

On the Two-sided Permutation Inversion Problem

Gorjan Alagic¹, Chen Bai², Alexander Poremba³, and Kaiyan Shi⁴

¹QuICS, University of Maryland, and NIST

²Dept. of Electrical and Computer Engineering, University of Maryland

³Computing and Mathematical Sciences, California Institute of Technology

⁴Dept. 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 $\pi : [N] \rightarrow [N]$ 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) [BDPA11].

We consider two options for this problem:

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 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 can also share an arbitrarily long uniformly random string; we refer to this string 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. (*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 succeeds with probability at least ϵ in the search inversion experiment, we say it is an ϵ -SPI. If a DPI succeeds with probability at least $1/2 + \delta$ in the decision inversion experiment, we say it is a δ -DPI. When we want to emphasize the number of queries T and the number of qubits S (in the advice state), we will also write, e.g., (S, T, ϵ) -SPI.

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.

1.2 Summary of results

Much is known about the standard (i.e., one-sided) inversion problem; we will review some of these results further below. The two-sided variant has received much less attention. Our results establish a series of basic facts about this variant. For the remainder of the paper, unless stated otherwise, we will refer to the two-sided permutation inversion problem as “the inversion problem” or simply “the problem.”

Amplification. We consider a simple form of amplification: the inversion algorithm \mathcal{A} is run ℓ times; once the ℓ executions are complete, the outputs x_i are tested to see if $\pi(x_i) = y$ (in the search case) or the majority bit is output (in the decision case.) To ensure that each execution behaves independently, the shared randomness is used to randomize the problem instance given to each execution of \mathcal{A} . The total advice state of the amplified algorithm then consists of the ℓ advice states generated by each execution of \mathcal{A} . We refer to the amplified algorithm as $\mathcal{A}[\ell]$. We show that this amplification boosts an (S, T, ϵ) -SPI to an $(\ell S, \ell(T + 1), 1 - (1 - \epsilon)^\ell)$ -SPI, and show a similar result for decision. This is formalized in [Lemma 4.1](#) and [Lemma 4.3](#).

Search-to-decision reduction. Clearly, the search version of any variant of the inversion problem is no easier than the corresponding decision version. We establish a simple reduction showing that search is in fact also not much harder than decision. Specifically, we show that an (S, T, δ) -DPI can be used to construct a $(n\ell S, n\ell T, 1 - ne^{-\ell\delta})$ -SPI (here and throughout, $n = \lceil \log N \rceil$.) This is formalized in [Theorem 5.1](#).

Lower bounds, search version. We establish a lower bound for the search version of the inversion problem with advice, showing that $ST^2 \geq \tilde{\Omega}(\epsilon^3 N)$ for any (S, T, ϵ) -SPI. While this bound

is not tight, we do establish a tighter bound of $ST^2 \geq \tilde{\Omega}(\epsilon N)$ for a restricted class of inverters (similarly to a result of [CLQ19].) These results are formalized in [Theorem 6.2](#) and [Theorem 6.1](#).

Lower bounds, decision version. For the decision version with advice, we combine two results above (search lower bound and search-to-decision reduction) to yield a (non-tight) bound of $ST^2 \geq \tilde{\Omega}(\delta^6 N)$ for any δ -DPI. In the case of no advice, we get a tight lower bound via a reduction from the unstructured search problem; this shows that $\tilde{\Omega}(\sqrt{N})$ queries are required. Our reduction is similar to that of Nayak [Nay10]. These results are formalized in [Corollary 6.3](#) and [Corollary 6.4](#).

Applications. We observe that the two-sided version of the permutation inversion problem can be viewed as the main task of an adversary in a natural cryptographic experiment. In this experiment, the adversary is tasked with decrypting an encryption of a random message, while having oracle access to both the encryption map and the decryption map. This is a standard security notion called OW-CCA (one-way security against chosen ciphertext attack). In our setting, we grant the attacker even more power: they can query quantumly, they can control the randomness of the encryption map, and they can deduce the randomness used to encrypt when applying the decryption map. We call this QCCRA (quantum chosen ciphertext with randomness-access attack).

We apply our lower bounds above to show that a natural encryption scheme constructed from random permutations is secure even against these powerful attacks. In the computational security setting, such a scheme can be instantiated efficiently using quantum-query-secure pseudorandom permutations [Zha16]. These results are formalized in [Theorem 7.2](#) and [Theorem 7.4](#).

Future work. The two-sided permutation inversion problem appears naturally in the context of sponge hashing [BDPA11] which is used by the international hash function standard SHA3 [Dwo15]. Previous work [CBH⁺17, 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.

1.3 Technical overview

Our main technical result is the lower bound for the search variant of the two-sided permutation inversion problem in [Section 6.1](#). At a high level, our proof uses a similar *compression argument* as in previous works on one-sided permutation inversion with advice [NABT14, CLQ19, HXY19]. We use information-theoretic lower bounds on the length of quantum random access codes [Wie83, ANTV98, ALMO08], which are a means of encoding classical bits in terms of (potentially fewer) qubits. In other words, we construct an encoder that compresses the truth table of a permutation by using the power of the search inverter, which then allows us to obtain the desired space-time lower bound $ST^2 = \tilde{\Omega}(\epsilon^3 N)$ in [Theorem 6.2](#). Along the way, we show how to amplify search inverters for the two-sided permutation inversion problem; this can be done via a careful *averaging argument* (which we prove in [Lemma 2.3](#)). Our approach allows us to obtain a simpler amplification analysis as compared to previous work [HXY19] which used *quantum rewinding* for the one-sided case.

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

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

To obtain a space-time trade-off $ST^2 \geq \tilde{\Omega}(\delta^6 N)$ for decision inverters that succeed with bias $\delta > 0$, we give a *search-to-decision* reduction in Theorem 5.1. Specifically, we show that a decision inverter can be used to solve the (search) permutation inversion problem by recovering one bit of the preimage at a time. Here, we invoke a self-reduction that *re-randomizes* the decision inverter in each execution while guaranteeing independence.

1.4 Related work

Previous works have considered the quantum-query *function* inversion problem [HXY19, CLQ19, CGLQ20, DKRS21, Liu22]. A number of recent papers gave lower bounds for the (one-sided) quantum-query permutation inversion problem, with and without advice [Amb02, Nay10, Ros21, NABT14, HXY19, CLQ19, FK15, BY23]. The highlights among these are summarized in Table 1. 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, for example by applying the amplification procedure described in Lemma 4.2.

To our knowledge, the two-way variant of the inversion problem has only been considered in two other works. First, [CX21] gives a lower bound for inverting random injective functions in the case of two-way access without advice. Their query complexity is $T > N^{1/5}$ with non-negligible success probability. Second, [BY23] briefly considers inverse access for the permutation inversion problem, but only in the trivial setting where a query on the challenge is allowed.

Another novelty of our work is that we give a lower bound for the average-case decision problem. While prior work by Chung et al. [CGLQ20] also considered the general decision game, their generic framework crucially relies on compressed oracles [Zha18] 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, [ABKM21, 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.

Acknowledgements

We thank Christian Majenz for useful discussions. AP is partially supported by AFOSR YIP (award number FA9550-16-1-0495), the Institute for Quantum Information and Matter (an NSF Physics Frontiers Center; NSF Grant PHY-1733907) and by a grant from the Simons Foundation

(828076, TV). Gorjan Alagic and Kaiyan Shi acknowledge support from the U.S. Army Research Office under Grant Number W911NF-20-1-0015. GA and Chen Bai acknowledge support from the U.S. Department of Energy under Award Number DE-SC0020312. GA acknowledges support from the AFOSR under Award Number FA9550-20-1-0108, and from the NSF under Award Number CNS-2154705.

2 Technical preliminaries

In this section we collect a series of known technical results, which we will need for our main proofs.

2.1 Some basic probabilistic lemmas

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

Lemma 2.1 (Multiplicative Chernoff Bound). *Let X_1, \dots, 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 the expected value. Then for any $\delta > 0$,*

$$\Pr[X < (1 - \delta)\mu] \leq \exp\left(\frac{e^{-\delta}}{(1 - \delta)^{1-\delta}}\right)^\mu.$$

Specifically, for binomial distribution with $\mu = np$ and $p > \frac{1}{2}$, we have

$$\Pr[X \leq n/2] \leq e^{-n(p-\frac{1}{2})^2/(2p)}$$

and correspondingly,

$$\Pr\left[X > \frac{n}{2}\right] \geq 1 - e^{-n(p-\frac{1}{2})^2/(2p)}.$$

Lemma 2.2 (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

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

Suppose that $X \geq \theta$. Then, Eq. (1) amounts to $1 - \theta \geq X - \theta$, which is satisfied because $X \leq 1$. Now suppose that $X < \theta$. In this case Eq. (1) amounts to $0 \geq X - \theta$, which is satisfied whenever $X \geq 0$. Taking the expectation over Eq. (1) 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 2.3 (Averaging argument). *Let \mathcal{X} and \mathcal{Y} be any finite sets and let $\Omega : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$ 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}_θ 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 in [Lemma 2.2](#), 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 $|\mathcal{X}_\theta| \geq (1 - \theta) \cdot \epsilon |\mathcal{X}|$. □

2.2 Swapping Lemma

The following lemma controls the ability of a query algorithm to distinguish two oracles, in terms of a concept of “total query magnitude” to locations at which the oracles take differing values.

Definition 2.4 (Query magnitude). *Let $|\psi\rangle = \sum_{x \in \mathcal{X}} \alpha_x |x\rangle$ be a state and let $\mathcal{S} \subseteq \mathcal{X}$ be a subset, for some finite set \mathcal{X} . Then, the query magnitude with respect to \mathcal{S} is given by*

$$q_{\mathcal{S}}(|\psi\rangle) = \sum_{x \in \mathcal{S}} |\alpha_x|^2.$$

Definition 2.5 (Total query magnitude). *Let \mathcal{A}^f be a quantum algorithm with quantum oracle access to a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, 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*

$$q(\mathcal{A}^f, \mathcal{S}) := \sum_{t=0}^{T-1} q_{\mathcal{S}}(|\psi_t\rangle) = \sum_{t=0}^{T-1} \|\Pi_{\mathcal{S}} |\psi_t\rangle\|^2,$$

where we represent \mathcal{A} as a sequence $|\psi_0\rangle, \dots, |\psi_{T-1}\rangle$ of intermediate states, where $\Pi_{\mathcal{S}}$ is a projector onto a query register of \mathcal{A} , and where $|\psi_t\rangle$ represents the state of \mathcal{A} just before the $t + 1^{\text{st}}$ query.

We use the following elementary properties of the total query magnitude:

Lemma 2.6. *Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a function for some finite sets \mathcal{X} and \mathcal{Y} , and let \mathcal{A}^f be a quantum algorithm with quantum oracle access to f . Then,*

- For any subset $\mathcal{S} \subseteq \mathcal{X}$, it holds

$$q(\mathcal{A}^f, \mathcal{S}) \leq T,$$

where T be an upper bound on the number of queries made by \mathcal{A} .

- For any disjoint subsets $\mathcal{S}_0, \mathcal{S}_1 \subseteq \mathcal{X}$ it holds

$$q(\mathcal{A}^f, \mathcal{S}_0 \cup \mathcal{S}_1) = q(\mathcal{A}^f, \mathcal{S}_0) + q(\mathcal{A}^f, \mathcal{S}_1).$$

- For any subsets $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \mathcal{X}$ it holds that

$$q(\mathcal{A}^f, \mathcal{S}_0) \leq q(\mathcal{A}^f, \mathcal{S}_1).$$

Lemma 2.7 (Swapping Lemma, [Vaz98]). Let $f, g : \mathcal{X} \rightarrow \mathcal{Y}$ be functions such that $f(x) = g(x)$ 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, it holds that

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

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} .

2.3 Lower bounds for quantum random access codes

Quantum random access codes [Wie83, ANTV98, ALMO08] are a means of encoding classical bits into (potentially fewer) qubits. We use the following variant from [CLQ19].

Definition 2.8 (Quantum random access codes with variable length). Let N be an integer and let $\mathcal{F}_N = \{f : [N] \rightarrow \mathcal{X}_N\}$ 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 :

- $\text{Enc}(f; R)$: 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).
- $\text{Dec}(\rho, x; R)$: 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

$$L := \mathbb{E}_f[\ell(f)]$$

be the average length of the encoding over the uniform distribution on $f \in \mathcal{F}_N$, and let

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

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 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 2.9 ([CLQ19], Corollary 1). For any QRAC-VL for permutations \mathcal{S}_N with decoding advantage $\delta = 1 - k/N$ and for any $k = \Omega(1/N)$, we have

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

3 The permutation inversion problem

We begin by formalizing the search version of the problem of inverting a permutation. We let $[N] = \{1, \dots, N\}$; typically we choose $N = 2^n$ for some positive integer n . For $f : \mathcal{X} \rightarrow \mathcal{Y}$ a function from a set \mathcal{X} to an additive group \mathcal{Y} , the quantum oracle \mathcal{O}_f is the unitary operator

$$\mathcal{O}_f : |x\rangle |y\rangle \rightarrow |x\rangle |y \oplus f(x)\rangle.$$

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 $N \in \mathbb{N}$. A search-version permutation inverter (SPI) is a pair $\mathbf{S} = (\mathbf{S}_0, \mathbf{S}_1)$ of quantum algorithms, where*

- \mathbf{S}_0 is an algorithm which receives as input a truth table for a permutation over $[N]$ and a random string r , and outputs a quantum state;
- \mathbf{S}_1 is an oracle algorithm which receives a quantum state, an image $y \in [N]$, and a random string r , and outputs $x \in [N]$.

We will consider the execution of a SPI \mathbf{S} in the following experiment, which we call `SearchInvert \mathbf{S}` .

1. (*sample coins*) a uniformly random permutation $\pi : [N] \rightarrow [N]$ and a uniformly random string $r \leftarrow \{0, 1\}^*$ are sampled;
2. (*prepare advice*) \mathbf{S}_0 is run, producing a quantum state $\rho_{\pi, r} \leftarrow \mathbf{S}_0(\pi, r)$;
3. (*sample instance*) a uniformly random image $y \in [N]$ is generated;
4. (*invert*) \mathbf{S}_1 is run with the two oracles below, and produces a candidate preimage x^* .

$$\mathcal{O}_\pi(|w\rangle |z\rangle) = |w\rangle |z \oplus \pi(w)\rangle \quad \mathcal{O}_{\pi_{\perp y}^{-1}}(|w\rangle |z\rangle) = |w\rangle |z \oplus \pi_{\perp y}^{-1}(w)\rangle, \quad (2)$$

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

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

To keep notation simple, we write this process as $x^* \leftarrow \mathbf{S}_1^{\pi_{\perp y}}(\rho_{\pi, r}, y, r)$. We will use $\pi_{\perp y}$ to denote simultaneous access to the two oracles in (2) throughout the paper.

5. (*check*) If $\pi(x^*) = y$, output 1; otherwise output 0.

Note that the two oracles allow for the evaluation of the permutation π in both the forward and inverse direction. To disallow trivial solutions, the oracle outputs a fixed “reject” element $1||1 \in [N] \times \{0, 1\}$ if queried on y in the inverse direction. If the probability that \mathbf{S} successfully inverts (i.e., that the experiment outputs 1) is at least ϵ , we say that \mathbf{S} is an ϵ -SPI.

Definition 3.2. *An ϵ -SPI is a search-version permutation inverter $\mathbf{S} = (\mathbf{S}_0, \mathbf{S}_1)$ satisfying*

$$\Pr \left[\pi^{-1}(y) \leftarrow \mathbf{S}_1^{\pi_{\perp y}}(\rho, y, r) : \rho \leftarrow \mathbf{S}_0(\pi, r) \right] \geq \epsilon,$$

where the probability is taken over $\pi \leftarrow \mathcal{S}_N$, $r \leftarrow \{0, 1\}^*$ and $y \leftarrow [N]$, along with all internal randomness and measurements of \mathbf{S} .

We will measure the computational resources required by a SPI $S = (S_0, S_1)$ in terms of only two quantities. The first is an upper bound on the number of qubits of the state produced by S_0 , denoted by $S(S)$ (or simply S , when the context is clear.) The second is an upper bound on the number of oracle queries made by S_1 , denoted by $T(S)$ (or simply T .) We emphasize that the running time of S 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. To further simplify things, we will assume without loss of generality that S_0 outputs *exactly* S qubits and S_1 makes *exactly* T queries whenever S is run in the experiment described above. We denote ϵ -SPI with S and T as (S, T, ϵ) -SPI, especially we have $(0, T, \epsilon)$ -SPI when there is no advice ($S = 0$).

Relationship to previous notion. In [CX21, CLQ19, NABT14], the success of a SPI is measured in an alternative way. First, \mathcal{A} is said to "invert y for π " if \mathcal{A} succeeds in the inversion experiment for the pair (π, y) with probability (over all remaining randomness and all measurements) at least $2/3$. Second, \mathcal{A} is said to "invert a δ -fraction of inputs" if $\Pr_{\pi, y}[\mathcal{A} \text{ inverts } y \text{ for } \pi] \geq \delta$. This type of inverter is clearly captured by Definition 3.1: it is an ϵ -SPI with $\epsilon = 2\delta/3$. However, there are inverters of interest which are captured by Definition 3.1, but not by the previous definition. For example, in a cryptographic context, one would definitely be concerned about adversaries which can invert every (π, y) with probability exactly $1/n$. Such an adversary is clearly a $(1/n)$ -SPI, but is not a valid inverter under the previous definition for any value of δ . Other works also consider the general average-case captured by Definition 3.1 (e.g., [CGLQ20, Liu22, HXY19]) but without two-way oracle access.

Decision version. The decision version of the permutation inversion problem is defined similarly to the search version above, with the modifications listed here:

- A decision-version permutation inverter (DPI) is denoted $D = (D_0, D_1)$, and outputs one bit b rather than a full candidate preimage;
- In the "check" phase of the experiment, the single-bit output b of D_1 is compared to the first bit $\pi^{-1}(y)|_0$ of the preimage of the challenge y ;
- A δ -DPI is a decision permutation inverter which succeeds at the decision inversion experiment with probability at least $\frac{1}{2} + \delta$;

4 Amplification

In this section, we show how to amplify the success probability of search and decision inverters. The construction for the search case is shown in Protocol 1.

Protocol 1 (" ℓ -time repetition" of ϵ -SPI). *Given an ϵ -SPI $S = (S_0, S_1)$ and an integer $\ell > 0$, define a SPI $S[\ell] = (S[\ell]_0, S[\ell]_1)$ as follows.*

1. (Advice Preparation) $S[\ell]_0$ proceeds as follows:

- (a) receives as input a random permutation $\pi : [N] \rightarrow [N]$ and randomness $r \leftarrow \{0, 1\}^*$ and parses the string r into 2ℓ substrings $r = r_0 \dots \|r_{\ell-1}\| r_\ell \dots \|r_{2\ell-1}$ (with lengths as

needed for the next step).

- (b) uses $r_0, \dots, r_{\ell-1}$ to generate ℓ permutations $\{\sigma_i\}_{i=0}^{\ell-1}$ in \mathcal{S}_N , and then runs $S_0(\sigma_i \circ \pi, r_{i+\ell})$ to get a quantum state $\rho_i := \rho_{\sigma_i \circ \pi, r_{i+\ell}}$ for all $i \in [0, \ell-1]$. Finally, $S[\ell]_0$ outputs the quantum state $\bigotimes_{i=0}^{\ell-1} \rho_i$.

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

- (a) receives $\bigotimes_{i=0}^{\ell-1} \rho_i$, randomness r and an image $y \in [N]$ as input.
- (b) parses $r = r_0 \| \dots \| r_{\ell-1} \| r_\ell \| \dots \| r_{2\ell-1}$ and uses the coins $r_0 \| \dots \| r_{\ell-1}$ to reconstruct the permutations $\{\sigma_i\}_{i=0}^{\ell-1}$ in \mathcal{S}_N .
- (c) run the following routine for all $i \in [0, \ell-1]$:
- i. runs S_1 with oracle access to $(\sigma_i \circ \pi)_{\perp \sigma_i(y)}$, which implements the permutation $\sigma_i \circ \pi$ and its inverse (with output \perp on input $\sigma_i(y)$).^a
 - ii. gets back $x_i \leftarrow S_1^{(\sigma_i \circ \pi)_{\perp \sigma_i(y)}}(\rho_i, \sigma_i(y), r_{i+\ell})$.
- (d) queries the oracle $\pi_{\perp y}$ (in the forward direction) on each x_i to see if $\pi(x_i) = y$. If such an x_i is found, output it; otherwise output 0.

^aHow to construct this quantum oracle is described in [Appendix A.1](#).

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 $S[\ell]$ is a $(\ell S, \ell(T+1), 1 - (1 - \epsilon)^\ell)$ -SPI.*

Proof. We consider the execution of the “ ℓ -time repetition” of ϵ -SPI, denoted by SPI $S[\ell]$, in the search permutation inversion experiment defined in [Protocol 1](#). By construction, $S[\ell]$ runs ℓ -many SPI procedures (S_0, S_1) . Because S is assumed to be an ϵ -SPI, it follows that for each iteration $i \in [0, \ell-1]$,

$$\begin{aligned} & \Pr \left[(\sigma_i \circ \pi)^{-1}(\sigma_i(y)) \leftarrow S_1^{(\sigma_i \circ \pi)_{\perp \sigma_i(y)}}(\rho_i, \sigma_i(y), r_{i+\ell}) : \rho_i \leftarrow S_0(\sigma_i \circ \pi, r_{i+\ell}) \right] \\ & \equiv \Pr \left[\pi^{-1}(y) \leftarrow S_1^{\pi \perp y}(\rho_{\pi, r_{i+\ell}}, y, r_{i+\ell}) : \rho_{\pi, r_{i+\ell}} \leftarrow S_0(\pi, r_{i+\ell}) \right] \geq \epsilon, \end{aligned}$$

where 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 S . Then, by the fact that all ℓ trials are completely independent from one another,

$$\begin{aligned} & \Pr \left[\pi^{-1}(y) \leftarrow S[\ell]_1^{\pi \perp y}(\rho, y, r) : \rho \leftarrow S[\ell]_0(\pi, r) \right] \\ & = 1 - \prod_{i=0}^{\ell-1} \Pr \left[x \leftarrow S_1^{(\sigma_i \circ \pi)_{\perp \sigma_i(y)}}(\rho_i, \sigma_i(y), r_{i+\ell}) : \rho_i \leftarrow S_0(\sigma_i \circ \pi, r_{i+\ell}), x \neq (\sigma_i \circ \pi)^{-1}(\sigma_i(y)), \right] \\ & \geq 1 - (1 - \epsilon)^\ell. \end{aligned}$$

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

We also need a variant of the above, due to the requirements of our search lower bound technique.

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

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

such that

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

The proof is given in [Appendix A.2](#).

We also consider amplification for the decision version; the construction is essentially the same, except that the final “check” step is replaced by simply outputting the majority bit. The construction is given explicitly in Protocol 3 in [Appendix A.3](#).

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

The proof is given in [Appendix A.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

First, 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 randomizing 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*

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

Proof. Given an δ -DPI (D_0, D_1) with storage size S and query size T , we can construct a η -DPI $(D[\ell]_0, D[\ell]_1)$ 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 $(D[\ell]_0, D[\ell]_1)$ runs (D_0, 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 $\text{swap}_{a,b}$, which acts as follows for any quantum state of m qubits:

$$\begin{aligned} \text{swap}_{a,b} |w\rangle &= \text{swap}_{a,b} |w_{m-1} \dots w_b \dots w_a \dots w_1 w_0\rangle \\ &= |w_{m-1} \dots w_a \dots w_b \dots w_1 w_0\rangle \end{aligned}$$

Note that $\text{swap}_{k,k}$ is equal to the identity, i.e. $\text{swap}_{k,k} |x\rangle = |x\rangle$ for $x \in [N]$ and $k \in [0, \lceil \log N \rceil - 1]$. We construct a SPI (S_0, S_1) as follows.

1. The algorithm S_0 proceeds as follows:

- (a) S_0 receives a random permutation $\pi : [N] \rightarrow [N]$ 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 $D[\ell]_0(\pi \circ \text{swap}_{0,j}, r_j)$ to obtain quantum advice $\rho_{\pi \circ \text{swap}_{0,j}, r_j}$ for each $j \in [0, \lceil \log N \rceil - 1]$. Finally, S_0 outputs a quantum state $\rho = \bigotimes_{j=0}^{\lceil \log N \rceil - 1} \rho_{\pi \circ \text{swap}_{0,j}, r_j}$. (Note: We let $\rho_j = \rho_{\pi \circ \text{swap}_{0,j}, r_j}$ for the rest of the proof.)

2. The oracle algorithm $S_1 \stackrel{\mathcal{O}_\pi, \mathcal{O}_{\pi^{-1}}}{\perp y}$ 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 $y \in [N]$ as inputs.
- (b) S_1 then runs the following routine for each $j \in [0, \lceil \log N \rceil - 1]$:

i. Run $D[\ell]_1$ with oracle access to $\mathcal{O}_{\pi \circ \text{swap}_{0,j}}$ and $\mathcal{O}_{(\pi \circ \text{swap}_{0,j})^{-1} \perp y}$, where

$$\begin{aligned} \mathcal{O}_{\pi \circ \text{swap}_{0,j}}(|w\rangle_1 |z\rangle_2) &= (\text{swap}_{0,j} \otimes I) \mathcal{O}_\pi (\text{swap}_{0,j} \otimes I) |w\rangle_1 |z\rangle_2 \\ \mathcal{O}_{(\pi \circ \text{swap}_{0,j})^{-1} \perp y}(|w\rangle_1 |z\rangle_2) &= (I \otimes \text{swap}_{0,j}) \mathcal{O}_{\pi^{-1} \perp y} |w\rangle_1 |z\rangle_2 \end{aligned}$$

ii. Let $b_j \leftarrow D[\ell]_1^{(\pi \circ \text{swap}_{0,j}) \perp y}(\rho_j, y, r_j)$ denote the output.

- (c) S_1 outputs $x^* \in [N]$ with respect to the binary decomposition $x^* = \sum_{j=0}^{\lceil \log N \rceil - 1} 2^j \cdot b_j$.

We now argue that the probability that $D[\ell]_1$ 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 $D[\ell]_1$ 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 $k \in [0, \ell - 1]$, the actual modified permutation under use is $\sigma_{i,k} \circ \pi \circ \text{swap}_{0,i}$ and image is $\sigma_{i,k}(y)$. Similarly for term j , $\sigma_{j,k} \circ \pi \circ \text{swap}_{0,j}$ 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.

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

Analyzing the success probability of (S_0, S_1) , we find that

$$\begin{aligned}
& \Pr [\pi^{-1}(y) \leftarrow S_1^{\pi \perp y}(\rho, y, r) : \rho \leftarrow S_0(\pi, r)] \\
&= \Pr \left[\bigwedge_{j=0}^{\lceil \log N \rceil - 1} \pi^{-1}(y)|_j \leftarrow D[\ell]_1^{(\pi \circ \text{swap}_{0,j}) \perp y}(\rho_j, y, r_j) \right] \\
&= \prod_{j=0}^{\lceil \log N \rceil - 1} \Pr \left[\pi^{-1}(y)|_j \leftarrow D[\ell]_1^{(\pi \circ \text{swap}_{0,j}) \perp y}(\rho_j, y, r_j) \right] \\
&\geq \left(1 - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right) \right)^{\lceil \log N \rceil} \\
&\geq 1 - \lceil \log N \rceil \cdot \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right).
\end{aligned}$$

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

5.2 A reduction from unstructured search

Second, we generalize the method used in [\[Nay10\]](#) to give a lower bound for 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 $f : [N] \rightarrow \{0, 1\}$ 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. (*UNIQUESEARCH_N*) *Given a function $f : [N] \rightarrow \{0, 1\}$, such that f maps at most one element to 1, output "yes" if $f^{-1}(1)$ is non-empty and "no" otherwise. In this work, the function f is restricted to map at most one element to 1.*

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) .*

Theorem 5.4. *Let \mathcal{A} be a $(0, T, \delta)$ -DPI. Then there exists a quantum algorithm \mathcal{B} that can solve $\text{UNIQUESEARCH}_{N-1}$ with at most $2T$ quantum queries with distributional error $(\frac{1}{2} - \delta, \frac{1}{2})$.*

Proof. Our proof is similar to that of Nayak [\[Nay10\]](#): given a $(0, T, \delta)$ -DPI \mathcal{A} , we construct another algorithm \mathcal{B} which solves the $\text{UNIQUESEARCH}_{N-1}$ problem. For any uniform image $t \in [N]$, define the "no" and "yes" instances sets (corresponding to the image t) of PERMUTATION_{N-1} (the permutation inversion problem of input size N):

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

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 $h : [N] \rightarrow [N]$ with

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

1. \mathcal{B} first samples a uniform random $t \in [N]$ and some randomness $r \in \{0, 1\}^*$. Then with probability $1/2$, it picks a uniform random permutation $\pi \in \pi_{t,0}$; with probability $1/2$, it picks a $\pi \in \pi_{t,1}$.
2. \mathcal{B} constructs a function $h_{f,\pi,t}$ and $h_{f,\pi,t}^{-1*}$ as follows. If $\pi \in \pi_{t,0}$, for any $i \in \{0, 1\}$ and $j \in \{0, 1\}^{\log N - 1}$,

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

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

$$h_{f,\pi,t}(i||j) = \begin{cases} t & \text{if } i = 0 \text{ and } f(j) = 1, \\ \pi(i||j) & \text{otherwise.} \end{cases} \quad (4)$$

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

$$h_{f,\pi,t}^{-1*}(k||b) = \begin{cases} \pi^{-1}(k)||0 & \text{if } b = 0 \text{ and } w \neq t, \\ 1||1 & \text{otherwise.} \end{cases} \quad (5)$$

3. \mathcal{B} then sends t and r to \mathcal{A} , runs it with quantum oracle access to $h_{f,\pi,t}$ and $h_{f,\pi,t}^{-1*}$, and finally gets back b' . For simplicity, we write this process as $b' \leftarrow \mathcal{A}^{h_{\perp t}}(t; r)$.²
4. \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 $\Pr[\pi \in \pi_{t,0}] = \Pr[\pi \in \pi_{t,1}] = 1/2$, 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 $\pi \in \pi_{t,1}$, $h_{f,\pi,t} = \pi$. It follows that $\mathcal{A}^{h_{\perp t}}(t, r) = \mathcal{A}^{\pi_{\perp t}}(t, r)$.

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

Therefore,

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

In the YES case, $f^{-1}(1)$ is not empty, so function $h_{f,\pi,t}$ 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 $h_{f,\pi,t}$ is uniform in $Q_{\pi,t}$. By arguments in the beginning of this proof, as π is uniform, the function is also uniform in Q_t . Let $p := \Pr_{h_{f,\pi,t} \leftarrow Q_t} [0 \leftarrow \mathcal{A}^{h_{\perp t}}(t; r)]$. Therefore,

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

□

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; 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]. We now give an overview of the proof below.

Suppose we are given an ϵ -SPI S that uses S -many qubits of advice and T -many queries (either in the forward or inverse direction) to random permutation $\pi : [N] \rightarrow [N]$ in order to output $x = \pi^{-1}(y)$ with advantage $\epsilon > 0$ for a random image $y \in [N]$. Using a careful *averaging argument* (Lemma 2.3), we show that we can amplify S to obtain an inverter S' (with only $O(1/\epsilon)$ space-time overhead) such that, with probability $1/5$ over the choice of π and y , S' succeeds at outputting x with probability at least $2/3$ (over the choice of random coins).

Similar to previous works [CLQ19, HXY19], we use information-theoretic lower bounds on the length of quantum random access codes [Wie83, ANTV98, ALMO08], which are a means of encoding classical bits in terms of (a potentially fewer amount of) qubits. In other words, we construct an encoder that compresses the truth table of π by using the power of the search inverter S' , which then allows us to obtain the desired space-time lower bound $ST^2 = \tilde{\Omega}(\epsilon^3 N)$ in [Theorem 6.2](#).

To define a suitable quantum random access code with respect to π , we choose a random subset $\mathcal{R} \in [N]$ (known to the encoder and decoder by means of shared randomness) such that each element is contained in \mathcal{R} with a certain probability. We then define a so-called *good subset* \mathcal{G} of elements $x \in \mathcal{R}$ with the following two properties: S' succeeds at inverting $\pi(x)$ with probability at least $2/3$ and that the *query magnitude* of S' on any element in $\mathcal{R} \setminus \{x\}$ (in the forward direction) and $\pi(\mathcal{R}) \setminus \{\pi(x)\}$ (in the inverse direction) is small when given $y = \pi(x)$ as input.³ Using an appropriate choice of parameters, we can show that our choice of \mathcal{G} is sufficiently large with high probability. The encoding then consists of the following items: a *partial* truth table of π on $[N] \setminus \mathcal{G}$, the entire image $\pi(\mathcal{G})$ as well as the auxiliary state used by S' . To recover the pre-image of the challenge input y , the decoder simply runs the inverter S' on the auxiliary state by simulating the oracle access to the permutation (and its inverse). Note, however, that the decoder only has access to a partial truth table for π , and thus has no means of answering queries on \mathcal{G} (in the forward direction) and $\pi(\mathcal{G})$ (in the inverse direction). Because the query magnitude with respect to the two sets is small, we can use a standard *swapping lemma* trick to show that the state prepared by S' with access to the simulated oracle (that answers incorrectly on $\mathcal{G} \setminus \{x\}$ and $\pi(\mathcal{G}) \setminus \{\pi(x)\}$) is sufficiently close to the state prepared with access to the real oracle. Therefore, the (simulated) inverter S' still succeeds at inverting y with good enough probability.

The statement and proof on a search lower bound for *restricted inverters* is formally described below.

Theorem 6.1. *Let $N \in \mathbb{N}$ and let $S = (S_0, S_1)$ be a $(S, T, 2\epsilon/3)$ -SPI that satisfies*

$$\Pr_{\pi, y} \left[\Pr_r \left[\pi^{-1}(y) \leftarrow S_1^{\pi \perp y}(\rho, y, r) : \rho \leftarrow S_0(\pi, r) \right] \geq \frac{2}{3} \right] \geq \epsilon.$$

Suppose that $\epsilon = \omega(1/N)$, $T = o(\epsilon\sqrt{N})$ and $S \geq 1$. Then, for sufficiently large N we have

$$ST^2 \geq \tilde{\Omega}(\epsilon N).$$

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 2.9](#). Let $S = (S_0, S_1)$ 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

$$\Pr_{\pi, y} \left[\Pr_r \left[\pi^{-1}(y) \leftarrow S_1^{\pi \perp y}(\rho, y, r) : \rho \leftarrow S_0(\pi, r) \right] \geq \frac{2}{3} \right] \geq \epsilon.$$

By the averaging argument in [Lemma 2.3](#) 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

$$\Pr_y \left[\Pr_r \left[\pi^{-1}(y) \leftarrow S_1^{\pi \perp y}(\rho, y, r) : \rho \leftarrow S_0(\pi, r) \right] \geq \frac{2}{3} \right] \geq \frac{\epsilon}{2}.$$

³Here it is crucial that the oracle for the inverse direction rejects if queried on the challenge input $y = \pi(x)$.

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 (Enc, Dec) for encoding permutations is described in detail in [Protocol 2](#). Below, we introduce some additional notation which will be relevant for 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 ⁴

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

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 $[N] \times \{0, 1\}$:

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

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

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

$$x \in \mathcal{R}; \tag{6}$$

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

$$q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \leq c/T(\mathsf{S}). \tag{7}$$

Claim 1. Let $\mathcal{G} \subseteq [N]$ be the set of x which satisfy the conditions in (6) and (7). Then, there exist constants $\gamma, c \in (0, 1)$ such that

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

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

Proof. 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 2.1](#) and the fact that $T(\mathsf{S}) = o(\epsilon\sqrt{N})$, we get

$$\Pr_{\mathcal{R}} \left[|\mathcal{H}| \geq \frac{\gamma|\mathcal{I}|}{2T(\mathsf{S})^2} \right] \geq 0.9, \tag{8}$$

⁴The (reversible) quantum oracle implementation is similar as the one in [Definition 3.1](#). 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.

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

$$q(S_1^{\pi^{\perp y}}, [N] \times \{0, 1\}) \leq T(S). \quad (9)$$

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

$$\begin{aligned} & \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}})] \\ &= \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \Sigma_0^{\mathcal{R}}) + q(S_1^{\pi^{\perp y}}, \Sigma_1^{\mathcal{R}})] && (\Sigma_0^{\mathcal{R}}, \Sigma_1^{\mathcal{R}} \text{ are disjoint, Lemma 2.6}) \\ &= \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \Sigma_0^{\mathcal{R}})] + \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \Sigma_1^{\mathcal{R}})] && (\text{linearity of expectation}) \\ &= \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \mathcal{R} \setminus \{x\} \times \{0\})] \\ &\quad + \mathbb{E}_{\mathcal{R}} [q(S_1^{\pi^{\perp y}}, \pi(\mathcal{R}) \setminus \{\pi(x)\} \times \{1\})] && (\text{by definition of } \Sigma_0^{\mathcal{R}}, \Sigma_1^{\mathcal{R}}) \\ &= \frac{\gamma}{T(S)^2} \cdot q(S_1^{\pi^{\perp y}}, [N] \setminus \{x\} \times \{0\}) \\ &\quad + \frac{\gamma}{T(S)^2} \cdot q(S_1^{\pi^{\perp y}}, \pi([N]) \setminus \{\pi(x)\} \times \{1\}) && (\text{by definition of } \mathcal{R}) \\ &= \frac{\gamma}{T(S)^2} \cdot q(S_1^{\pi^{\perp y}}, [N] \setminus \{x\} \times \{0\}) \\ &\quad + \frac{\gamma}{T(S)^2} \cdot q(S_1^{\pi^{\perp y}}, [N] \setminus \{\pi(x)\} \times \{1\}) && (\pi \text{ is a permutation}) \\ &\leq \frac{\gamma}{T(S)^2} \cdot [q(S_1^{\pi^{\perp y}}, [N] \times \{0\}) + q(S_1^{\pi^{\perp y}}, [N] \times \{1\})] && (\text{supersets, Lemma 2.6}) \\ &= \frac{\gamma}{T(S)^2} \cdot q(S_1^{\pi^{\perp y}}, [N] \times \{0, 1\}) && (\text{disjoint sets, Lemma 2.6}) \\ &\leq \frac{\gamma}{T(S)}. && (\text{by the inequality in (9)}) \end{aligned}$$

Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}} \left[q(S_1^{\pi^{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \geq \frac{c}{T(S)} \right] \leq \frac{T(S)}{c} \cdot \frac{\gamma}{T(S)} = \frac{\gamma}{c}. \quad (10)$$

Let us now denote by \mathcal{J} the subset of $x \in \mathcal{I}$ that satisfy Eq. (6) but not Eq. (7). Note that Eq. (6) and Eq. (7) are independent for each $x \in \mathcal{I}$, since Eq. (6) is about whether $x \in \mathcal{R}$ and Eq. (7) 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 (10), the probability that $x \in \mathcal{I}$ satisfies $x \in \mathcal{J}$ is at most $\gamma^2/(cT(S)^2)$. Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}} \left[|\mathcal{J}| \leq \frac{10|\mathcal{I}|\gamma^2}{cT(S)^2} \right] \geq 0.9. \quad (11)$$

Using (8) and (11), we get with probability at least 0.8 over the the choice of \mathcal{R} ,

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

given that γ is a sufficiently small positive constant. \square

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 QRAC-VL = (Enc, Dec) defined as follows:

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

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

Case 2: $|\mathcal{G}| \geq \frac{\epsilon\gamma N}{4T(S)^2} \left(1 - \frac{5\gamma^2}{c}\right)$. 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 $\alpha \leftarrow S_0(\pi, r)$ 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.)

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

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

Case 2: This corresponds to the flag **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, parses $\alpha_1, \alpha_2, \dots, \alpha_\rho$ and runs $S_1^{\bar{\pi}_{\perp y}}(\alpha_i, y, r)$ for each $i \in [\rho]$ and outputs their majority vote, where we let ^a

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

^aThe (reversible) quantum oracle implementation for $\bar{\pi}_{\perp y}$ is provided in [Appendix C](#).

Let us now analyze the performance of our QRAC-VL scheme (Enc, 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

the Swapping Lemma (Lemma 2.7) and Lemma 2.6:

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

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.,

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

Therefore, except with probability $1 - 0.4\epsilon$, our encoding will result in the flag `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 `case = 2`, and the size equals

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

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

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

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 2.1. Overall, the average encoding length is

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

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 2.9](#), we have

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

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

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

As we are conditioning on the event that \mathcal{G} is large, i.e.

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

plugging in the lower bound on $|\mathcal{G}|$, we have that for sufficiently large N ,

$$\begin{aligned} S(\mathcal{S}) &\geq \tilde{\Omega}(|\mathcal{G}|) \\ S(\mathcal{S}) \cdot T(\mathcal{S})^2 &\geq \tilde{\Omega}(\epsilon N). \end{aligned}$$

This gives the desired space-time trade-off. \square

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](#), [Zha18](#)].

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 \mathcal{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*

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

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

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

such that

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

From [Theorem 6.1](#) it follows that for sufficiently large $N \geq 1$,

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

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

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

the trade-off between space and time complexities is

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

□

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

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 $T = \tilde{o}(\delta^2 \sqrt{N})$ and $S \geq 1$. Then, for sufficiently large N we have*

$$S(\mathsf{D}) \cdot T(\mathsf{D})^2 \gtrsim \tilde{\Omega}(\delta^6 N).$$

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

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

S satisfies the space-time trade-off lower bound

$$\begin{aligned} n^3 \left(\frac{n(1+2\delta)}{\delta^2}\right)^3 S(\mathsf{D}) \cdot T(\mathsf{D})^2 &\geq \tilde{\Omega}(\eta^3 N) \approx \tilde{\Omega}(N) \\ S(\mathsf{D}) \cdot T(\mathsf{D})^2 &\gtrsim \tilde{\Omega}(\delta^6 N) \end{aligned}$$

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.

In the case of no 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 $(0, T, \delta)$ -DPI. Then $T^2 \geq \tilde{\Omega}(\delta N)$.*

Proof. Since D is a $(0, T, \delta)$ -DPI, by [Theorem 5.4](#) we get a $2T$ -query algorithm for the unique search problem with distributional error $(\frac{1}{2} - \delta, \frac{1}{2})$. Since the "yes" and "no" cases are uniformly distributed, we can write the distribution error as

$$\delta' = \frac{1}{2} \left(\frac{1}{2} - \delta \right) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} - \frac{\delta}{2}.$$

For average-case unique search problem, previous work [[Gro96](#), [Zal99](#), [Zha18](#)] gave a optimal bound $T^2 \geq \Omega(pN)$, where p is the success probability of the unique search problem. This concludes our proof. \square

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 two-sided permutation inversion problem. Recall that a symmetric-key encryption scheme consists of three algorithms:

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

When the randomness is to be selected uniformly, we suppress it, e.g., we write $\text{Gen}(1^n)$.

Consider the following security definition.

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

1. A key k is generated by running $\text{Gen}(1^n)$;
2. \mathcal{A} gets quantum oracle access to $\text{Enc}_k(\cdot; \cdot)$ and $\text{Dec}_k(\cdot)$;
3. Uniform $m \in \mathcal{M}$ and $r \in \mathcal{R}$ are chosen, and a challenge ciphertext $c = \text{Enc}_k(m; r)$ is computed and given to \mathcal{A} ;
4. \mathcal{A} gets quantum oracle access to $\text{Enc}_k(\cdot; \cdot)$ and $\text{Dec}_k^{\perp c}(\cdot)$. Eventually, it outputs a bit b .
5. The experiment outputs 1, if $b = m|_0$, 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. The acronym OW stands for “one-way” and QCCRA stands for “quantum chosen-ciphertext randomness-access attack.” 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 $\pi^{-1}(c)$.

PRP Scheme. Let $\{P_k : \{0, 1\}^{2n} \mapsto \{0, 1\}^{2n}\}$ be a family quantum-query-secure strong pseudo-random permutations (PRPs) [KL20, Zha16] 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 OW-QCCRA-secure. Specifically, we apply our decision inversion lower bound (Corollary 6.3) to prove the below security theorem.

Theorem 7.2. *The PRP scheme is OW-QCCRA-secure. In other words, for all quantum polynomial time (QPT) adversaries \mathcal{A} , it holds that*

$$\Pr\left[\text{Exp}_{\mathcal{A}, \text{RP}}^{\text{OW-QCCRA}}(1^n) = 1\right] \leq \frac{1}{2} + \text{negl}(n).$$

Proof. Given an adversary \mathcal{A} that attacks the RP scheme in the OW-QCCRA experiment, we can construct a δ -DPI $D = (D_0, D_1)$ in DecisionInvert_D , which takes place as follows:

1. **(sample instance and coins)** a random permutation $\pi : \{0, 1\}^n \rightarrow \{0, 1\}^n$, a random image $y \leftarrow \{0, 1\}^n$, and a random string $r \leftarrow \{0, 1\}^*$ are sampled;
2. **(prepare advice)** D_0 is given the whole permutation table of π . Then it constructs oracles $\text{Enc}(\cdot; \cdot) = \pi(\cdot||\cdot)$ and $\text{Dec}(\cdot) = \pi^{-1}(\cdot)$ and gives \mathcal{A} poly-time quantum oracle access. D_0 will get back an output state ρ and then output it.
3. **(invert)** D_1 is run with a random instance y , advice ρ and quantum oracle access \mathcal{O}_π and $\mathcal{O}_{\pi^{-1}}$. It then directly passes y and two oracles to \mathcal{A} and gets back a bit b and outputs it.
4. **(check)** If $b = \pi^{-1}(y)|_0$, output 1; otherwise output 0.

It trivially follows that

$$\Pr\left[\text{Exp}_{\mathcal{A}, \text{RP}}^{\text{OW-QCCRA}}(1^n) = 1\right] \leq \Pr[\text{DecisionInvert}_D = 1].$$

By assumption we have that, for all QPT \mathcal{A} , there exists a negligible function negl such that

$$\left| \Pr \left[\mathcal{A}^{P_k(\cdot), P_k^{-1}(\cdot)}(1^n) = 1 \right] - \Pr \left[\mathcal{A}^{\pi(\cdot), \pi^{-1}(\cdot)}(1^n) = 1 \right] \right| \leq \text{negl}(n),$$

where P_k is a quantum-query-secure strong pseudorandom permutation [Zha16]. Therefore

$$\begin{aligned} \Pr \left[\text{Exp}_{\mathcal{A}, \text{PRP}}^{\text{OW-QCCRA}}(1^n) = 1 \right] &\leq \Pr \left[\text{Exp}_{\mathcal{A}, \text{RP}}^{\text{OW-QCCRA}}(1^n) = 1 \right] + \text{negl}(n) \\ &\leq \Pr[\text{DecisionInvert}_{\text{D}} = 1] + \text{negl}(n) \\ &= \frac{1}{2} + \delta + \text{negl}(n). \end{aligned}$$

If $\delta < w(1/N)$, the security statement directly follows. Otherwise, by Corollary 6.3,

$$\delta \leq \tilde{O} \left(\frac{S^{1/6} T^{1/3}}{2^{n/6}} \right) \leq \text{negl}(n),$$

as both S and T are of polynomial size. \square

We also show that the (idealized, inefficient) RP scheme actually satisfies an even stronger version of the OW-QCCRA security notion. In this strengthening, \mathcal{A} is computationally unlimited, and also gets *unlimited* quantum oracle access to $\text{Enc}_k(\cdot; \cdot)$ and $\text{Dec}_k(\cdot)$ in the pre-challenge phase. This is defined and proved formally below.

Definition 7.3. (*strong OW-QCCRA*) Let $\text{SKE} = (\text{Gen}, \text{Enc}, \text{Dec})$ be a private-key encryption scheme. We say that SKE is *strong OW-QCCRA-secure* if the advantage for any unbounded quantum adversary \mathcal{A} in the following experiment is at most negligible:

1. A key k is generated by running $\text{Gen}(1^n)$;
2. \mathcal{A} gets **unlimited** quantum oracle access to $\text{Enc}_k(\cdot; \cdot)$ and $\text{Dec}_k(\cdot)$; but can only writes down a $\text{poly}(n)$ -qubits state;
3. Uniform $m \in \mathcal{M}$ and $r \in \mathcal{R}$ are chosen, and a challenge ciphertext $c = \text{Enc}_k(m; r)$ is computed and given to \mathcal{A} ;
4. \mathcal{A} now gets poly-time quantum oracle access to $\text{Enc}_k(\cdot; \cdot)$ and $\text{Dec}_k^{\perp c}(\cdot)$. Eventually, it outputs a bit b .
5. The experiment outputs 1, if $b = m|_0$, and 0 otherwise.

Theorem 7.4. The RP scheme is *strong OW-QCCRA-secure*. In other words, for all quantum adversaries \mathcal{A} , it holds that

$$\Pr \left[\text{Exp}_{\mathcal{A}, \text{RP}}^{\text{strong OW-QCCRA}}(1^n) = 1 \right] \leq \frac{1}{2} + \text{negl}(n).$$

Proof. Given an adversary \mathcal{A} that attacks RP scheme in the strong OW-QCCRA experiment, we can construct a δ -DPI $\text{D} = (\text{D}_0, \text{D}_1)$ in $\text{DecisionInvert}_{\text{D}}$.

1. (**sample instance and coins**) a truly random permutation $\pi : \{0, 1\}^n \rightarrow \{0, 1\}^n$, a random image $y \leftarrow \{0, 1\}^n$, and a random string $r \leftarrow \{0, 1\}^*$ are sampled;

2. **(prepare advice)** D_0 is given the whole permutation table of π and grants \mathcal{A} **unlimited** oracle access to π and π^{-1} . Then D_0 gets back an output state ρ and outputs it.
3. **(invert)** D_1 is run with a random instance y , advice ρ and quantum oracle access to \mathcal{O}_π and $\mathcal{O}_{\pi^{-1}}$. It then directly passes y and two oracles to \mathcal{A} and gets back a bit b and outputs it.
4. **(check)** If $b = \pi^{-1}(y)|_0$, output 1; otherwise output 0.

It trivially follows that

$$\Pr\left[\text{Exp}_{\mathcal{A},\text{RP}}^{\text{strong OW-QCCRA}}(n) = 1\right] \leq \Pr[\text{DecisionInvert}_D = 1] = \frac{1}{2} + \delta.$$

If $\delta < w(1/N)$, the statement directly follows. Otherwise, by [Corollary 6.3](#), assuming storage size S and $T = \tilde{o}(\delta^2\sqrt{2^n})$,

$$\delta \leq \tilde{O}\left(\frac{S^{1/6}T^{1/3}}{2^{n/6}}\right).$$

The assumption can also be expressed as $\delta \geq \tilde{\omega}\left(\frac{T^{1/2}}{2^{n/4}}\right)$ which is smaller than $\left(\frac{S^{1/6}T^{1/3}}{2^{n/6}}\right)$. Therefore, given $S = \text{poly}(n)$ given in the experiment, we can directly bound the success probability by $\left(\frac{S^{1/6}T^{1/3}}{2^{n/6}}\right)$ which is negligible of n . \square

Note that the adversary \mathcal{A} in the above experiment has **unlimited** query access while the adversary in [Theorem 7.2](#) only has poly-time query access. Therefore the truly random permutation RP cannot be replaced by the pseudorandom permutation PRP while still satisfying **strong OW-QCCRA**.

Finally, we remark that the above results hold for the following strengthening of OW-QCCRA, described as follows. Suppose that an encryption scheme satisfies the property that there exists an *alternative* decryption algorithm which 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.

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) (cit. on p. 4).
- [ABKM21] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. “Post-Quantum Security of the Even-Mansour Cipher”. In: *arXiv preprint arXiv:2112.07530* (2021) (cit. on p. 4).
- [ALMO08] Andris Ambainis, Debbie Leung, Laura Mancinska, and Maris Ozols. *Quantum Random Access Codes with Shared Randomness*. 2008. DOI: [10.48550/ARXIV.0810.2937](https://arxiv.org/abs/0810.2937). URL: <https://arxiv.org/abs/0810.2937> (cit. on pp. 3, 7, 16).
- [Amb02] Andris Ambainis. “Quantum lower bounds by quantum arguments”. In: *Journal of Computer and System Sciences* 64.4 (2002), pp. 750–767 (cit. on pp. 1, 4).
- [ANTV98] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. *Dense Quantum Coding and a Lower Bound for 1-way Quantum Automata*. 1998. DOI: [10.48550/ARXIV.QUANT-PH/9804043](https://arxiv.org/abs/quant-ph/9804043). URL: <https://arxiv.org/abs/quant-ph/9804043> (cit. on pp. 3, 7, 16).
- [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 (cit. on p. 1).
- [BDPA11] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. *Cryptographic sponge functions*. Submission to NIST (Round 3). 2011. URL: <http://sponge.noekeon.org/CSF-0.1.pdf> (cit. on pp. 1, 3).
- [BY23] Aleksandrs Belovs and Duyal Yolcu. “One-Way Ticket to Las Vegas and the Quantum Adversary”. In: *arXiv preprint arXiv:2301.02003* (2023) (cit. on p. 4).
- [CBH⁺17] Jan Czajkowski, Leon Groot Bruinderink, Andreas Hülsing, Christian Schaffner, and Dominique Unruh. *Post-quantum security of the sponge construction*. Cryptology ePrint Archive, Paper 2017/771. <https://eprint.iacr.org/2017/771>. 2017. URL: <https://eprint.iacr.org/2017/771> (cit. on p. 3).
- [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 (cit. on pp. 4, 9, 10).
- [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) (cit. on pp. 3, 4, 7, 9, 15, 16).
- [CMSZ21] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. *Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability*. 2021. arXiv: [1904.11477](https://arxiv.org/abs/1904.11477) [quant-ph] (cit. on p. 3).
- [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 (cit. on pp. 4, 9).

- [DH08] Catalin Dohotaru and Peter Hoyer. “Exact quantum lower bound for Grover’s problem”. In: *arXiv preprint arXiv:0810.3647* (2008) (cit. on p. 21).
- [DKRS21] Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. “Quantum Time/Memory/Data Tradeoff Attacks”. In: *Cryptology ePrint Archive* (2021) (cit. on p. 4).
- [Dwo15] Morris Dworkin. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. en. 2015-08-04 2015. DOI: <https://doi.org/10.6028/NIST.FIPS.202> (cit. on p. 3).
- [FK15] Bill Fefferman and Shelby Kimmel. “Quantum vs classical proofs and subset verification”. In: *arXiv preprint arXiv:1510.06750* (2015) (cit. on p. 4).
- [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 (cit. on pp. 1, 21, 23).
- [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 (cit. on pp. 3, 4, 9, 10, 15, 16, 22).
- [KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020 (cit. on pp. 1, 24).
- [Liu22] Qipeng Liu. “Non-uniformity and Quantum Advice in the Quantum Random Oracle Model”. In: *arXiv preprint arXiv:2210.06693* (2022) (cit. on pp. 4, 9).
- [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) (cit. on pp. 3, 4, 9, 15).
- [Nay10] Ashwin Nayak. “Inverting a permutation is as hard as unordered search”. In: *arXiv preprint arXiv:1007.2899* (2010) (cit. on pp. 1, 3, 4, 13).
- [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 (cit. on p. 26).
- [Ros21] Ansis Rosmanis. “Tight bounds for inverting permutations via compressed oracle arguments”. In: *arXiv preprint arXiv:2103.08975* (2021) (cit. on p. 4).
- [Vaz98] Umesh Vazirani. “On the power of quantum computation”. In: *Philosophical Transactions of the Royal Society of London A* 365: 1759–1768 (1998) (cit. on p. 7).
- [Wie83] Stephen Wiesner. “Conjugate Coding”. In: *SIGACT News* 15.1 (Jan. 1983), pp. 78–88. ISSN: 0163-5700. DOI: [10.1145/1008908.1008920](https://doi.org/10.1145/1008908.1008920). URL: <https://doi.org/10.1145/1008908.1008920> (cit. on pp. 3, 7, 16).
- [Zal99] Christof Zalka. “Grover’s quantum searching algorithm is optimal”. In: *Physical Review A* 60.4 (1999), p. 2746 (cit. on p. 23).
- [Zha16] Mark Zhandry. “A note on quantum-secure PRPs”. In: *arXiv preprint arXiv:1611.05564* (2016) (cit. on pp. 3, 24, 25).

[Zha18] Mark Zhandry. *How to Record Quantum Queries, and Applications to Quantum Indifferentiability*. Cryptology ePrint Archive, Paper 2018/276. <https://eprint.iacr.org/2018/276>. 2018. URL: <https://eprint.iacr.org/2018/276> (cit. on pp. 4, 21, 23).

A Amplification proofs

A.1 Quantum oracle construction in Protocol 1

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

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

$$\begin{aligned}
& |w\rangle_1 |z\rangle_2 \\
\begin{array}{c} \text{add aux register} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z\rangle_2 |0\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\sigma_i, 1, \text{aux}} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z\rangle_2 |\sigma_i(w)\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\pi, \text{aux}, 2} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi \circ \sigma_i(w)\rangle_2 |\sigma_i(w)\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\sigma_i, 1, \text{aux}} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi \circ \sigma_i(w)\rangle_2 |0\rangle_{\text{aux}} \\
\begin{array}{c} \text{drop aux} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi \circ \sigma_i(w)\rangle_2.
\end{aligned}$$

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

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

$$\begin{aligned}
& |w\rangle_1 |z\rangle_2 \\
\begin{array}{c} \text{add aux register} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z\rangle_2 |0\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\sigma_{i,*}^{-1}, 1, \text{aux}} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z\rangle_2 |\sigma_{i,*}^{-1}(w)\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\pi_{\perp y}^{-1}, \text{aux}, 2} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi_{\perp y}^{-1} \circ \sigma_{i,*}^{-1}(w)\rangle_2 |\sigma_{i,*}^{-1}(w)\rangle_{\text{aux}} \\
\begin{array}{c} \mathcal{O}_{\sigma_{i,*}^{-1}, 1, \text{aux}} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi_{\perp y}^{-1} \circ \sigma_{i,*}^{-1}(w)\rangle_2 |0\rangle_{\text{aux}} \\
\begin{array}{c} \text{drop aux} \\ \hline \end{array} & \rightarrow |w\rangle_1 |z \oplus \pi_{\perp y}^{-1} \circ \sigma_{i,*}^{-1}(w)\rangle_2,
\end{aligned}$$

where $\sigma_{i,*}^{-1} : [N] \times \{0, 1\} \rightarrow [N] \times \{0, 1\}$ is given below

$$\sigma_{i,*}^{-1}(w||b) := \sigma_i^{-1}(w)||b.$$

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

A.2 Another amplification lemma proof

Lemma 4.2. *Let $S = (S_0, S_1)$ be an ϵ -SPI with space and time complexity given by $S(S)$ and $T(S)$, respectively, for some $\epsilon > 0$. Then, we can construct an SPI $S[l] = (S[l]_0, S[l]_1)$ with space and time complexities*

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

such that

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

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 $S[l] = (S[l]_0, S[l]_1)$, with $\eta = 1 - (1 - \epsilon)^\ell$ and space and time complexities $S(S[l]) = \ell \cdot S(S)$ and $T(S[l]) = \ell \cdot (T(S) + 1)$. In other words,

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

Let \mathcal{S}_N denote the set of permutations over $[N]$. From Lemma 2.3 it follows that there exists $\theta = 7/9$ and a subset $\mathcal{X}_\theta \subseteq \mathcal{S}_N \times [N]$ of size at least

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

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

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

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

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

This proves the claim. □

A.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 $D = (D_0, D_1)$, the construction of an “ ℓ -time serial repetition of D ” $D[\ell] = (D[\ell]_0, D[\ell]_1)$ is as follows:*

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

- (a) $D[\ell]_0$ receives as input a random permutation $\pi : [N] \rightarrow [N]$ and randomness $r \leftarrow \{0, 1\}^*$ and parses the string r into 2ℓ substrings, i.e. $r = r_0 \| \dots \| r_{\ell-1} \| r_\ell \| \dots \| r_{2\ell-1}$ (the length is clear in context).
- (b) $D[\ell]_0$ uses $r_0, \dots, r_{\ell-1}$ to generate ℓ permutations $\{\sigma_i\}_{i=0}^{\ell-1}$ in \mathcal{S}_N , and then runs $D_0(\sigma_i \circ \pi, r_{i+\ell})$ to get a quantum state $\rho_i := \rho_{\sigma_i \circ \pi, r_{i+\ell}}$ for all $i \in [0, \ell - 1]$. Finally, $D[\ell]_0$

outputs a quantum state $\bigotimes_{i=0}^{\ell-1} \rho_i$.

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

- (a) $D[\ell]_1$ receives $\bigotimes_{i=0}^{\ell-1} \rho_i$, randomness r and an image $y \in [N]$ as input.
- (b) $D[\ell]_1$ parses $r = r_0 \| \dots \| r_{\ell-1} \| r_\ell \| \dots \| r_{2\ell-1}$ and uses the coins $r_0 \| \dots \| r_{\ell-1}$ to generate ℓ different permutations $\{\sigma_i\}_{i=0}^{\ell-1}$ in \mathcal{S}_N .
- (c) $D[\ell]_1$ then runs the following routine for all $i \in [0, \ell-1]$:
 - i. Runs D_1 with oracle access to $(\sigma_i \circ \pi)_{\perp \sigma_i(y)}$, which implements the permutation $\sigma_i \circ \pi$ and its inverse (but \perp at $\sigma_i(y)$).
 - ii. Get back $b_i \leftarrow D_1^{(\sigma_i \circ \pi)_{\perp \sigma_i(y)}}(\rho_i, \sigma_i(y), r_{i+\ell})$.
- (d) $D[\ell]_1$ outputs b^* which is the majority vote of $\{b_0, \dots, b_{\ell-1}\}$.

Lemma 4.3. Let (D_0, 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 $D[\ell] = (D[\ell]_0, D[\ell]_1)$, which is an η -DPI for $\eta \geq \frac{1}{2} - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right)$, and has space and time complexities given by

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

Proof. Let (D_0, 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 $D[\ell]$, which we define in Protocol 3. For each iteration $i \in [0, \ell-1]$, we have

$$\begin{aligned} & \Pr[b_i = \pi^{-1}(y)|_0] \\ &= \Pr\left[(\sigma_i \circ \pi)^{-1}(\sigma_i(y))|_0 \leftarrow D_1^{(\sigma_i \circ \pi)_{\perp \sigma_i(y)}}(\rho_i, \sigma_i(y), r_{i+\ell}) : \rho_i \leftarrow D_0(\sigma_i \circ \pi, r_{i+\ell})\right] \\ &\equiv \Pr\left[\pi^{-1}(y)|_0 \leftarrow D_1^{\pi \perp y}(\rho_{\pi, r_{i+\ell}}, y, r_{i+\ell}) : \rho_{\pi, r_{i+\ell}} \leftarrow D_0(\pi, r_{i+\ell})\right] \geq \frac{1}{2} + \delta, \end{aligned}$$

where 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 . Let X_i be the indicator variable for the event that $b_i = \pi^{-1}(y)|_0$. Letting $X = \sum_{i=0}^{\ell-1} X_i$, we have that $\mathbb{E}[X] \geq \ell \cdot (\frac{1}{2} + \delta)$ by the linearity of expectation. Note that $D[\ell]$ succeeds in DecisionInvert if and only if $D[\ell]_1$ can output $b^* = \pi^{-1}(y)|_0$, i.e. $X > \frac{\ell}{2}$ in which case more than half of the elements in $\{b_0, \dots, b_{\ell-1}\}$ are equal to $\pi^{-1}(y)|_0$. By the multiplicative Chernoff bound in Lemma 2.1, the probability that 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. \square

B 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 $h_{f, \pi, t}$ and $h_{f, \pi, t}^{-1*}$. Here we give detailed constructions of $\mathcal{O}_{h_{f, \pi, t}}$ and $\mathcal{O}_{h_{f, \pi, t}^{-1*}}$. Note that π is sampled

by \mathcal{B} and so it is easy for it to construct quantum oracles \mathcal{O}_π and $\mathcal{O}_{\pi_{\perp t}^{-1}}$. Since $h_{f,\pi,t}^{-1*} = \pi_{\perp t}^{-1}$, the partial inverse oracle $\mathcal{O}_{h_{f,\pi,t}^{-1*}}$ can be simply simulated by $\mathcal{O}_{\pi_{\perp t}^{-1}}$. So we only need to show how to construct $\mathcal{O}_{h_{f,\pi,t}}$.

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

$$h_{f,\pi,t}(x_0 \dots x_{n-1}) = (x_0 \cdot f(x_1 \dots x_{n-1})) \cdot t + \overline{(x_0 \cdot f(x_1 \dots x_{n-1}))} \cdot \pi(x).$$

Then define a function $g : [N] \rightarrow \{0, 1\}$, such that $g(x) = x_0 \cdot f(x_1 \dots x_{n-1})$. 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 $\mathcal{O}_{h_{f,\pi,t}}$ 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 $y \oplus (g(x) \cdot t) \oplus (\overline{g(x)} \cdot \pi(x)) = y \oplus h_{f,\pi,t}(x)$. Therefore, to respond to one query to $\mathcal{O}_{h_{f,\pi,t}}$, \mathcal{B} needs to query \mathcal{O}_f *twice* (once for computing and once for eliminating). The same thing can be done when $\pi \in \pi_{t,1}$.

C 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 $\mathcal{O}_{\bar{\pi}_{\perp y}^{-1}}$, 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

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

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 $\mathcal{O}_{\bar{\pi}_{\perp y}^{-1}}$.

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