DecisionInvert}_{\mathsf{aD}} = 1] + \epsilon \cdot T(n)\\ &= \frac{1}{2} + \delta + \epsilon T(n).\end{aligned}$$

Where $\delta \le O(\frac{\ell^2 2^{n-1}}{2^{2n}})$ by Theorem 6.4, and by Definition 7.2 ϵ is negligible. Remark that the above bound becomes $\frac{1}{2} + \mathsf{negl}(n)$ when $\mathcal{A}$ is a quantum polynomial time (QPT) adversary since both δ and ϵT are negligible when t and ℓ are of polynomial size.

∎

Theorem 7.4. *The* PRP *scheme is* OW-QCCRA2-v2*. In other words, for any quantum adversary* $\mathcal{A}$ *who makes* $t(n)$ *quantum queries in the pre-challenge phase and* $\ell(n)$ *quantum queries in the post-challenge phase, it holds that*

$$\Pr\left[\mathsf{Exp}^{\mathsf{OW\text{-}QCCRA2\text{-}v1}}_{\mathcal{A},\mathrm{PRP}}(1^n) = 1\right] \leq \frac{1}{2} + \delta + \epsilon \cdot T(n).$$

Here, $\delta \leq O(\frac{\ell^2 S}{2^{2n}})^{\frac{1}{6}}$*,* $T(n) = t(n) + \ell(n)$ *and* ϵ *is a negligible function.*

Proof. Given an adversary $\mathcal{A}$ that attacks the RP scheme in the OW-QCCRA2 experiment described in Definition 7.1 with $|\mu| = 0$, we can construct a (S, T, δ)-DPI $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$ in the decision inversion experiment. The construction is the same as Theorem 7.3, with slight modifications at the "prepare advice" and the "invert" step:

(prepare advice) D_0 is given the whole permutation table of π. Then it constructs oracles $\mathsf{Enc}(\cdot;\cdot) = \pi(\cdot\|\cdot)$ and $\mathsf{Dec}(\cdot) = \pi^{-1}(\cdot)$ and gives $\mathcal{A}$ quantum oracle access. D_0 will get back a S-qubit quantum state ρ and then output it. Suppose $\mathcal{A}$ makes $t(n)$ quantum queries.

(invert) An instance $c = \pi(b||r)$ is computed, with $b \in \{0,1\}$ and $r \in \{0,1\}^n$ are sampled. D_1 is run with c, quantum advice ρ and quantum oracle access $\mathcal{O}_\pi$ and $\mathcal{O}_{\pi^{-1}_{\perp y}}$. It then directly passes c and two oracles to $\mathcal{A}$ and gets back a bit b' and outputs it. Suppose $\mathcal{A}$ makes $\ell(n)$ quantum queries.

By following the same procedure as in Theorem 7.3 but using the bound of Corollary 6.3, we get the desired bound. ☐

Finally, we remark that the above results hold for the following strengthening of OW-QCCRA2, described as follows. Suppose that an encryption scheme satisfies the property that there exists an *alternative* decryption algorithm that can both compute the plaintext and also deduce the randomness that was initially used to encrypt. This property is true for the RP and PRP schemes, as well as some other standard encryption methods (e.g., Regev's secret-key LWE scheme, implicit in [Reg09]). For schemes in this category, one can also grant access to such an alternative decryption algorithm, thus expanding the form of "randomness access" that the adversary has. Our proofs show that the RP and PRP schemes are secure (in their respective setting) even against this form of additional adversarial power.

8 Future Work

For future applications, the two-sided permutation inversion problem appears naturally in the context of sponge hashing [GJMG11] which is used by the international hash function standard SHA3 [Dwo15]. Previous work [CGH+18, CMSZ21] studied the post-quantum security of the sponge construction where the block function is either a random function or a (non-invertible) random permutation. However, as the core permutation in SHA3 is public and efficiently invertible, the "right setting" of theoretical study is one in which the block function consists of an invertible permutation. This setting is far less understood, and establishing the security of the sponge in this setting is a major open problem in post-quantum cryptography. Our results on two-sided permutation inversion may serve as a stepping stone towards this goal.

References

[ABK+22] Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz, and Patrick Struck. "Post-Quantum Security of the (Tweakable) FX Construction, and Applications". In: *Cryptology ePrint Archive* (2022). URL: https://eprint.iacr.org/2022/1097 (cit. on p. 3).

[ABKM22] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. "Post-quantum security of the Even-Mansour cipher". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2022, pp. 458–487. DOI: https://doi.org/10.1007/978-3-031-07082-2_17 (cit. on p. 3).

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. "Quantum security proofs using semi-classical oracles". In: *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer. 2019, pp. 269–295. DOI: https://doi.org/10.1007/978-3-030-26951-7_10 (cit. on p. 10).

[ALMO08] Andris Ambainis, Debbie Leung, Laura Mancinska, and Maris Ozols. "Quantum random access codes with shared randomness". In: *arXiv preprint arXiv:0810.2937* (2008). DOI: https://doi.org/10.48550/arXiv.0810.2937 (cit. on p. 4).

[Amb02] Andris Ambainis. "Quantum lower bounds by quantum arguments". In: *Journal of Computer and System Sciences* 64.4 (2002), pp. 750–767. DOI: https://doi.org/10.1145/335305.335394 (cit. on pp. 1, 2).

[ANTV99] Andris Ambainis, Ashwin Nayak, Ammon Ta-Shma, and Umesh Vazirani. "Dense quantum coding and a lower bound for 1-way quantum automata". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 1999, pp. 376–383. DOI: https://doi.org/10.1145/301250.301347 (cit. on p. 4).

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. "Strengths and weaknesses of quantum computing". In: *SIAM journal on Computing* 26.5 (1997), pp. 1510–1523. DOI: https://doi.org/10.1137/S0097539796300933 (cit. on p. 1).

[BY23] Aleksandrs Belovs and Duyal Yolcu. "One-Way Ticket to Las Vegas and the Quantum Adversary". In: *arXiv preprint arXiv:2301.02003* (2023). DOI: https://doi.org/10.48550/arXiv.2301.02003 (cit. on p. 2).

[CGH+18] Jan Czajkowski, Leon Groot Bruinderink, Andreas Hülsing, Christian Schaffner, and Dominique Unruh. "Post-quantum security of the sponge construction". In: *International Conference on Post-Quantum Cryptography*. Springer. 2018, pp. 185–204. DOI: https://doi.org/10.1007/978-3-319-79063-3_9 (cit. on p. 22).

[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)*. IEEE. 2020, pp. 673–684. DOI: 10.1109/FOCS46700.2020.00068 (cit. on pp. 2, 3, 7).

[CLQ19] Kai-Min Chung, Tai-Ning Liao, and Luowen Qian. "Lower bounds for function inversion with quantum advice". In: *arXiv preprint arXiv:1911.09176* (2019). DOI: https://doi.org/10.48550/arXiv.1911.09176 (cit. on pp. 2–4, 13).

[CMSZ21] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. “Quantum lazy sampling and game-playing proofs for quantum indifferentiability”. In: *arXiv preprint arXiv:1904.11477* (2021). DOI: https://doi.org/10.48550/arXiv.1904.11477 (cit. on p. 22).

[CX21] Shujiao Cao and Rui Xue. “Being a permutation is also orthogonal to one-wayness in quantum world: Impossibilities of quantum one-way permutations from one-wayness primitives”. In: *Theoretical Computer Science* 855 (2021), pp. 16–42. DOI: https://doi.org/10.1016/j.tcs.2020.11.013 (cit. on pp. 2, 3).

[DH08] Catalin Dohotaru and Peter Hoyer. “Exact quantum lower bound for Grover’s problem”. In: *arXiv preprint arXiv:0810.3647* (2008). DOI: https://doi.org/10.26421/QIC9.5-6-12 (cit. on p. 17).

[DKRS23] Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. “Quantum time/memory/data tradeoff attacks”. In: *Designs, Codes and Cryptography* (2023), pp. 1–19. DOI: https://doi.org/10.1007/s10623-023-01300-x (cit. on p. 2).

[Dwo15] Morris J Dworkin. “SHA-3 standard: Permutation-based hash and extendable-output functions”. In: *Federal Inf. Process. Stds. (NIST FIPS)* (2015). DOI: https://doi.org/10.6028/NIST.FIPS.202 (cit. on p. 22).

[FK15] Bill Fefferman and Shelby Kimmel. “Quantum vs classical proofs and subset verification”. In: *arXiv preprint arXiv:1510.06750* (2015). DOI: https://doi.org/10.48550/arXiv.1510.06750 (cit. on p. 2).

[GJMG11] Bertoni Guido, Daemen Joan, P Michaël, and VA Gilles. *Cryptographic sponge functions*. 2011. URL: https://keccak.team/files/CSF-0.1.pdf (cit. on pp. 1, 22).

[Gro96] Lov K Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219. DOI: https://doi.org/10.1145/237814.237866 (cit. on pp. 1, 17, 19).

[HXY19] Minki Hhan, Keita Xagawa, and Takashi Yamakawa. “Quantum random oracle model with auxiliary input”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2019, pp. 584–614. DOI: https://doi.org/10.1007/978-3-030-34578-5_21 (cit. on pp. 2, 3, 7, 13, 18).

[KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020. DOI: https://doi.org/10.1201/9781420010756 (cit. on pp. 1, 20).

[Liu23] Qipeng Liu. “Non-uniformity and Quantum Advice in the Quantum Random Oracle Model”. In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 117–143. DOI: https://doi.org/10.1007/978-3-031-30545-0_5 (cit. on pp. 2, 3).

[NABT14] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. “Quantum lower bound for inverting a permutation with advice”. In: *arXiv preprint arXiv:1408.3193* (2014). DOI: https://doi.org/10.48550/arXiv.1408.3193 (cit. on pp. 2, 3, 13).

[Nay10] Ashwin Nayak. “Inverting a permutation is as hard as unordered search”. In: *arXiv preprint arXiv:1007.2899* (2010). DOI: https://doi.org/10.48550/arXiv.1007.2899 (cit. on pp. 1, 2, 10, 11, 19).

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40. DOI: https://doi.org/10.1145/1568318.1568324 (cit. on p. 22).

[Ros21] Ansis Rosmanis. "Tight bounds for inverting permutations via compressed oracle arguments". In: *arXiv preprint arXiv:2103.08975* (2021). DOI: https://doi.org/10.48550/arXiv.2103.08975 (cit. on p. 2).

[Vaz98] Umesh Vazirani. "On the power of quantum computation". In: *Philosophical Transactions of the Royal Society of London A 365: 1759-1768* (1998). DOI: https://doi.org/10.1137/S0097539796298637 (cit. on p. 4).

[Wie83] Stephen Wiesner. "Conjugate coding". In: *ACM Sigact News* 15.1 (1983), pp. 78–88. DOI: https://doi.org/10.1145/1008908.1008920 (cit. on p. 4).

[Zal99] Christof Zalka. "Grover's quantum searching algorithm is optimal". In: *Physical Review A* 60.4 (1999), p. 2746. DOI: https://doi.org/10.1103/PhysRevA.60.2746 (cit. on p. 19).

[Zha16] Mark Zhandry. "A note on quantum-secure PRPs". In: *arXiv preprint arXiv:1611.05564* (2016). DOI: https://doi.org/10.48550/arXiv.1611.05564 (cit. on p. 20).

[Zha19] Mark Zhandry. "How to record quantum queries, and applications to quantum indifferentiability". In: *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*. Springer. 2019, pp. 239–268. DOI: https://doi.org/10.1007/978-3-030-26951-7_9 (cit. on pp. 3, 17, 19).

A Some basic probabilistic lemmas

In this section we collect a series of known probabilistic results, which we used in our main proofs.

We first record some basic lemmas about the behavior of certain types of random variables.

Lemma A.1 (Multiplicative Chernoff Bound). *Let $X_1, \ldots, X_n$ be independent random variables taking values in $\{0,1\}$. Let $X = \sum_{i\in[n]} X_i$ denote their sum and let $\mu = \mathbb{E}[X]$ denote its expected value. Then for any $\delta > 0$,*

$$\Pr[X < (1-\delta)\mu] \leq 2e^{-\delta^2\mu/2}.$$

Specifically, when X_i is a Bernoulli trial and X follows the binomial distribution with $\mu = np$ and $p > \frac{1}{2}$, we have $\Pr[X \leq n/2] \leq e^{-n(p-\frac{1}{2})^2/(2p)}$.

Lemma A.2 (Reverse Markov's inequality). *Let X be a random variable taking values in $[0,1]$. Let $\theta \in (0,1)$ be arbitrary. Then, it holds that*

$$\Pr[X \geq \theta] \geq \frac{\mathbb{E}[X] - \theta}{1-\theta}.$$

Proof. Fix $\theta \in (0,1)$. We first show that

$$(1-\theta) \cdot \mathbb{I}_{[X\geq\theta]} \geq X - \theta, \tag{11}$$

where $\mathbb{I}_{[X \geq \theta]}$ is the indicator function for the event that $X \geq \theta$. Suppose that $X \geq \theta$. Then, Eq. (11) amounts to $1 - \theta \geq X - \theta$, which is satisfied because $X \leq 1$. Now suppose that $X < \theta$. In this case Eq. (11) amounts to $0 \geq X - \theta$, which is satisfied whenever $X \geq 0$. Taking the expectation over Eq. (11) and noting that $\mathbb{E}[\mathbb{I}_{[X \geq \theta]}] = \Pr[X \geq \theta]$, we get

$$(1 - \theta) \cdot \Pr[X \geq \theta] \geq \mathbb{E}[X] - \theta.$$

This proves the claim. □

Lemma A.3 (Averaging argument)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be any finite sets and let $\Omega : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a predicate. Suppose that $\Pr_{x,y}[\Omega(x,y) = 1] \geq \epsilon$, for some $\epsilon \in [0,1]$, where x is chosen uniformly at random in $\mathcal{X}$. Let $\theta \in (0,1)$. Then, there exists a subset $\mathcal{X}_\theta \subseteq \mathcal{X}$ of size $|\mathcal{X}_\theta| \geq (1-\theta) \cdot \epsilon |\mathcal{X}|$ such that*

$$\Pr_y[\Omega(x,y) = 1] \geq \theta \cdot \epsilon, \quad \forall x \in \mathcal{X}_\theta.$$

Proof. Define $p_x = \Pr_y[\Omega(x,y) = 1]$, for $x \in \mathcal{X}$. Then, for $\epsilon \in [0,1]$, we have

$$\mathbb{E}_x[p_x] = \Pr_{x,y}[\Omega(x,y) = 1] = |\mathcal{X}|^{-1} \sum_{x \in \mathcal{X}} \Pr_y[\Omega(x,y) = 1] \geq \epsilon.$$

Fix $\theta \in (0,1)$. Because the weighted average above is at least ϵ, there must exist a subset $\mathcal{X}_\theta$ such that

$$p_x = \Pr_y[\Omega(x,y) = 1] \geq \theta \cdot \epsilon, \quad \forall x \in \mathcal{X}_\theta.$$

Recall that x is chosen uniformly at random in $\mathcal{X}$. Using the reverse Markov's inequality, it follows that

$$\frac{|\mathcal{X}_\theta|}{|\mathcal{X}|} = \Pr[p_x \geq \theta \cdot \epsilon] \geq \frac{\mathbb{E}[p_x] - \theta \cdot \epsilon}{1 - \theta \cdot \epsilon} \geq \frac{\epsilon \cdot (1-\theta)}{1 - \theta \cdot \epsilon} > \epsilon \cdot (1 - \theta).$$

In other words, the subset $\mathcal{X}_\theta \subseteq \mathcal{X}$ is of size at least $|\mathcal{X}_\theta| \geq (1-\theta) \cdot \epsilon |\mathcal{X}|$. □

B Amplification proofs

B.1 Quantum oracle construction in Protocol 1

In Protocol 1 step 2(c), $\mathsf{S}[\ell]_1$, with quantum oracle access to $\mathcal{O}_\pi, \mathcal{O}_{\pi^{-1}_{\perp y}}$, needs to grant S_1 quantum oracle access to $(\sigma_{1,i} \circ \pi \circ \sigma_{2,i})_{\perp \sigma_{1,i}(y)}$, which is a simplified notation of $\mathcal{O}_{\sigma_{1,i} \circ \pi \circ \sigma_{2,i}}$ and $\mathcal{O}_{(\sigma_{1,i} \circ \pi \circ \sigma_{2,i})^{-1}_{\perp \sigma_{1,i}(y)}}$. Here we give detailed constructions of these two oracles:

- Whenever the algorithm S_1 queries the oracle $\mathcal{O}_{\sigma_{1,i} \circ \pi \circ \sigma_{2,i}}$ on $|w\rangle_1 |z\rangle_2$, $\mathsf{S}[\ell]_1$ performs the following reversible operations

$$
\begin{aligned}
&|w\rangle_1 |z\rangle_2 \\
\xrightarrow{\text{add aux registers}} &|w\rangle_1 |z\rangle_2 |0\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma_{2,i},1,\mathsf{aux2}}} &|w\rangle_1 |z\rangle_2 |0\rangle_{\mathsf{aux1}} |\sigma_{2,i}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\pi,\mathsf{aux2},\mathsf{aux1}}} &|w\rangle_1 |z\rangle_2 |\pi \circ \sigma_{2,i}(w)\rangle_{\mathsf{aux1}} |\sigma_{2,i}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma_{1,i},\mathsf{aux1},2}} &|w\rangle_1 |z \oplus \sigma_{1,i} \circ \pi \circ \sigma_{2,i}(w)\rangle_2 |\pi \circ \sigma_{2,i}(w)\rangle_{\mathsf{aux1}} |\sigma_{2,i}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\pi,\mathsf{aux2},\mathsf{aux1}}} &|w\rangle_1 |z \oplus \sigma_{1,i} \circ \pi \circ \sigma_{2,i}(w)\rangle_2 |0\rangle_{\mathsf{aux1}} |\sigma_{2,i}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma_{2,i},1,\mathsf{aux2}}} &|w\rangle_1 |z \oplus \sigma_{1,i} \circ \pi \circ \sigma_{2,i}(w)\rangle_2 |0\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\text{drop aux}} &|w\rangle_1 |z \oplus \sigma_{1,i} \circ \pi \circ \sigma_{2,i}(w)\rangle_2 .
\end{aligned}
$$

Then, $\mathsf{S}[\ell]_1$ sends the final state back to S_1.

- Whenever S_1 queries the oracle $\mathcal{O}_{(\sigma_{1,i}\circ\pi\circ\sigma_{2,i})^{-1}_{\perp\sigma_{1,i}(y)}}$ on $|w\rangle_1 |z\rangle_2$, the algorithm $\mathsf{S}[\ell]_1$ performs the following reversible operations:

$$
\begin{aligned}
&|w\rangle_1 |z\rangle_2 \\
\xrightarrow{\text{add aux register}} &|w\rangle_1 |z\rangle_2 |0\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma^{-1}_{1,i,*},1,\mathsf{aux1}}} &|w\rangle_1 |z\rangle_2 |\sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\pi^{-1}_{\perp y},\mathsf{aux1},\mathsf{aux2}}} &|w\rangle_1 |z\rangle_2 |\sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux1}} |\pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma^{-1}_{2,i,*},2,\mathsf{aux2}}} &|w\rangle_1 |z \oplus \sigma^{-1}_{2,i,*} \circ \pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_2 |\sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux1}} |\pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\pi^{-1}_{\perp y},\mathsf{aux1},\mathsf{aux2}}} &|w\rangle_1 |z \oplus \sigma^{-1}_{2,i,*} \circ \pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_2 |\sigma^{-1}_{1,i,*}(w)\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\mathcal{O}_{\sigma^{-1}_{1,i,*},1,\mathsf{aux1}}} &|w\rangle_1 |z \oplus \sigma^{-1}_{2,i,*} \circ \pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_2 |0\rangle_{\mathsf{aux1}} |0\rangle_{\mathsf{aux2}} \\
\xrightarrow{\text{drop aux}} &|w\rangle_1 |z \oplus \sigma^{-1}_{2,i,*} \circ \pi^{-1}_{\perp y} \circ \sigma^{-1}_{1,i,*}(w)\rangle_2 .
\end{aligned}
$$

where $\sigma^{-1}_{\cdot,i,*} : [N] \times \{0,1\} \to [N] \times \{0,1\}$ is given below

$$\sigma^{-1}_{\cdot,i,*}(w\|b) := \sigma^{-1}_{\cdot,i}(w)\|b.$$

Then, $\mathsf{S}[\ell]_1$ sends the final state back to S_1.

B.2 Another amplification lemma proof

Lemma 4.2. *Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an ϵ-SPI with space and time complexity given by $S(\mathsf{S})$ and $T(\mathsf{S})$, respectively, for some $\epsilon > 0$. Then, we can construct an SPI $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ with space and time* *complexities*

$$S(\mathsf{S}[\ell]) = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil \cdot S(\mathsf{S}) \quad \textit{and} \quad T(\mathsf{S}[\ell]) = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil \cdot (T(\mathsf{S}) + 1)$$

such that

$$\Pr_{\pi,y}\left[\Pr_r\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \frac{1}{5}.$$

Proof. Let $\ell = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil$. Using Lemma 4.1, we can construct an ℓ-time repetition of S (η)-SPI, denoted by $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$, with $\eta = 1-(1-\epsilon)^\ell$ and space and time complexities $S(\mathsf{S}[\ell]) = \ell \cdot S(\mathsf{S})$ and $T(\mathsf{S}[\ell]) = \ell \cdot (T(\mathsf{S}) + 1)$. In other words,

$$\Pr_{\pi,y,r}\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \geq 1 - (1-\epsilon)^\ell \geq \frac{9}{10}.$$

Let $\mathcal{S}_N$ denote the set of permutations over $[N]$. From Lemma A.3 it follows that there exists $\theta = 7/9$ and a subset $\mathcal{X}_\theta \subseteq \mathcal{S}_N \times [N]$ of size at least

$$|\mathcal{X}_\theta| \geq (1-\theta) \cdot \frac{9}{10} \cdot \left|\mathcal{S}_N \times [N]\right| = \frac{1}{5} \cdot \left|\mathcal{S}_N \times [N]\right|.$$

such that, for every $(\pi, y) \in \mathcal{X}_\theta$, we have

$$\Pr_r\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \geq \theta \cdot \frac{9}{10} > \frac{2}{3}.$$

Because $|\mathcal{X}_\theta| \cdot |\mathcal{S}_N \times [N]|^{-1} \geq \frac{1}{5}$, it follows that

$$\Pr_{\pi,y}\left[\Pr_r\left[\pi^{-1}(y) \leftarrow \mathsf{S}[\ell]_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathsf{S}[\ell]_0(\pi, r)\right] \geq \frac{2}{3}\right] \geq \frac{1}{5}.$$

This proves the claim. □

B.3 Decision amplification proof

Same as the search amplification, we amplify the success probability of a δ-DPI through ℓ-time repetition defined in Protocol 3.

Protocol 3 (ℓ-time repetition of δ-DPI)**.** *Given a δ-DPI $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$, the construction of an "ℓ-time serial repetition of D" $\mathsf{D}[\ell] = (\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ is as follows:*

1. *(Advice Preparation) the algorithm $\mathsf{D}[\ell]_0$ proceeds as follows:*

   (a) $\mathsf{D}[\ell]_0$ receives as input a random permutation $\pi : [N] \to [N]$ and randomness $r \leftarrow \{0,1\}^*$ and parses the string r into 2ℓ substrings, i.e. $r = r_0\|...\|r_{\ell-1}\|r_\ell\|...\|r_{2\ell-1}$ (the length is clear in context).

   (b) $\mathsf{D}[\ell]_0$ uses $r_0, ..., r_{\ell-1}$ to generate ℓ permutation pairs $\{\sigma_{1,i}, \sigma_{2,i}\}_{i=0}^{\ell-1}$ in $\mathcal{S}_N$, where $\sigma_{1,i}$ is a random permutation, $\sigma_{2,i}$ has the following form

   $$\sigma_{2,i}(x_1, ..., x_n) = (x_1 \oplus r_i^*, x_2, ..., x_n), \tag{12}$$

   where r_i^* is some random bit generated from r_i for all $i \in [0, \ell-1]$. Then runs $\mathsf{D}_0(\sigma_{1,i} \circ$

$\pi \circ \sigma_{2,i}, r_{i+\ell})$ *to get a quantum state* $\rho_i := \rho_{\sigma_{1,i}\circ\pi\circ\sigma_{2,i},r_{i+\ell}}$ *for all* $i \in [0, \ell-1]$*. Finally,* $\mathsf{D}[\ell]_0$ *outputs a quantum state* $\bigotimes_{i=0}^{\ell-1} \rho_i$*.*

2. *(Oracle Algorithm)* $\mathsf{D}[\ell]_1^{\pi_{\perp y}}$ *is an oracle algorithm that proceeds as follows:*

   (a) $\mathsf{D}[\ell]_1$ *receives* $\bigotimes_{i=0}^{\ell-1} \rho_i$*, randomness* r *and an image* $y \in [N]$ *as input.*

   (b) $\mathsf{D}[\ell]_1$ *parses* $r = r_0\|...\|r_{\ell-1}\|r_\ell\|...\|r_{2\ell-1}$ *and uses the coins* $r_0\|...\|r_{\ell-1}$ *to generate* ℓ *different permutation pairs* $\{\sigma_{1,i}, \sigma_{2,i}\}_{i=0}^{\ell-1}$ *in* $\mathcal{S}_N$ *as shown above.*

   (c) $\mathsf{D}[\ell]_1$ *then runs the following routine for all* $i \in [0, \ell-1]$*:*

      i. *Run* D_1 *with oracle access to* $(\sigma_{1,i} \circ \pi \circ \sigma_{2,i})_{\perp\sigma_{1,i}(y)}$*, which implements the permutation* $\sigma_{1,i} \circ \pi \circ \sigma_{2,i}$ *and its inverse (but* $\perp$ *at* $\sigma_{1,i}(y)$*).*

      ii. *Get back* $b_i \leftarrow \mathsf{D}_1^{(\sigma_{1,i}\circ\pi\circ\sigma_{2,i})_{\perp\sigma_{1,i}(y)}}(\rho_i, \sigma_{1,i}(y), r_{i+\ell})$*.*

   (d) $\mathsf{D}[\ell]_1$ *pads* b_i *with all zero string of size* $n-1$ *and computes* $b_i^* = \sigma_{2,i}(b_i\|0^{n-1})|_0$ *for all* $i \in [0, \ell-1]$*, then outputs* b^* *which is the majority vote of* $\{b_0^*, \ldots, b_{\ell-1}^*\}$*.*

Lemma 4.3. *Let* $(\mathsf{D}_0, \mathsf{D}_1)$ *be a* δ*-DPI, where* D_0 *outputs an* S*-qubit state and* D_1 *makes* T *queries. Then, we can construct an* ℓ*-time repetition of* D*, denoted by* $\mathsf{D}[\ell] = (\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$*, which is an* η*-DPI for* $\eta \geq \frac{1}{2} - \exp\Big(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\Big)$*, and has space and time complexities given by*

$$S(\mathsf{D}[\ell]) = \ell \cdot S(\mathsf{D}) \quad \text{and} \quad T(\mathsf{D}[\ell]) = \ell \cdot T(\mathsf{D}).$$

Proof. Let $(\mathsf{D}_0, \mathsf{D}_1)$ be a δ-DPI for some $\delta > 0$, where D_0 outputs an S-qubit state and D_1 makes T queries. Similarly as in Lemma 4.1, we consider the execution of the ℓ-time repetition of δ-DPI, denoted by DPI $\mathsf{D}[\ell]$, which we define in Protocol 3. For each iteration $i \in [0, \ell-1]$, we have

$$\begin{aligned}
&\Pr\big[b_i = \pi^{-1}(y)|_0\big] \\
&= \Pr\left[(\bar{\pi})^{-1}(\sigma_{1,i}(y))|_0 \leftarrow \mathsf{D}_1^{(\bar{\pi})_{\perp\sigma_{1,i}(y)}}\big(\rho_i, \sigma_{1,i}(y), r_{i+\ell}\big) : \rho_i \leftarrow \mathsf{D}_0(\bar{\pi}, r_{i+\ell})\right] \\
&\equiv \Pr\left[((\sigma_{2,i})^{-1} \circ \pi^{-1}(y))|_0 \leftarrow \mathsf{D}_1^{\pi_{\perp y}}\big(\rho_{\pi\circ\sigma_{2,i},r_{i+\ell}}, y, r_{i+\ell}\big) : \rho_{\pi\circ\sigma_{2,i},r_{i+\ell}} \leftarrow \mathsf{D}_0(\pi \circ \sigma_{2,i}, r_{i+\ell})\right] \\
&\geq \ \frac{1}{2} + \delta,
\end{aligned}$$

where $\bar{\pi} = \sigma_{1,i} \circ \pi \circ \sigma_{2,i}$. The probability is taken over $\pi \leftarrow \mathcal{S}_N$, $r \leftarrow \{0,1\}^*$ (which is used to sample permutations σ_i) and $x \leftarrow [N]$, along with all internal measurements of D.

Recall that $b_i \leftarrow \mathsf{D}_1^{(\sigma_{1,i}\circ\pi\circ\sigma_{2,i})_{\perp\sigma_{1,i}(y)}}(\rho_i, \sigma_{1,i}(y), r_{i+\ell})$, for $i \in [\ell]$. Let X_i be the indicator variable for the event that $b_i = (\pi \circ \sigma_{2,i})^{-1}(y)|_0$. Similar to the search case, we argue that all X_i are mutually independent. For any $i \in [0, \ell-1]$ and any subset $K \subset [0, l-1]$ where $i \notin K$, let

$$\begin{aligned}
\text{Event } A &= \{X_i = 0\} \\
&= \{b_i \neq \big((\sigma_{2,i})^{-1} \circ \pi^{-1}(y)\big)|_0\} \\
&= \{b_i||0^{n-1}|_0 \neq \big((\sigma_{2,i})^{-1} \circ \pi^{-1}(y)\big)|_0\} \\
&= \{\big(\sigma_{2,i} \circ (b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0\},
\end{aligned}$$

Note that the last equality holds because of Equation 12. We then define another event

$$\begin{aligned}\textsf{Event } B &= \bigcap_{j\in K}\{X_j = 0\}\\ &= \bigcap_{j\in K}\{\big(\sigma_{2,j}\circ(b_j||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0\}\end{aligned}$$

Given that B happens, we have $\{b_j\}_{j\in K}$ such that for all $j\in K$, $\big(\sigma_{2,j}\circ(b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$. We now consider the probability that A happens. In Equation 12, since all r_i^* are independently randomly generated, the value of $\big(\sigma_{2,i}\circ(b_i||0^{n-1})\big)|_0$ is independent of all other values of $\big(\sigma_{2,j}\circ(b_i||0^{n-1})\big)|_0$. Therefore, the event that $\big(\sigma_{2,i}\circ(b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$ is not correlated with all other $\big(\sigma_{2,j}\circ(b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$, i.e., $\Pr[A|B] = \Pr[A]$. This is true for any i and K. Same as the search case, in each trial, the base inverter is solving a completely independent permutation inversion problem, thus we conclude that all ℓ trails are mutually independent.

Let $X = \sum_{i=0}^{\ell-1} X_i$, we have that $\mathbb{E}[X] \geq \ell\cdot(\frac{1}{2}+\delta)$ by the linearity of expectation. Note that $\mathsf{D}[\ell]$ succeeds in $\mathsf{DecisionInvert}$ if and only if $\mathsf{D}[\ell]_1$ can output $b^* = \pi^{-1}(y)|_0$, i.e. $X > \frac{\ell}{2}$ in which case more than half of the elements in $\{b_0, ..., b_{\ell-1}\}$ are equal to $\pi^{-1}(y)|_0$. By the multiplicative Chernoff bound in Lemma A.1, the probability that $\mathsf{DecisionInvert}$ fails is at most

$$\Pr\left[X < \frac{\ell}{2}\right] \leq \exp\left(-\frac{\delta^2}{(1+2\delta)}\cdot\ell\right).$$

Note that the resource requirements needed for the amplification procedure amount to space and time complexities ℓS and ℓT, respectively, similar as in Lemma 4.1. □

C Quantum oracle constructions in Theorem 5.4

In Theorem 5.4, $\mathcal{B}$, with quantum oracle access to f, needs to grant $\mathcal{A}$ quantum oracle access to $h_{f,\pi,t,\mu}$ and $h^{-1*}_{f,\pi,t,\mu}$. Here we give detailed constructions of $\mathcal{O}_{h_{f,\pi,t,\mu}}$ and $\mathcal{O}_{h^{-1*}_{f,\pi,t,\mu}}$. Note that π is sampled by $\mathcal{B}$ and so it is easy for it to construct quantum oracles $\mathcal{O}_\pi$ and $\mathcal{O}_{\pi^{-1}_{\perp t}}$. Since $h^{-1*}_{f,\pi,t,\mu} = \pi^{-1}_{\perp t}$, the partial inverse oracle $O_{h^{-1*}_{f,\pi,t,\mu}}$ can be simply simulated by $\mathcal{O}_{\pi^{-1}_{\perp t}}$. So we only need to show how to construct $\mathcal{O}_{h_{f,\pi,t,\mu}}$.

Let $x = x_0\ldots x_{n-1}$, where $n = \log N$. When $\pi\in\pi_{t,0,\mu}$, the function becomes

$$\begin{aligned}h_{f,\pi,t,\mu}(x_0\ldots x_{n-1}) = &\,(x_0\cdot f(x_1...x_{n-m-1})\cdot \mathbb{1}(x_{n-m}...x_n = \mu))\cdot t\\ &+\overline{(x_0\cdot f(x_1...x_{n-m-1})\cdot \mathbb{1}(x_{n-m}...x_n = \mu))}\cdot\pi(x).\end{aligned}$$

Then define a function $g : [N]\to\{0,1\}$, such that $g(x) = x_0\cdot f(x_1...x_{n-m-1})\cdot\mathbb{1}(x_{n-m}...x_n = \mu)$. With access to $\mathcal{O}_f$, it is easy to construct $\mathcal{O}_g$ by applying $\mathcal{O}_f$ to the last $n-1$ bits followed by an AND gate.

Now when $\mathcal{A}$ queries the oracle $\mathcal{O}_{h_{f,\pi,t,\mu}}$ on $|x\rangle\,|y\rangle$, $\mathcal{B}$ performs the following reversible operations

$$
\begin{aligned}
&|x\rangle\,|y\rangle\\
\xrightarrow{\text{add aux registers}}\ &|x\rangle_1\,|y\rangle_2\,|0\rangle_3\,|0\rangle_4\,|0^n\rangle_5\,|0^n\rangle_6\\
\xrightarrow{\mathcal{O}_{g,1,3}X_4\mathcal{O}_{1,4}\mathcal{O}_{\pi,1,5}U_t}\ &|x\rangle\,|y\rangle\,|g(x)\rangle\,|\overline{g(x)}\rangle\,|\pi(x)\rangle\,|t\rangle\\
\xrightarrow{\text{CCNOT}_{3,6,2}}\ &|x\rangle\,|y\oplus(g(x)\cdot t)\rangle\,|g(x)\rangle\,|\overline{g(x)}\rangle\,|\pi(x)\rangle\,|t\rangle\\
\xrightarrow{\text{CCNOT}_{4,5,2}}\ &|x\rangle\,|y\oplus(g(x)\cdot t)\oplus(\overline{g(x)}\cdot\pi(x))\rangle\,|g(x)\rangle\,|\overline{g(x)}\rangle\,|\pi(x)\rangle\,|t\rangle\\
\xrightarrow{\mathcal{O}_{g,1,3}X_4\mathcal{O}_{1,4}\mathcal{O}_{\pi,1,5}U_t}\ &|x\rangle\,|y\oplus(g(x)\cdot t)\oplus(\overline{g(x)}\cdot\pi(x))\rangle\,|0\rangle\,|0\rangle\,|0^n\rangle\,|0^n\rangle\\
\xrightarrow{\text{drop aux}}\ &|x\rangle\,|y\oplus(g(x)\cdot t)\oplus(\overline{g(x)}\cdot\pi(x))\rangle
\end{aligned}
$$

It is easy to see that $y\oplus(g(x)\cdot t)\oplus(\overline{g(x)}\cdot\pi(x)) = y\oplus h_{f,\pi,t,\mu}(x)$. Therefore, to respond to one query to $O_{h_{f,\pi,t,\mu}}$, $\mathcal{B}$ needs to query $\mathcal{O}_f$ *twice* (once for computing and once for eliminating). The same thing can be done when $\pi\in\pi_{t,1,\mu}$.

D Quantum oracle constructions in Protocol 2

Here, we show how to implement the function $\bar{\pi}_{\perp y}$ by means of a (reversible) quantum oracle. This can be done by two separate oracles $\mathcal{O}_{\bar{\pi}}$ and $\mathcal{O}_{\bar{\pi}^{-1}_{\perp y}}$, where the corresponding functions are

$$\bar{\pi}(w)=\begin{cases} y & \text{if } w\in\mathcal{G}\\ \pi(w) & \text{if } w\notin\mathcal{G}\end{cases}$$

and

$$\bar{\pi}^{-1}_{\perp y}(w,b)=\begin{cases}\pi^{-1}(w)||0 & \text{if } w\notin\pi(\mathcal{G})\ \wedge\ b=0\\ 1||1 & \text{if } w\in\pi(\mathcal{G})\ \wedge\ b=1.\end{cases}$$

Let f be an indicator function on whether $w\in\mathcal{G}$. Given β as an input, the permutation π restricted to inputs outside of $\mathcal{G}$ is known (denoted as π'). Therefore given input y, with quantum oracle access to $\mathcal{O}_f$ and $\mathcal{O}_{\pi'}$, we can easily construct $\mathcal{O}_{\bar{\pi}}$ and $\mathcal{O}_{\bar{\pi}^{-1}_{\perp y}}$.

The following procedure gives a construction of $\mathcal{O}_{\bar{\pi}}$.

$$
\begin{aligned}
&|w\rangle\,|z\rangle\\
\xrightarrow{\text{add aux registers}}\ &|w\rangle_1\,|z\rangle_2\,|0\rangle_3\,|0\rangle_4\,|0^n\rangle_5\,|0^n\rangle_6\\
\xrightarrow{\mathcal{O}_{f,1,3}X_4\mathcal{O}_{1,4}\mathcal{O}_{\pi',1,5}U_y}\ &|w\rangle\,|z\rangle\,|f(w)\rangle\,|\overline{f(w)}\rangle\,|\pi'(w)\rangle\,|y\rangle\\
\xrightarrow{\text{CCNOT}_{3,6,2}}\ &|w\rangle\,|z\oplus(f(w)\cdot y)\rangle\,|f(x)\rangle\,|\overline{f(w)}\rangle\,|\pi'(w)\rangle\,|t\rangle\\
\xrightarrow{\text{CCNOT}_{4,5,2}}\ &|x\rangle\,|z\oplus(f(w)\cdot y)\oplus(\overline{f(w)}\cdot\pi'(w))\rangle\,|f(w)\rangle\,|\overline{f(w)}\rangle\,|\pi'(w)\rangle\,|y\rangle\\
\xrightarrow{\mathcal{O}_{f,1,3}X_4\mathcal{O}_{1,4}\mathcal{O}_{\pi',1,5}U_y}\ &|x\rangle\,|z\oplus(f(w)\cdot y)\oplus(\overline{f(w)}\cdot\pi'(w))\rangle\,|0\rangle\,|0\rangle\,|0^n\rangle\,|0^n\rangle\\
\xrightarrow{\text{drop aux}}\ &|x\rangle\,|z\oplus(f(w)\cdot y)\oplus(\overline{f(w)}\cdot\pi'(w))\rangle\\
\equiv\ &|x\rangle\,|z\oplus\bar{\pi}(w)\rangle
\end{aligned}
$$

The backward oracle $\mathcal{O}_{\bar{\pi}^{-1}_{\perp y}}$ would be constructed similarly.