ABSTRACT

Title of Dissertation: LIMITATIONS OF
QUANTUM METHODS FOR
BREAKING CRYPTOGRAPHY
AND FOR OPTIMIZATION

Kaiyan Shi
Doctor of Philosophy, 2026

Dissertation Directed by: Professor Gorjan Alagic
Department of Computer Science

The development of quantum computers raises fundamental questions about the extent to which quantum algorithms can outperform classical methods. This thesis studies limitations of quantum methods in two domains: symmetric-key cryptography and combinatorial optimization.

The first part of the thesis investigates the post-quantum security of block-cipher-based constructions. We obtain the first post-quantum security proofs for several important symmetric-key primitives, including the key-length extension scheme FX, the tweakable block ciphers LRW, and a broad class of block-cipher modes. Our results show that quantum adversaries with limited quantum oracle access do not achieve attacks substantially stronger than those captured by tight classical bounds.

The thesis also studies permutation inversion in the quantum query model, proving lower bounds showing that the problem remains essentially as hard as standard inversion even with inverse access or quantum advice. These results highlight fundamental barriers to quantum speedups that are independent of specific cryptographic constructions.

Finally, we study multiangle QAOA and show numerically that exploiting problem symmetries can reduce parameter complexity while preserving most of the algorithm's performance, although the extent and generality of such reductions are inherently limited. This study provides insight into both the potential and the limitations of quantum optimization algorithms.

LIMITATIONS OF QUANTUM METHODS FOR BREAKING CRYPTOGRAPHY AND OPTIMIZATION

by

Kaiyan Shi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2026

Advisory Committee:
Professor Andrew Childs, Chair
Professor Gorjan Alagic, Advisor & Co-chair
Professor Yun Yang, Dean's Representative
Professor Alexey V. Gorshkov
Professor Alan Zaoxing Liu

Acknowledgments

I am deeply grateful to my advisor, Gorjan, for his guidance and support over the past six years. It has been a true privilege to be his student throughout this journey. He has offered me invaluable ideas in research, both in broad vision and in careful detail. He has taught me not only how to do research, but also, through his example, what it means to be gentle, kind, and thoughtful. His influence on me has extended beyond research and has profoundly shaped me as a person.

I am also sincerely grateful to Alexey for supporting me during my last semester and for offering sharp insights and incisive advice during our research meetings. I have learned a great deal from the way he thinks, and his genuine curiosity and enthusiasm for research have deeply influenced me.

I would like to sincerely thank my committee members, Andrew, Gorjan, Yun, Alan, and Alexey, for agreeing to serve on my committee. Their valuable suggestions and questions helped me develop a deeper understanding of the scope and significance of my research during my defense. I also truly appreciate their prompt responses and generous support. Without their time and encouragement, this milestone would not have been possible.

I would like to sincerely thank my undergraduate advisor, Adam, for his guidance and for bringing me into the world of theoretical computer science and cryptography. He was the one who first introduced me to research, cryptography, and differential privacy. He has always been sharp, patient, and inspiring, and he has greatly shaped my understanding of what a wonderful professor, teacher and mentor can be.

I would like to sincerely thank all of my past mentors, Jeffrey, Ruslan, Omar, Kaushik, and Shinnyih, for their generous mentorship, dedicated guidance, and unwavering support during my internships at Argonne, JPMorgan, and Bank of America. Every step along the way has counted, and I can hardly express how lucky I feel to have always met such wonderful, thoughtful, and inspiring mentors throughout my journey.

I would like to sincerely thank all of my collaborators. Without their support, valuable discussions, guidance and contributions, this work would not have been possible.

I would like to sincerely thank all of my friends. Thank you for being my friends, and thank you for letting me into your worlds.

最后，我想深深地感谢我的父母，告诉我身后永远是家。我永远可以回家。

Table of Contents

Chapter 1: Introduction

1.1 Motivation

The development of large-scale quantum computers poses a fundamental challenge to both modern cryptography and the theory of algorithms. On the one hand, quantum algorithms have demonstrated dramatic advantages over classical ones in certain settings, most notably through Shor's algorithm for integer factorization and discrete logarithms. On the other hand, for many computational problems of practical and theoretical interest, it remains unclear whether quantum computation can offer substantial improvements, or whether there exist intrinsic limitations on the power of quantum methods.

This thesis is motivated by the goal of understanding such limitations. We aim to characterize the extent to which quantum computation can provide advantages in two broad domains: symmetric-key cryptography and combinatorial optimization. In both settings, quantum computation is often believed to offer meaningful advantages, yet existing theoretical and empirical evidence suggests that these advantages may be more limited than is commonly expected.

Post-quantum security. Post-quantum cryptography (PQC) usually refers to cryptographic schemes designed to remain secure against adversaries equipped with large, fault-tolerant quantum computers. Shor's algorithm [Sho94] shows that such adversaries can efficiently solve integer factorization and discrete logarithms, thereby completely breaking widely deployed public-key primitives such as RSA and elliptic curve cryptography (ECC). In contrast, the impact of quantum computation on symmetric-key cryptography is more nuanced and remains an active area of research. The best-known generic quantum attack is Grover's algorithm [Gro96], which provides a quadratic speedup for unstructured key search. This also means that symmetric systems are often believed to be protectable by increasing key lengths. However, understanding whether additional quantum advantages arise for symmetric-key constructions requires more careful analysis.

In the study of post-quantum security of symmetric primitives, it is standard to distinguish between two models of quantum adversaries, known as the Q1 and Q2 models. In the Q1 model, the adversary has quantum computational power but interacts with keyed oracles only through classical queries, whereas in the stronger Q2 model the adversary is additionally allowed to query these oracles in quantum superposition.

Q2 model. Superposition access can enable attacks that go well beyond Grover's generic quadratic speedup, because it permits period-finding and related techniques to exploit algebraic structure in constructions. A series of works has shown that superposition query access enables Simon-type period finding and related linearization techniques that exploit hidden structure, leading to complete breaks of several symmetric constructions in the Q2 model [KLLNP16, BLNPS21, BCF+22]. Some other work also studied what remains secure in the Q2 model and how to repair insecure designs. [ATTU16, LL23]. This model usually serves as a useful theoretical benchmark for understanding the ultimate limits of security in the presence of quantum access.

Q1 model. In contrast to the Q2 model, in the Q1 model the adversary is quantum but is restricted to making only classical oracle queries to keyed oracles. A common heuristic is that Grover's algorithm captures the main quantum threat, so that increasing key sizes should suffice. However, this intuition has been challenged by a line of Q1 attacks that use a small number of classical queries together with quantum period-finding and related techniques [BHNP+19, KM12, HS18, BSS22].

At the same time, there are also positive results and lower bounds in the Q1 setting [JST21, ABKM22, ABK+24]. Together, these works suggest that many structured block-cipher constructions remain secure against Q1 adversaries up to well-understood limits, and that proving tight post-quantum bounds requires tools beyond the basic Grover heuristic.

Motivated by the gap between powerful Q1 attacks and the usual security expectation, this thesis studies the post-quantum security of symmetric-key constructions in the Q1 model. We focus on Q1 since it is often the more realistic setting: in typical applications, an adversary interacts with encryption and MAC functionality through a classical interface (e.g., a network API, protocol endpoint, or hardware device) that necessarily measures/parses the input and returns a classical string. Even if the adversary possesses a quantum computer, classical channels do not transmit unknown quantum states, so the adversary cannot submit a superposition of queries and receive a coherent superposition of responses.

Our goal is to establish tight bounds and general proof techniques that clarify the limitations on the advantages quantum algorithms can achieve against practical block-cipher-based schemes.

Quantum optimization algorithms. Beyond cryptography, quantum algorithms are also widely studied as potential tools for solving combinatorial optimization problems. In this setting, unlike the dramatic breaks caused by Shor's algorithm in public-key cryptography, quantum methods are not expected to yield exponential speedups in general, but they may still offer practical improvements for certain problem families. A leading candidate is the Quantum Approximate Optimization Algorithm (QAOA), a hybrid quantum-classical variational algorithm that prepares parameterized quantum states using alternating applications of a problem Hamiltonian and a mixing Hamiltonian, and then uses classical optimization to tune these parameters. QAOA has attracted significant attention both as a near-term approach to optimization and as a framework for studying the structure and limits of quantum advantage. Motivated by the broader goal of understanding the limitations of quantum methods, this thesis investigates when and how QAOA can be made more efficient, and what limits its performance and scalability.

1.2 Roadmap

In Chapter 2, we introduce the basic notation and background needed throughout the thesis, including core quantum knowledge such as the quantum circuit and query models, as well as standard cryptographic security definitions and preliminaries on random and pseudorandom permutations.

In Chapter 3, we study the post-quantum security of block-cipher-based symmetric cryptography in the Q1 setting [ABMS25]. We develop general proof tools, including resampling techniques in the ideal model, and apply them to obtain tight or near-tight security bounds for key-length extension schemes such as FX, for tweakable block ciphers such as LRW and XEX2, and for a broad class of block cipher modes.

In Chapter 4, we study the permutation inversion problem in the two-sided quantum random permutation model [ABPS24]. In this model, a publicly accessible oracle is treated as a completely random permutation, and an adversary can make superposition queries to both the permutation and its inverse. The permutation inversion problem naturally arises in the context of sponge hashing [BDPA11] used by the international hash function standard SHA-3 [Dwo15]. Understanding the trade-offs between time and space complexity is crucial for balancing real-world applications while ensuring adequate security.

In Chapter 5, we turn to quantum optimization and study QAOA. We focus on multiangle variants of QAOA and investigate how problem symmetries can be used to reduce the number of variational parameters without significantly degrading performance [SHS+22]. Using Max-Cut as a benchmark, we provide numerical evidence that symmetry-based parameter reduction can sometimes simplify optimization, while also highlighting intrinsic limitations on when such reductions are effective.

1.3 Additional Work

In addition to the three research work presented in this thesis, I also contributed to several projects in other relate areas.

In [SCK+24], we develop a formal framework for relativistic zero-knowledge quantum proofs of knowledge with classical communication. We construct quantum knowledge extractors for existing relativistic protocols satisfying a weaker variant of special soundness. We further show that, for certain proof systems without special soundness, one can still construct a quantum knowledge extractor, though this requires multiple rounds of rewinding.

In [KSGB23], we present a generic black-box framework for transforming arbitrary hypothesis tests into differentially private ones. We analyze the resulting private tests both theoretically and experimentally, and show that the framework performs well even on small data sets. Compared with the only prior generic framework of this kind, as well as with several individually designed private hypothesis tests, our approach achieves higher power than other generic methods and is competitive with, and often outperforms, specialized alternatives.

Chapter 2: Preliminary

2.1 Basic notations

For a positive integer n, we write $[n] := 1, 2, \ldots, n$. We use $\mathbb{N}$ to denote the set of nonnegative integers, and $\mathbb{Z}$ for the set of integers. The set of real and complex numbers are denoted by $\mathbb{R}$ and $\mathbb{C}$ correspondingly. We denote by $\{0,1\}^n$ the set of binary strings of length n, and by $\{0,1\}^*$ the set of all finite binary strings. For a finite set $\mathcal{X}$, the notation $x \xleftarrow{\$} \mathcal{X}$ denotes that x is sampled uniformly at random from $\mathcal{X}$.

For a positive integer n, we denote by $\mathcal{P}_n$ the set of all permutations over $\{0,1\}^n$. When the domain is clear from context, we write $\pi \xleftarrow{\$} \mathcal{P}_n$ or $P \xleftarrow{\$} \mathcal{P}_n$ to denote a uniformly random permutation. Throughout this thesis, the symbols P, R and π are all used to denote permutations, with their intended roles being clear from context.

We use the symbol $\mathcal{O}$ to denote an oracle. An algorithm with oracle access to $\mathcal{O}$ is allowed to make adaptive queries to $\mathcal{O}$ and receive responses according to the oracle's specification. We write classical oracle access explicitly as $\mathcal{A}^{\mathcal{O}}$ and quantum oracle access as $\mathcal{A}^{|\mathcal{O}\rangle}$ when necessary, where $\mathcal{A}$ typically denotes an adversary, as determined by context.

We use standard asymptotic notation. For functions $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$, we write $f(n) = O(g(n))$ if there exist constants $c, n_0 > 0$ such that $f(n) \leq c, g(n)$ for all $n \geq n_0$. Similarly, $f(n) = \Omega(g(n))$ if there exist constants $c, n_0 > 0$ such that $f(n) \geq c, g(n)$ for all $n \geq n_0$. We write $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ hold.

2.2 Quantum computation

2.2.1 Quantum mechanics

In this section, we introduce the basic concepts, including quantum states, evolution, measurement, and entanglement. Each concept corresponds directly to one of the foundational postulates of quantum mechanics and provides the necessary background for this thesis.

Quantum States (Postulate I). The first postulate of quantum mechanics states that each isolated physical system is associated with a complex Hilbert space, and the state of the system is completely described by a unit vector in this space.

In quantum computation, the fundamental system of interest is the qubit. The state space of a single qubit is the two-dimensional Hilbert space $\mathbb{C}^2$. With respect to the computational basis $\{|0\rangle, |1\rangle\}$, which corresponds to classical bit values, a general single-qubit state can be written as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

More generally, a pure state of an n-qubit system lies in the Hilbert space $(\mathbb{C}^2)^{\otimes n}$. Its computational basis consists of states $|x\rangle$ for $x \in \{0,1\}^n$, and an arbitrary n-qubit *pure state* can be expressed as

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle ,$$

where $\alpha_x \in \mathbb{C}$ and $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

Mixed States and Density Operators. In many settings, a quantum system is not in a known pure state but rather in a probabilistic ensemble of pure states. Such a state is described by a *density operator* (or *density matrix*) ρ, which is a positive semidefinite operator with unit trace:

$$\rho \succeq 0, \qquad \mathrm{Tr}(\rho) = 1.$$

If the system is in state $|\psi_i\rangle$ with probability p_i, then its density operator is

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| .$$

Pure states are a special case of mixed states, corresponding to rank-one density operators of the form $\rho = |\psi\rangle\langle\psi|$.

Unitary Evolution and Quantum Gates (Postulate II). The second postulate of quantum mechanics states that the evolution of an isolated quantum system is described by unitary transformations. If a system is in state $|\psi\rangle$, its evolution is given by

$$|\psi\rangle \mapsto U |\psi\rangle ,$$

where U is a unitary operator, i.e., $U^\dagger U = I$. Further details will be discussed in the context of the quantum circuit model in Section 2.2.2.

Measurement (Postulate III). The third postulate states that measurements are probabilistic and are modeled by collections of operators $\{M_i\}$ satisfying

$$\sum_i M_i^\dagger M_i = I.$$

When a measurement is applied to a state $|\psi\rangle$, outcome i occurs with probability $\|M_i |\psi\rangle\|^2$, and the post-measurement state collapses accordingly.

Measurement serves to extract the classical information from a quantum system. Most quantum algorithms perform measurements in the computational basis at the end of the computation, outputting a classical random variable.

Trace Distance and Distinguishability. To compare two quantum states, one may ask how well they can be distinguished by measurement. The resulting optimal distinguishing advantage is given by the *trace distance*, which therefore serves as the standard mathematical measure of distinguishability.

For two quantum states (density operators) ρ and σ, the trace distance is defined by

$$\Delta(\rho, \sigma) := \frac{1}{2}\|\rho - \sigma\|_1,$$

where $\|X\|_1 := \mathrm{Tr}\big(\sqrt{X^\dagger X}\big)$ is the trace norm.

The trace distance has a direct operational interpretation: it characterizes the largest possible gap in acceptance probabilities over all two-outcome measurements. Concretely,

$$\max_{0 \leq E \leq I} |\mathrm{Tr}(E\rho) - \mathrm{Tr}(E\sigma)| = \Delta(\rho, \sigma),$$

where E ranges over all measurement.

Composite Systems and Entanglement (Postulate IV). The fourth postulate of quantum mechanics states that the state space of a composite system is given by the tensor product of the state spaces of its subsystems.

This tensor-product structure gives rise to entanglement, a key source of quantum computational advantage. While some states can be written as tensor products of individual subsystem states, known as product states, the tensor-product space also contains states that cannot be factorized in this way. Such non-separable states are called entangled states.

For example, the two-qubit state

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

cannot be written as a tensor product of single-qubit states. Entangled states exhibit correlations with no classical analogue and play a central role in many quantum algorithms and cryptographic protocols.

2.2.2 Quantum circuit model

The quantum circuit model is a standard framework for describing and analyzing quantum computation. It generalizes the classical Boolean circuit model by replacing classical bits and logic gates with qubits and quantum gates, allowing quantum mechanical behavior such as superposition and entanglement.

In this model, a quantum computation is described by a finite sequence of properly ordered operations acting on a fixed number of qubits. The input is placed in a designated *input register*. For a classical input string $x \in \{0,1\}^n$, the computation typically starts in the state

$$|x\rangle\,|0\rangle^{\otimes a(n)}\,,$$

where the first n qubits store the input and the remaining $a(n)$ qubits are ancillary (work) qubits initialized to $|0\rangle$. More generally, one may also allow quantum inputs. Each operation in the circuit corresponds to a quantum gate, which is mathematically represented by a unitary operator acting on one or more qubits. The evolution of the system is therefore given by the composition of these unitaries. At the end of the computation, measurements are performed on selected qubits to obtain a classical output and the outcomes are probabilistic. Unlike classical logic gates, quantum gates are generally reversible, and the entanglement between multiple qubits makes quantum states inseparable, which is essential to many quantum algorithms and can lead to potential quantum advantage.

To analyze the resources used by a quantum algorithm in the circuit model, one commonly considers:

- *Time complexity* (or *gate complexity*): the total number of elementary quantum gates in the circuit (sometimes also the circuit depth, when parallel time is relevant);

- *Space complexity*: the total number of qubits used, including input, output, and work qubits.

A quantum algorithm is typically considered *efficient* if it is implemented by a uniform family of quantum circuits using resources polynomial in the input length n (e.g., polynomially many gates and qubits).

2.2.3 Quantum query model

The quantum query model can be viewed as a specific quantum circuit model in which access to the input is provided through a fixed oracle. In this model, the complexity of an algorithm is only measured by the number of oracle queries it makes, and the cost of implementing the oracle itself is not considered. In particular, the difficulty of physically constructing the query gates is ignored.

A key feature of the quantum query model is that the query register may be in superposition, i.e., *superposition oracle access*. By linearity, if the input to the oracle $\mathcal{O}_f$ is

$$\sum_{x,y} \alpha_{x,y} |x\rangle |y\rangle ,$$

then the oracle $\mathcal{O}_f$ acts as

$$\sum_{x,y} \alpha_{x,y} |x\rangle |y\rangle \mapsto \sum_{x,y} \alpha_{x,y} |x\rangle |y \oplus f(x)\rangle .$$

Here, a single query can coherently access the values of f on many inputs at once. Classical query access is recovered as the special case in which the query register is always in a computational-basis state. Throughout this thesis, we use notation such as $\mathcal{A}^{|f\rangle}$ to denote an algorithm $\mathcal{A}$ with quantum oracle access to O_f. In particular, in Chapter 3, we write $\mathcal{A}^{|E\rangle}$ to specially denote an algorithm $\mathcal{A}$ with quantum oracle access to an ideal cipher oracle E.

Formally, a quantum query algorithm consists of a sequence of unitary operators

$$U_0, O_f, U_1, O_f, \ldots, O_f, U_T,$$

where each U_i is an input-independent unitary operation, and O_f denotes the oracle unitary corresponding to the input function f.

Unlike the classical query model, a direct query operation that outputs $f(x)$ cannot, in general, be implemented within a quantum circuit, since such an operation may not be unitary. To ensure unitarity, quantum query access is therefore provided via a reversible oracle gate. The standard oracle is defined as

$$U_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle ,$$

where $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$. See Figure 2.1 for an illustration. Thus, the query register is left unchanged, while the function value is written coherently into an auxiliary register. An equivalent formulation is the phase oracle

$$|x\rangle \mapsto (-1)^{f(x)} |x\rangle .$$

In the query model, as mentioned before, the primary resource of interest is the *query complexity*, namely, the number of oracle calls to O_f. One may also track the number of non-query gates (i.e., the cost of the U_i's) and the number of qubits used, but the purpose of the query model is to isolate the complexity of accessing the input. An algorithm is considered *query-efficient* if it makes at most poly-

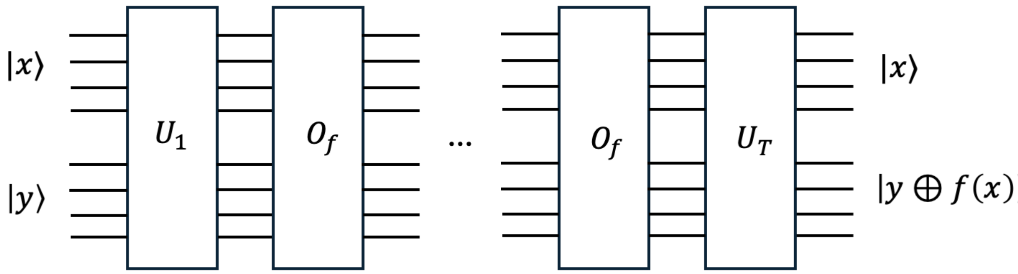

Figure 2.1: An exemplary quantum circuit with oracle access to f.

nomially many queries (in the relevant input parameter), and it is typically considered computationally efficient if both its query complexity and its non-query gate complexity are polynomial.

2.2.4 The unique search query problem

A standard problem in the quantum query model is UNIQUE-SEARCH. Let $f : \{0,1\}^n \to \{0,1\}$ be an oracle function, with the promise that either (i) there is no marked input, i.e., $f(x) = 0$ for all $x \in \{0,1\}^n$, or (ii) there exists a unique marked input $x^\star \in \{0,1\}^n$ such that $f(x^\star) = 1$ and $f(x) = 0$ for all $x \neq x^\star$. The task is to determine which of the two cases holds (decision version), or to output $x^\star$ when it exists (search version). Under the uniqueness promise, the decision and search versions are inter-reducible: search trivially solves decision, while decision solves search by bisection with only $O(n)$ overhead [Nay10,Chi17]. This problem is a canonical benchmark in query complexity: classically it requires $\Theta(2^n)$ queries in the worst case, and quantumly it can be solved with $\Theta(2^{n/2})$ queries via Grover's search (under the unique-solution promise). Moreover, these bounds are tight (up to constant factors): any bounded-error classical algorithm requires $\Omega(2^n)$ queries, and any bounded-error quantum algorithm requires $\Omega(2^{n/2})$ queries [Nay10, Gro96, Zal99].

UNIQUE-SEARCH also serves as a common source problem in reductions and lower-bound arguments, and we will use it as a basic reference point when discussing quantum query complexity.

2.3 Classical cryptography

2.3.1 Security definitions

Modern cryptography defines security via formal experiments (or games) between an adversary and a challenger. Given a scheme Π and an adversary $\mathcal{A}$, the challenger usually runs an experiment

$$\mathsf{Exp}^{\mathsf{sec}}_{\Pi,\mathcal{A}}\left(1^{\lambda}\right) \in \{0,1\},$$

where output 1 means that $\mathcal{A}$ "wins" (i.e., successfully breaks the target security notion).

Crucially, security is usually stated in terms of an adversary's *advantage* in the experiment, rather than its raw success probability. For example, in indistinguishability-based notions, an adversary can often win by random guessing with probability $1/2$, so the meaningful quantity is how much its success probability exceeds $1/2$. Formally, one defines an advantage function of the form

$$\mathsf{Adv}^{\mathsf{sec}}_{\Pi}\left(\mathcal{A},1^{\lambda}\right) \ := \ \left|\Pr\left[\mathsf{Exp}^{\mathsf{sec}}_{\Pi,\mathcal{A}}\left(1^{\lambda}\right)=1\right]-p_{\text{base}}\left(1^{\lambda}\right)\right|,$$

where $p_{\text{base}}\left(1^{\lambda}\right)$ is the success probability of a trivial strategy (e.g., $1/2$ in a binary guessing experiment). Then we say Π is secure if every efficient adversary has only negligible advantage:

$$\forall \text{ efficient adversary } \mathcal{A}, \quad \mathsf{Adv}^{\mathsf{sec}}_{\Pi}\left(\mathcal{A},1^{\lambda}\right) \le \mathsf{negl}(\lambda).$$

Here, an efficient adversary usually means one that runs in probabilistic polynomial time (PPT) with respect to the security parameter $\lambda \in \mathbb{N}$, which controls key lengths and computational resources. A function $\mathsf{negl}(\lambda)$ is negligible if for every polynomial $p(\cdot)$, there exists λ_0 such that for all $\lambda \geq \lambda_0$, $\mathsf{negl}(\lambda) < \frac{1}{p(\lambda)}$.

Experiment IND-CPA as an example. We now give the formal definition of IND-CPA security for a private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

$\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Pi}(n)$:

1. The adversary $\mathcal{A}$ is given input 1^λ and oracle access to $\mathsf{Enc}_k(\cdot)$. It outputs a pair of messages (m_0, m_1) with $|m_0| = |m_1|$.

2. A key k is generated by running $\mathsf{Gen}(1^\lambda)$, and a uniform bit $b \leftarrow \{0, 1\}$ is chosen. A challenge ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.

3. $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a bit b'.

4. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. If $\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Pi}(n) = 1$, we say that $\mathcal{A}$ succeeds.

We say that Π is $\mathsf{IND\text{-}CPA}$ secure if for every probabilistic polynomial-time (PPT) adversary $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}_{\Pi}\left(\mathcal{A}, 1^\lambda\right) \;:=\; \left|\Pr\left[\mathsf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{A},\Pi}\left(1^\lambda\right) = 1\right] - \frac{1}{2}\right| \;\leq\; \mathsf{negl}(\lambda).$$

2.3.2 Random and Pseudorandom Permutations

Let $\mathcal{P}_n$ denote the set of all permutations over $\{0, 1\}^n$. A *random permutation* $\pi \xleftarrow{\$} \mathcal{P}_n$ is sampled uniformly at random from $\mathcal{P}_n$, and defines a bijection $P : \{0, 1\}^n \to \{0, 1\}^n$. In cryptographic analyses, random permutations are typically used in idealized models and are accessed via oracle queries.

A *pseudorandom permutation* (PRP) is a family of efficiently computable keyed permutations $\mathcal{E} = \{E_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^m}$, where m is the key length and n is the block length. For every key $k \in \{0,1\}^m$, the map E_k is a permutation on $\{0,1\}^n$, and both E_k and its inverse E_k^{-1} are computable in polynomial time.

We say that $\mathcal{E}$ is a *strong pseudorandom permutation* (SPRP) if it is secure even if the distinguisher is additionally given oracle access to the inverse permutation. Formally:

Definition 2.3.1 (Strong Pseudorandom Permutation)**.** *Let $\mathcal{E} = \{E_k : \{0,1\}^n \to \{0,1\}^n\}_{k \in \{0,1\}^m}$ be an efficient family of keyed permutations. We say that $\mathcal{E}$ is an* SPRP *if for every probabilistic polynomial-time distinguisher $\mathcal{D}$, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*

$$\left| \Pr\left[\mathcal{D}^{E_k(\cdot), E_k^{-1}(\cdot)}(1^m, 1^n) = 1\right] - \Pr\left[\mathcal{D}^{P(\cdot), P^{-1}(\cdot)}(1^m, 1^n) = 1\right] \right| \leq \mathsf{negl}(m),$$

where in the first experiment $k \xleftarrow{\$} \{0,1\}^m$, and in the second experiment $P \xleftarrow{\$} \mathcal{P}_n$ is a uniformly random permutation on $\{0,1\}^n$; the probabilities are also over the internal randomness of $\mathcal{D}$.

In practice, block ciphers are designed to be secure instantiations of (strong) pseudorandom permutations with fixed key length m and block length n.

2.3.3 Reduction proofs

A central proof technique in cryptography is the reduction proof. At a high level, a reduction argues that breaking a cryptographic scheme would imply the ability to solve some underlying computational problem that is widely believed to be hard. In this sense, the security of the scheme is "reduced" to the hardness of the assumed problem.

More formally, suppose that a cryptographic scheme Π is claimed to satisfy a security property $\mathcal{S}$ under the assumption that a computational problem $\mathcal{P}$ is hard. A reduction constructs an algorithm $\mathcal{R}$ that treats any adversary $\mathcal{A}$ breaking $\mathcal{S}$ as a black-box subroutine and uses it to solve $\mathcal{P}$. If $\mathcal{P}$ is assumed to be hard for all PPT algorithms, it follows that no PPT adversary $\mathcal{A}$ can exist.

A standard argument in reduction-based proofs is to express security in terms of a *distinguishing advantage* between two experiments (or games). If G_0 and G_1 are two games and $\mathcal{A}$ outputs a bit, one defines

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{dist}}(G_0, G_1) := \left|\Pr\left[\mathcal{A}^{G_0} \to 1\right] - \Pr\left[\mathcal{A}^{G_1} \to 1\right]\right|.$$

The reduction $\mathcal{R}$ is then designed so that any non-negligible distinguishing advantage of $\mathcal{A}$ between G_0 and G_1 translates into a non-negligible advantage for $\mathcal{R}$ in solving the underlying problem $\mathcal{P}$. In other words, if $\mathcal{R}$ can simulate one game or the other depending on its own challenge instance, then $\mathcal{A}$'s output can be used as a decision procedure for $\mathcal{P}$.

When a proof proceeds through a sequence of intermediate games

$$G_0, G_1, \ldots, G_t,$$

one typically applies the triangle inequality (the *hybrid argument*) to bound the overall distinguishing advantage:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{dist}}(G_0, G_t) \leq \sum_{i=0}^{t-1} \mathsf{Adv}_{\mathcal{A}}^{\mathrm{dist}}(G_i, G_{i+1}).$$

Thus, it suffices to bound the difference between each pair of consecutive games. This hybrid method is the standard way of organizing reduction proofs, and it will be used repeatedly throughout this thesis.

Reduction-based proofs therefore provide a precise way to relate the security of cryptographic schemes to well-studied hardness assumptions, and they form the standard methodology for security proofs throughout this thesis.

2.3.4 Symmetric-key encryption

Definition 2.3.2. *A symmetric-key encryption scheme is a tuple of probabilistic efficient algorithms* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *defined as follows:*

- $\mathsf{Gen}(1^\lambda)$ *is a probabilistic algorithm that outputs a secret key* k*. We assume without loss of generality that any key* k *output by* $\mathsf{Gen}(1^\lambda)$ *satisfies* $|k| \geq \lambda$*.*
- $\mathsf{Enc}_k(m)$ *is a probabilistic algorithm that takes a key* k *and a message* m *and outputs a ciphertext* $c \leftarrow \mathsf{Enc}_k(m)$*.*
- $\mathsf{Dec}_k(c)$ *is a deterministic algorithm that takes a key* k *and a ciphertext* c *and outputs a message* $m := \mathsf{Dec}_k(c)$ *or a special symbol* $\perp$ *indicating failure.*

The scheme must satisfy correctness: for all messages m in the message space and all keys k generated by $\mathsf{Gen}(1^\lambda)$, it holds that

$$\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m.$$

Security of symmetric-key encryption is typically defined via indistinguishability-based notions. In particular, a scheme is said to be secure if no PPT adversary can distinguish encryptions of chosen messages with non-negligible advantage. Classical variants of this notion, such as security against chosen-plaintext attacks (IND-CPA introduced above) and chosen-ciphertext attacks (IND-CCA), are defined by allowing the adversary additional oracle access during the security experiment.

In practice, symmetric-key encryption schemes are usually constructed from block ciphers, which serve as the primary low-level cryptographic primitive. A block cipher is a family of keyed permutations on fixed-length bit strings, and is intended to behave like a pseudorandom permutation when the key is chosen uniformly at random. Well-known examples include AES and its variants. A useful generalization is a *tweakable block cipher* (TBC), which, in addition to the secret key and plaintext block, also takes a public *tweak* as input. Intuitively, the tweak plays the role of a non-secret parameter (such as a block index, sector number, or nonce-like value) that selects a different permutation under the same key. Thus, for each fixed key and tweak, the cipher acts as a permutation on the message block, while varying the tweak provides domain separation and helps prevent structural attacks across different blocks or contexts. To encrypt messages of arbitrary length, block ciphers are combined with additional structure through modes of operation, such as CBC, GCM, or XEX2-based constructions.

Throughout this thesis, block ciphers are usually treated as idealized or computationally secure primitives, and encryption schemes are analyzed as higher-level constructions built on top of them. Our focus is on the post-quantum security of these block-cipher-based constructions.

Chapter 3: Post-Quantum Security of Block Cipher Constructions

The first part of this thesis identifies limitations of quantum algorithms for breaking cryptography. While Shor's algorithm [Sho94] spurred intense scrutiny of the impact of quantum computers on public-key cryptography, post-quantum security of symmetric-key cryptography has received scant attention. Quantum algorithms such as Grover's search suggest generic quadratic speedups, leaving open how far these advantages extend to real-world block-cipher-based constructions [ABKM22, ABK+24].

Our analysis reveals intrinsic limitations on quantum adversaries in this setting: even quantum adversaries with some quantum oracle access to the underlying primitives do not obtain attacks substantially more powerful than those already captured by tight classical lower bounds. These results clarify where quantum methods fail to significantly weaken symmetric cryptography, establishing limits on their effectiveness.

3.1 Summary of contribution

We set the foundations for a theory of post-quantum security for block ciphers and associated constructions. Leveraging our new resampling techniques, we provide the first post-quantum security proofs for the key-length extension scheme FX, the tweakable block ciphers LRW and $\mathsf{XEX2}$, and most block cipher encryption and authentication modes. Our work takes significant initial steps in establishing a rigorous understanding of the post-quantum security of practical symmetric-key cryptography.

3.2 Background

3.2.1 Block cipher

A block cipher is a keyed family of efficiently-implementable permutations of $\{0, 1\}^n$. A block cipher π is secure if, for a uniformly random key k, the permutation π_k is indistinguishable from random to adversaries that can make forward and inverse queries to π_k. The most well-known example of a block cipher is the Advanced Encryption Standard (AES) [DBN+01], which is a fundamental ingredient in secure Internet communications and much more.

Block ciphers are ubiquitous in real-world cryptography. They are versatile ingredients that can be adapted to a variety of cryptographic applications. Their security can be increased via simple key-length extension schemes like FX [KR96, KR01]. They can also be expanded into an exponentially-large family of ciphers via tweakable constructions like LRW and XEX2 [LRW02, Rog13]. Finally, by means of *block cipher modes*, block ciphers can be used to construct a multitude of symmetric-key encryption and/or authentication schemes, with a wide range of options for both security and performance (see, e.g., [Dwo01, Dwo07, Dwo10a, GLL17]). In applications, block cipher modes are ubiquitous in secure communications, while tweakable ciphers are commonly used in disk encryption schemes.

The impact of large-scale quantum computation on the security of block ciphers and associated cryptographic constructions is not fully understood. While there are no known dramatic quantum attacks against currently-deployed symmetric-key schemes, there are important reasons to understand this setting in general (and the block cipher case in particular), as we now briefly explain.

First, it is *in principle* possible that some symmetric-key schemes in use today are not post-quantum secure[1]. Second, many symmetric-key constructions rely crucially on security proofs, and only a handful of such proofs for post-quantum security are known (mainly for Even-Mansour-type constructions [ABKM22, ABK+24, BEM24] and the Ascon scheme [Hos25]). Finally, fine-grained bounds in proofs often inform parameter selection when deploying symmetric-key schemes, and such bounds are essentially unavailable in the post-quantum case. Instead, practitioners typically assume that one can simply double the key length to avoid Grover search. However, the few results we do have indicate that this is sometimes overkill [ABKM22] and could in principle even be insufficient [BSS22].

3.2.2 This work

We set the foundations for a theory of post-quantum security for block ciphers and associated constructions. We then apply our techniques to give the first post-quantum security proofs for a variety of schemes, including key-length extension schemes, tweakable block ciphers, and block cipher modes. Our work takes significant initial steps in establishing a rigorous understanding of the post-quantum security of practical symmetric-key cryptography.

Our techniques can be used for security proofs both in the *plain model* (i.e., without any special assumptions or idealizations) as well as the *ideal cipher model* (ICM), a commonly used model in relevant classical proofs. In the ICM, one assumes that a *perfect* block cipher π was sampled (i.e., π_k is a uniformly random permutation for all k), and all parties have oracle access to it. In the traditional classical setting, this oracle access allows anyone to make queries of the form

$$(x, k) \mapsto \pi_k(x) \qquad \text{and} \qquad (x, k) \mapsto \pi_k^{-1}(x)\,. \tag{3.1}$$

[1]One can easily construct relatively natural examples of schemes that are secure against classical computers but not against quantum computers. However, we doubt that this holds for any currently-deployed schemes.

In a post-quantum setting, however, adversaries in possession of a quantum computer will be able to execute the circuits of AES in superposition. It is thus natural to also grant quantum adversaries such access in the ideal cipher setting, via black-box unitaries

$$|k\rangle\,|x\rangle\,|y\rangle \mapsto |k\rangle\,|x\rangle\,|y \oplus \pi_k(x)\rangle \qquad \text{and} \qquad |k\rangle\,|x\rangle\,|y\rangle \mapsto |k\rangle\,|x\rangle\,|y \oplus \pi_k^{-1}(x)\rangle\ . \tag{3.2}$$

The resulting model is called the quantum ideal cipher model (or QICM)[2] [CHL$^+$25, SS19]. Proving results in the QICM has turned out to be quite difficult. Classical techniques are based on transcripts and do not translate to this setting. While Zhandry's compressed oracle [Zha19] does offer a somewhat analogous quantum lower bound method, it only applies to *random function oracles*, and extending it to permutations and ideal ciphers has proved challenging. Moreover, the post-quantum setting involves mixed query types: the ideal cipher is queried both *classically* (through the construction) and *quantumly* (directly by the adversary). This is the right post-quantum model: the construction is always a classical cipher implemented by the honest party on a classical computer using knowledge of the actual secret keys, while the block cipher is a public algorithm. Moreover, correct modeling of the construction oracle makes a crucial difference in security, as many commonly-used ciphers are *insecure* if the construction can be queried quantumly [CHL$^+$25], but *secure* if the construction can only be queried classically [ABKM22].

Here, we only consider the model in which the ideal cipher is quantumly-accessible and any constructions involving the secret key are only classically-accessible. We refer to this model as the post-quantum model. In previous literature, this model was sometimes referred to as the $\mathsf{Q1}$ model [BHNP$^+$19, HS18, JST21, CHL$^+$25], in contrast with the (unrealistic) $\mathsf{Q2}$ model where all ora-

[2] The ICM (resp., QICM) is the natural block cipher analogue of the well-known Random Oracle Model (ROM, resp., QROM), which is often used to model hash functions in security proofs.

cles can be queried quantumly.

Our results are summarized as follows. We give a more technical summary further below, including the precise security bounds we establish.

1. **QICM resampling.** We give a general *resampling lemma* that can be used to ascertain the ability of a quantum adversary to detect modifications to the ideal cipher (e.g., in a simulation as part of a proof). Previous resampling lemmas only held for random functions and permutations [ABKM22, ABK+24, GHHM21, Hos25]. This new tool can be combined with an adaptation of a proof approach of [ABKM22] to yield a powerful technique for QICM security proofs. We apply this technique to establish further results below.

2. **FX construction.** We prove post-quantum security of the FX key-length extension scheme in the QICM. This problem has evaded analysis for a number of years (a 2021 result only held for non-adaptive adversaries [JST21]). The bound we give for the number of queries needed to achieve a constant distinguishing probability is tight. A straightforward application of our result establishes post-quantum security for the lightweight cipher PRINCE [BCG+12].

3. **Tweakable ciphers.** We prove security of two widely-used tweakable block ciphers (LRW and XEX2) in both the plain model and in the QICM (with different bounds). We note that XEX2 is the basis of the XTS-AES disk encryption scheme used by most operating systems.

4. **Block cipher modes.** Finally, we observe that the security proofs of most block cipher modes (including all modes used in secure Internet traffic) translate easily to the post-quantum setting. Moreover, one can translate classical security bounds to post-quantum ones simply by replacing the appropriate strong pseudorandomness advantage term with its post-quantum analogue.

3.2.3 Technical summary of results

A technical summary of our main results is as follows.

A proof technique for the QICM. The first proof of post-quantum security of a block cipher construction was for the plain Even-Mansour construction [ABKM22]. Since then, the technique of [ABKM22] has been extended to show security of tweakable Even-Mansour [ABK+24], the Ascon lightweight cipher [Hos25], key-alternating ciphers [BEM24, BBC+25]. The technique has even been adapted to give certain proofs in the quantum Haar-random oracle model [HY24].

At a high level, the technique of [ABKM22] can be described as follows. The goal is to show indistinguishability between (i.) a pair of "real" oracles $(E_{(k)}, |E\rangle)$ and (ii.) an "ideal" uncorrelated pair $(R, |E\rangle)$. In both worlds, the first oracle is classical and the second oracle is quantum. Starting with a real-world q-query sequence

$$|E\rangle, E_{(k)}, |E\rangle, E_{(k)}, \dots, |E\rangle, E_{(k)} \tag{3.3}$$

we switch the classical queries to the ideal case, one-by-one. A naive j-th hybrid would then be

$$\underbrace{|E\rangle, R, \dots, |E\rangle, R}_{j}, \underbrace{|E\rangle, E_{(k)}, \dots, |E\rangle, E_{(k)}}_{q-j}. \tag{3.4}$$

Unfortunately, this does not work: showing indistinguishability of hybrids $j-1$ and j would require first showing that the adversary cannot extract the key k during the last $q-j$ queries; indeed, with knowledge of k one easily notices the switch in past queries. This leads to a circular argument.

Instead, when switching classical queries to the ideal case, we also modify E (for all remaining queries) in a small number of locations, and then reset these modifications at a later stage:

$$\underbrace{|E\rangle, R, \ldots, |E\rangle, R}_{j}, \underbrace{|E^{[j]}\rangle, E^{[j]}_{(k)}, \ldots, |E^{[j]}\rangle, E^{[j]}_{(k)}}_{q-j}. \tag{3.5}$$

where $E^{[j]}$ denotes the quantum oracle with roughly j modified locations. Making such modifications necessitates the use of a particular technical ingredient: a *resampling lemma*[3]. Such a lemma states that an adversary cannot detect such modifications unless it has made a very large number of quantum queries to E prior to the modification being made.

Ideal cipher resampling. We develop the above technique in the case where E is an ideal cipher. This enables us to give the first full post-quantum security proofs for a variety of block cipher constructions. The key technical advance is a resampling lemma for the ideal cipher model. Consider the advantage of a computationally unbounded distinguisher $\mathcal{D}$ in the following three-phase experiment.

1. An ideal cipher $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ is sampled, and $\mathcal{D}$ gets (two-way) quantum-query access to E. Then $\mathcal{D}$ (adaptively) selects and outputs a description of a distribution M.

2. A sample (k, s_0, s_1) is drawn from M, and $E_k(s_0)$ and $E_k(s_1)$ are swapped ($b = 1$) or not ($b = 0$).

3. Now $\mathcal{D}$ continues with access to (the possibly modified) E, and must eventually output a guess b' for the value of b.

We show that the guessing advantage of $\mathcal{D}$ is at most $4\sqrt{q \cdot 2^{n-h}}$ where h is the min-entropy of M and q is the number of queries made by $\mathcal{D}$ *in phase 1*. Note that the bound does not depend on the number

[3] One also needs a *reprogramming lemma* (to reset these modifications at a later stage) but this is a relatively straightforward quantum search lower bound [ABKM22].

of queries made in phase 3.

The full technical statement of this resampling lemma is given in Section 3.4, with proofs in Section 3.8.1. Using this lemma and the hybrid technique described above, we are able to show the security of several constructions, discussed below.

FX construction. The FX construction [KR96, KR01] transforms a block cipher $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ with key length m into another block cipher with the same block size but a longer key, as follows.

$$\mathsf{FX}^E_{k_0,k_1,k_2}(x) = E_{k_0}(x \oplus k_1) \oplus k_2 \tag{3.6}$$

Here $k_0 \leftarrow \{0,1\}^m$ and $k_1, k_2 \leftarrow \{0,1\}^n$. For post-quantum security in the QICM, the relevant quantity is the distinguishing advantage

$$\mathsf{Adv}^{\mathsf{FX}}_{\mathcal{A}} := \left| \Pr_{E,k_0,k_1,k_2}\left[\mathcal{A}^{\mathsf{FX}[E],|E\rangle} = 1\right] - \Pr_{R,E}\left[\mathcal{A}^{R,|E\rangle} = 1\right] \right| \tag{3.7}$$

of an (unbounded) adversary $\mathcal{A}$ in distinguishing FX from a random permutation R using q_C classical queries, while making q_Q quantum queries to E. Both forward and inverse queries are allowed. In Section 3.5, we show that

$$\mathsf{Adv}^{\mathsf{FX}}_{\mathcal{A}} \le 4(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}) \cdot 2^{-(m+n)/2} \qquad \text{for } q_C \ll 2^n \tag{3.8}$$

$$\mathsf{Adv}^{\mathsf{FX}}_{\mathcal{A}} \le q_Q^2 \cdot 2^{-m} \qquad \text{for } q_C \approx 2^n \tag{3.9}$$

For calculating the number of queries required for constant success probability, our bounds are tight. For small q_C as well as $q_C \approx 2^n$ (i.e., the full codebook case), Grover search for remaining keys matches the bound. For $q_C \approx 2^{n/3}$, BHT collision-finding [KM12] also matches our bound.

In contrast, in the classical ideal cipher model, the FX construction is tightly secure (i.e., indistinguishable from an independent uniformly random permutation) with distinguishing advantage [KR01]

$$\mathsf{Adv}^{\mathsf{FX}}_{\mathcal{A}} \leq \frac{q_{\mathsf{FX}} \cdot q_E}{2^{m+n}}. \tag{3.10}$$

against an adversary making q_{FX} queries to FX_K and q_E queries to the ideal cipher E.

Applications to lightweight ciphers. A relatively straightforward application of our lower bound for FX establishes post-quantum security of the PRINCE [BCG+12] cipher in the QICM. The security of PRINCE is derived from the security of a primitive $\widetilde{\mathsf{FX}}$ which is closely related to standard FX. Specifically, let H be an $(m-1)$-dimensional subspace of $\mathbb{F}_2$, choose nonzero $\alpha \in \mathbb{F}_2^m$ not in H, and define

$$\widetilde{\mathsf{FX}}_{k_0,k_1,k_2}(x) \;=\; \begin{cases} \mathsf{FX}_{k_0,k_1,k_2}(x), & \text{if } k_0 \in H, \\ \mathsf{FX}^{-1}_{k_0 \oplus \alpha,\, k_1,\, k_2}(x), & \text{if } k_0 \in \alpha \oplus H. \end{cases}$$

The post-quantum security of $\widetilde{\mathsf{FX}}$ then follows directly from the same security for FX.

Tweakable ciphers. As a second major application of our technique for the QICM, we establish post-quantum security of two tweakable block ciphers: LRW and $\mathsf{XEX2}$ in Section 3.6. Tweakable block ciphers like LRW have a variety of applications, e.g., to modes for authentication and authenticated encryption [Rog04]. The $\mathsf{XEX2}$ cipher forms the core of the widely-used and standardized XTS-AES disk encryption scheme [Dwo10b, P1606].

The LRW construction is defined by

$$\mathsf{LRW}^{E,h}_{k,k'}(\tau, x) = E_k(x \oplus h_{k'}(\tau)) \oplus h_{k'}(\tau)\,. \tag{3.11}$$

Here, τ is a tweak parameter, h is an XOR-universal hash function, and E is a block cipher. $\mathsf{XEX2}$ is defined similarly, except that (roughly speaking) E'_k (for independent k') is used in place of h. Our post-quantum bound for distinguishing LRW from an ideal tweakable cipher is similar to that for FX, but with an additional additive term of $6q_C^2 \cdot 2^{-n}$. This corresponds to a (purely classical) collision attack that is possible against LRW but not against FX. For $\mathsf{XEX2}$, our bound is $q_Q^2 \cdot 2^{-m} + 3q_C^2 \cdot 2^{-n}$, and is derived from Theorem 3.2.1 below.

In the classical setting, LRW achieves birthday-bound security. In the ideal cipher model, its distinguishing advantage is bounded by approximately $2^{-n/2} + 2^{-m/2}$. Prior work [LRW02, JN20] establishes that any classical adversary requires $q = O(2^{n/2})$ queries to distinguish LRW from an ideal tweakable cipher, and this bound is tight due to the existence of matching collision-based attacks. Similarly, the standardized $\mathsf{XEX2}$ construction used in $\mathsf{XTS\text{-}AES}$ [P1619] has been extensively analyzed [CSR+08, LM08, Rog13], exhibiting comparable birthday-bound security in the classical model.

In the post-quantum setting, our additional term $q_C^2 \cdot 2^{-n}$ in the bound corresponds directly to the classical collision attack that also limits LRW 's classical security. The remaining terms in our bound arise from the inherent quantum speedups in key search and collision finding.

Block cipher modes. In Section 3.7, we give the following general lifting theorem for establishing post-quantum security of block cipher modes.

Theorem 3.2.1 (informal)**.** *Let Exp be a security experiment and Con a construction instantiated with* *a block cipher E. Then the post-quantum security of* Con *is bounded as follows:*

$$\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E]}(q,t) \leq \mathsf{Adv}^{\mathsf{SPRP\text{-}PQ}}_{E}(q',t) + \delta(q), \tag{3.12}$$

where q' is the number of E-queries made by Con *and the challenger, (q,t) denotes the (query, time)-complexity of the quantum adversary, and $\delta(q)$ is the classical information-theoretic security of* Con*.*

While this lifting theorem is straightforward to prove, it offers a useful and rather general observation for establishing post-quantum security for a wide variety of block cipher modes, including CBC, ECBC, CMAC, GCM, and GCM-SST. It can also be applied in the QICM, with the SPRP advantage term becoming $(q')^2/2^m$, corresponding to a lower bound for quantum key search. This also yields lower bounds for LRW and XEX2, although, in these cases, better bounds are available using the hybrid technique discussed above. Some lower bounds computed via Theorem 3.2.1 are given in Table 3.2.

We remark that, while Theorem 3.2.1 can be applied to FX, LRW, and XEX2 in the ideal cipher model, the specialized bounds we establish using the hybrid technique are far better.

3.2.4 Related work

We focus on related work concerning post-quantum (i.e, Q1-model) security of keyed symmetric ciphers. We first discuss lower bounds, and then attacks. Since our results are query lower bounds, we will account separately for offline computation time and queries to the "offline primitive" (e.g., a public random permutation). Some cryptanalytic works do not make this distinction.

Alagic et al. [ABKM22] established a tight lower bound for Even-Mansour using the hybrid technique discussed above. These methods were then extended to prove security of tweakable Even-

Mansour [ABK+24]. Recently, these techniques were also used to prove results about the security of key-alternating Feistel [BBC+25], multi-round Even-Mansour [BEM24], and Ascon [Hos25].

All ciphers under consideration can be attacked by first making (a small number of) classical queries, and then performing Grover search on the offline primitive for the full key. Kuwakado and Morii observed that BHT collision-finding [BHT97] can be used to perform key recovery for Even-Mansour using $O(2^{n/3})$ classical and quantum queries (and the same amount of time and space) [KM12]. Bonnetain et al. improved on this with the offline Simon algorithm, achieving the same query complexity but only using polynomial quantum space [BHNP+19].

The offline-Simon algorithm also applies to key recovery on the FX, demanding $O(2^{(m+n)/3})$ classical queries while maintaining $\mathsf{poly}(n)$ classical memory. Additionally, the earlier meet-in-the-middle attack [HS18] on the FX requires $O(2^{3(m+n)/7})$ classical queries and $O(2^{(m+n)/7})$ classical memory. Bonnetain et al. [BSS22] propose the first general quantum key-recovery attack in the post-quantum setting on a symmetric block cipher, offering a super quadratic speedup compared to the best classical attacks. Their work extends the offline-Simon algorithm to attack the $\mathsf{2XOR}$ Cascade construction [GT12], achieving a $2.5\times$ quantum speedup in the exponent over the best-known classical attack.

3.3 Technical preliminaries

Notations and definitions. Sampling an element s uniformly at random from a set S is denoted by $s \leftarrow S$. We let $\mathcal{P}(n)$ denote the set of all permutations on $\{0,1\}^n$. A block cipher $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ is a keyed permutation, i.e., $E_k(\cdot) = E(k, \cdot)$ is a permutation of $\{0,1\}^n$ for all

$k \in \{0,1\}^m$. We also define the *swap operator* $\mathsf{swap}_{a,b} : \mathcal{X} \to \mathcal{X}$ as

$$\mathsf{swap}_{a,b}(x) = \begin{cases} a, & \text{if } x = b, \\ b, & \text{if } x = a, \\ x, & \text{otherwise.} \end{cases} \tag{3.13}$$

That is, $\mathsf{swap}_{a,b}$ exchanges a and b and leaves all other elements unchanged.

For an oracle $\mathcal{O}$, we write $\pm\mathcal{O}$ to denote two-directional access, i.e., the adversary gets access to both $\mathcal{O}$ and $\mathcal{O}^{-1}$. Given a function $F : \{0,1\}^t \to \{0,1\}^\ell$, we let $|F\rangle$ denote the appropriate quantum oracle $|x\rangle\,|y\rangle \to |x\rangle\,|y \oplus F(x)\rangle$ for making quantum queries to F. So, for example, the notation $\mathcal{A}^{F,|\pm P\rangle}$ denotes an algorithm $\mathcal{A}$ that can make classical queries to a function F (forward only) and quantum queries to a permutation P (in both the forward and inverse direction).

A reprogramming lemma. For a function $F : \{0,1\}^\ell \to \{0,1\}^n$ and a set $B \subset \{0,1\}^\ell \times \{0,1\}^n$ such that each $x \in \{0,1\}^\ell$ is the first element of at most one tuple in B, define

$$F^{(B)}(x) := \begin{cases} y & \text{if } (x,y) \in B \\ F(x) & \text{otherwise.} \end{cases} \tag{3.14}$$

The following is taken verbatim from [ABKM22, Lemma 3]:

Lemma 3.3.1 (Reprogramming Lemma)**.** *Let $\mathcal{D}$ be a quantum distinguisher in the following experiment:*

Phase 1: *$\mathcal{D}$ outputs descriptions of a function $F_0 = F : \{0,1\}^\ell \to \{0,1\}^n$ and a randomized algo-*

rithm $\mathcal{B}$ whose output is a set $B \subset \{0,1\}^\ell \times \{0,1\}^n$ where each $x \in \{0,1\}^\ell$ is the first element of at most one tuple in B. Let $B_1 = \{x \mid \exists y : (x,y) \in B\}$ and $\epsilon = \max_{x \in \{0,1\}^\ell} \{\Pr_{B \leftarrow \mathcal{B}}[x \in B_1]\}$.

Phase 2: *$\mathcal{B}$ is run to obtain B. Let $F_1 = F^{(B)}$. A uniform bit b is chosen, and $\mathcal{D}$ is given quantum access to F_b.*

Phase 3: *$\mathcal{D}$ loses access to F_b, and receives the randomness r used to invoke $\mathcal{B}$ in phase 2. Then $\mathcal{D}$ outputs a guess b'.*

For any $\mathcal{D}$ making q queries in expectation when its oracle is F_0, it holds that

$$|\Pr[\mathcal{D} \textit{ outputs 1} \mid b=1] - \Pr[\mathcal{D} \textit{ outputs 1} \mid b=0]| \leq 2q \cdot \sqrt{\epsilon}\,. \tag{3.15}$$

3.4 Ideal cipher resampling

The hybrid technique described in Section 3.2.3 is rather general, and could in principle be applied to a wide variety of settings. One particular setting of interest is the Quantum Ideal Cipher Model, where the oracle $|E\rangle$ gives quantum oracle access to the ideal cipher, and the oracle $E_{(k)}$ gives classical oracle access to some construction based on E (e.g., a tweakable cipher.) The main technical ingredient that is then required is an appropriate resampling lemma. We now briefly introduce the Quantum Ideal Cipher Model (QICM) and state our new resampling lemma for that model.

Quantum Ideal Cipher Model. The QICM was first discussed in [HS18]. Define $\mathcal{E}(m,n)$ to be the space of all keyed permutations where the key $k \in \{0,1\}^m$ and the input $x \in \{0,1\}^n$. In the quantum ideal cipher model, we treat the underlying block cipher as an ideal cipher $E \leftarrow \mathcal{E}(m,n)$, meaning that for each key k, the function E_k is a random permutation. Additionally, adversaries have quantum-computational capabilities in this model, and are thus allowed to make both forward and backward quantum queries to the ideal cipher.

In the security notions mentioned above, we consider algorithms having only classical access to secretly keyed primitives. When we consider constructions of keyed primitives (e.g., a tweakable block cipher) from public primitives (e.g., a random permutation), however, we provide the distinguisher with *quantum* oracle access to the public primitive. Thus, for example, a quantum distinguisher in the QICM can apply the unitary operators

$$|k\rangle\,|x\rangle\,|y\rangle \mapsto |k\rangle\,|x\rangle\,|E_k(x) \oplus y\rangle \tag{3.16}$$

$$|k\rangle\,|x\rangle\,|y\rangle \mapsto |k\rangle\,|x\rangle\,|E_k^{-1}(x) \oplus y\rangle \tag{3.17}$$

to quantum registers of the adversary's choice. We will denote an algorithm $\mathcal{A}$ with such access to a block cipher E by $\mathcal{A}^{|\pm E\rangle}$.

A resampling lemma for the QICM. We first give a resampling lemma for the ideal cipher model, generalizing a lemma of Hosoyamada [Hos25, Lemma 3]. Our generalization differs from [Hos25] in that it works in the QICM rather than the random permutation model, and in that it allows an arbitrary (distinguisher-chosen) distribution of the key and resampling points, rather than restricting to uniform sampling.

Lemma 3.4.1. *Let $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ be a quantum distinguisher interacting with the following experiment:*

Phase 1: *Choose a uniform ideal cipher $E \in \mathcal{E}(m, n)$ and give $\mathcal{D}$ quantum access to E and E^{-1}. Then $\mathcal{D}_0$ outputs a distribution D over $\{0,1\}^{m+2n}$.*

Phase 2: *Sample* $(k_0, s_0, s_1) \in \{0,1\}^m \times \{0,1\}^n \times \{0,1\}^n$ *according to* D*. Define* $E^{(0)} = E$ *and* $E^{(1)}$ *as*

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x), & \text{if } k^* \neq k_0, \\ \\ E_{k^*} \circ \mathsf{swap}_{s_0, s_1}(x), & \text{if } k^* = k_0, \end{cases} \tag{3.18}$$

where swap_{s_0, s_1} *is the transposition swapping* s_0 *and* s_1*. A uniform bit* $b \in \{0,1\}$ *is chosen, and* $\mathcal{D}$ *is given* k_0, s_0, s_1 *along with quantum access to* $E^{(b)}$*. Finally,* $\mathcal{D}$ *outputs a guess* b'*.*

For a distribution D *on* $\{0,1\}^{m+2n}$*, define*

$$\epsilon \;=\; \max_{(k_0^*, s_0^*, s_1^*) \in \{0,1\}^{m+2n}} D(k_0^*, s_0^*, s_1^*)\,. \tag{3.19}$$

Then for any distinguisher $\mathcal{D}$ *making at most* q *queries to* E *in Phase 1, it holds that*

$$|\Pr[\mathcal{D} \textit{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \textit{ outputs } 1 \mid b = 0]| \;\leq\; 4\sqrt{2^n \cdot q \cdot \epsilon}\,. \tag{3.20}$$

The proof follows the approach of [ABKM22, Hos25], which proceeds by bounding the trace distance between the quantum states produced after Phase 1 in the original case ($b = 0$) and the reprogrammed case ($b = 1$). Crucially, the distinguishing advantage depends only on the queries made in Phase 1: the swap operation itself does not alter the distribution of E globally, and so unless the adversary has queried the resampled points during Phase 1, the two cases are identically distributed. The proof of Lemma 3.4.1 is given in Section 3.8.1.

3.5 Post-quantum security of FX

3.5.1 The construction

The FX key-length extension construction was analyzed in [KR96, KR01]. It is defined as follows.

Definition 3.5.1 (FX construction)**.** *Let m and n be positive integers. Let $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. The FX construction is defined as*

$$\mathsf{FX}_K^E(x) = E_{k_0}(x \oplus k_1) \oplus k_2\,, \tag{3.21}$$

where $K = (k_0, k_1, k_2)$ with $k_0 \leftarrow \{0,1\}^m$ and $k_1, k_2 \leftarrow \{0,1\}^n$.

3.5.2 Post-quantum security proof

We now state and prove a post-quantum security bound for FX. As in the classical case discussed above, we are concerned with the maximum distinguishing advantage between FX and an independent uniformly random permutation in the ideal-cipher model.

Theorem 3.5.2. *Let $\mathcal{A}$ be an adversary making q_C classical queries to its first oracle and q_Q quantum queries to its second oracle. Then*

$$\left|\Pr_{\substack{(k_0,k_1,k_2)\leftarrow\{0,1\}^{m+2n}\\ E\leftarrow\mathcal{E}(m,n)}}\left[\mathcal{A}^{\pm\mathsf{FX}_K[E],|\pm E\rangle}=1\right] - \Pr_{\substack{R\leftarrow\mathcal{P}(n);\\ E\leftarrow\mathcal{E}(m,n)}}\left[\mathcal{A}^{\pm R,|\pm E\rangle}=1\right]\right|$$
$$\leq \frac{4}{\sqrt{2^m(2^n-q_C+1)}}\cdot q_C\sqrt{q_Q} + \frac{2}{\sqrt{2^{m+n}}}q_Q\sqrt{q_C}.$$

Proof. For a general adversary, whether a particular classical query is made in the forward or inverse direction can be chosen in an arbitrary, adaptive manner. Without loss of generality, we consider adversaries who fix this order in advance; this incurs a factor 2 cost in the total number of queries. Our proof will proceed using a hybrid-by-classical-queries approach. We will now give the proof in the case where every classical query is in the forward direction. In Section 3.8.2, we describe how to handle hybrid transitions for the case where the relevant classical query is in the inverse direction.

We begin by setting down a way of modifying a given cipher based on a choice of key and a list of classical queries to an FX oracle. Let $E \in \mathcal{E}(m,n)$, $K = (k_0, k_1, k_2) \in \{0,1\}^{m+2n}$, and fix a list $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{q_C}, y_{q_C})\}$. Each pair (x_i, y_i) represents an input–output pair of a classical query. Repeated queries are not allowed, i.e., $x_i \neq x_{i'}$ for all $i \neq i'$. For any $j \leq q_C$, let T_j be the list containing the first j queries.

The "modified cipher" after j-many queries is denoted $E^{T_j,K}$, and is given by

$$E^{T_j,K}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k_0 \\ E^{T_j,K}_{k_0}(x) & \text{if } k^* = k_0. \end{cases} \tag{3.22}$$

where $E^{T_j,K}_{k_0}$ is defined as follows. First, define $E^{T_0,K} = E$, and for all $j \in [1, q_C]$,

$$E^{T_j,K}_{k_0}(x) = \mathsf{swap}_{E^{T_{j-1},K}_{k_0}(x_j \oplus k_1),\, y_j \oplus k_2} \circ E^{T_{j-1},K}_{k_0}(x)\,. \tag{3.23}$$

Note that the above definition differs from prior work [ABKM22, ABK+24, BBC+25]. Specifically, $E^{T_i,K}$ is now constructed recursively based on $E^{T_{i-1},K}$, whereas in previous work, it was not. Roughly speaking, $E^{T_j,K}$ denotes a slight modification of E that remains consistent with the transcript

T_j, as stated in Proposition 1, with the proof provided in Section 3.8.2.

Proposition 1. *For any $E \in \mathcal{E}(m,n)$, $K = (k_0, k_1, k_2) \in \{0,1\}^{m+2n}$, $j \in \{1,\ldots,q_C\}$, transcript $T_j = \{(x_1,y_1),\ldots,(x_j,y_j)\}$ without repetition, and any $i \in \{1,\ldots,j\}$, it holds that*

$$E_{k_0}^{T_j,K}(x_i \oplus k_1) \oplus k_2 = y_i.$$

For compactness, we occasionally write E^j in place of $E^{T_j,K}$ when T_j and K are understood from the context. We also set $E^0 = E$.

We now define a sequence

$$\mathbf{H}_0, \mathbf{H}_0^1, \mathbf{H}_0^2, \mathbf{H}_0^3, \mathbf{H}_1, \mathbf{H}_1^1, \ldots, \mathbf{H}_{q_C}^3 \tag{3.24}$$

of hybrid experiments. Each experiment begins with sampling uniform $R \in \mathcal{P}(n)$ and $E \in \mathcal{E}(m,n)$, and a uniform key $K = (k_0, k_1, k_2) \in \{0,1\}^{m+2n}$. The remaining steps of each hybrid are as follows.

Experiment $\mathbf{H}_j$.

1. $\mathbf{H}_j$ begins by running $\mathcal{A}$, answering its classical queries using R and quantum queries using E, stopping immediately *before* its $(j+1)^{\text{st}}$ classical query. Let T_j be the list of classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries by $\mathsf{FX}_K[E^{T_j,K}]$ and its quantum queries by $E^{T_j,K}$.

Experiment $\mathbf{H}_j^1$.

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using E, until $\mathcal{A}$ makes

its $(j+1)^{\text{st}}$ classical query, which we assume to be in the forward direction. Let T_j be the list of classical queries so far.

2. Define set $S = \{0,1\}^n \backslash \{x_1 \oplus k_1, \cdots, x_j \oplus k_1\}$. Choose uniform $s \in S$, and define $E^{(1)}$ as

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k_0 \\ \left(E_{k_0} \circ \mathsf{swap}_{k_1 \oplus x_{j+1},\, s}\right)(x) & \text{if } k^* = k_0. \end{cases}$$

Continue running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{FX}_K[(E^{(1)})^{T_j,K}]$, and its quantum queries using $(E^{(1)})^{T_j,K}$.

Experiment $\mathbf{H}_j^2$.

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using E, until $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query, which we assume to be in the forward direction. Let T_{j+1} be the list of classical queries so far, and T_j the subset consisting of the first j queries.

2. Answer query $j+1$ as in $\mathbf{H}_j^1$, i.e., with $(j+1)^{\text{st}}$ query using $y_{j+1} := \mathsf{FX}_K[(E^{(1)})^{T_j,K}](x_{j+1})$. Then construct $E^{j+1} \equiv E^{T_{j+1},K}$ as defined adaptively as in Equation 3.22. Continue running $\mathcal{A}$, answering its remaining classical queries using $\mathsf{FX}_K[E^{T_{j+1},K}]$, and its quantum queries using $E^{T_{j+1},K}$.

Experiment $\mathbf{H}_j^3$.

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using E, stopping immediately *after* its $(j+1)^{\text{st}}$ classical query. Let T_{j+1} be the set of classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $\mathsf{FX}_K\left[E^{T_{j+1},K}\right]$ and its quantum queries using $E^{T_{j+1},K}$, i.e. $|E^{j+1}\rangle$.

We can compactly represent the hybrids $\left\{\mathbf{H}_j, \mathbf{H}_j^1, \mathbf{H}_j^2, \mathbf{H}_j^3, \mathbf{H}_{j+1}\right\}$ as the experiments in which $\mathcal{A}$'s queries are answered using the following oracle sequences. Let $(E^{(1)})^j$ denote $(E_{k_0}^{(1)})^{T_j,K}$.

$$\begin{aligned}
\mathbf{H}_j :&\ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[\ \ E^j\ \ ], |E^j\rangle \qquad , \mathsf{FX}_K\left[E^j\right], |E^j\rangle, \cdots \\
\mathbf{H}_j^1 :&\ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle\ , \mathsf{FX}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots \\
\mathbf{H}_j^2 :&\ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |E^{j+1}\rangle \quad , \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}_j^3 :&\ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \qquad R \qquad , |E^{j+1}\rangle \quad , \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}_{j+1} :&\ \ \underbrace{|E\rangle, R, |E\rangle, \cdots, R, |E\rangle}_{j \text{ classical queries}}, \underbrace{\quad R \qquad , |E\rangle}_{(j+1)^{\text{st}} \text{ classical query}}, \underbrace{\mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots}_{q_C - j - 1 \text{ classical queries}}.
\end{aligned}$$

Then we establish the following bounds on the distinguishability of $\mathbf{H}_j$ and $\mathbf{H}_{j+1}$, step by step, for $0 \leq j < q_C$:

Lemma 3.5.3: $\quad |\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1\right]| \leq 4\sqrt{\dfrac{q_Q}{2^m(2^n - j)}}$

Lemma 3.5.4: $\quad \mathbf{H}_j^1 = \mathbf{H}_j^2$

Lemma 3.5.5: $\quad \mathbf{H}_j^2 = \mathbf{H}_j^3$

Lemma 3.5.6: $\quad |\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{Q,j+1}\sqrt{\dfrac{2(j+1)}{2^{m+n}}}$,

where $q_{Q,j+1}$ is the expected number of queries $\mathcal{A}$ makes to P in the $(j+1)^{\text{st}}$ stage, i.e., the stage between the $(j+1)^{\text{st}}$ and $(j+2)^{\text{nd}}$ classical queries.

Using the above, we have

$$
\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_0)=1]-\Pr[\mathcal{A}(\mathbf{H}_{q_C})=1]| \\
&\le \sum_{j=0}^{q_C-1}\left(4\sqrt{\frac{q_Q}{2^m(2^n-j)}}+2\cdot q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{m+n}}}\right) \\
&\le \sum_{j=0}^{q_C-1}\left(4\sqrt{\frac{q_Q}{2^m(2^n-j)}}+2\cdot q_{Q,j+1}\sqrt{\frac{2q_C}{2^{m+n}}}\right) \\
&\le \frac{4}{\sqrt{2^m(2^n-q_C+1)}}\cdot q_C\sqrt{q_Q}+\frac{2}{\sqrt{2^{m+n}}}q_Q\sqrt{q_C}.
\end{aligned} \tag{3.25}
$$

$\square$

Remark. When $q_C < \frac{3}{4}2^n$, our bound can be simplified to

$$
\frac{8}{\sqrt{2^{m+n}}}\left(q_C\sqrt{q_Q}+q_Q\sqrt{q_C}\right).
$$

As q_C approaches 2^n (in particular when $q_C = 2^n$), the bound above is no longer tight. In this case, as explained in Section 3.7.2, the distinguishing problem between (R, E) and (FX_K, E) is equivalent to the $\mathsf{UNIQUE\text{-}SEARCH}$ problem on k_0. By Corollary 3.7.7, this yields an advantage of $\frac{q_Q^2}{2^m}$.

We now prove Lemma 3.5.3, Lemma 3.5.4, Lemma 3.5.5 and Lemma 3.5.6.

Lemma 3.5.3. *For $j = 0, \ldots, q_C$,*

$$
\left|\Pr[\mathcal{A}(\mathbf{H}_j)=1]-\Pr\left[\mathcal{A}(\mathbf{H}_j^1)=1\right]\right| \le 4\sqrt{\frac{q_Q}{2^m(2^n-j)}}.
$$

Proof. In this lemma, we bound the distinguishability of $\mathbf{H}_j$ and $\mathbf{H}_j^1$.

$$\mathbf{H}_j: \ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[\ \ E^j\ \ ], |E^j\rangle \qquad , \mathsf{FX}_K\left[E^j\right], |E^j\rangle, \cdots$$

$$\mathbf{H}_j^1: \ \ |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle \ , \mathsf{FX}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots.$$

Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j$ and $\mathbf{H}_j^1$. We construct from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for the resampling experiment of Lemma 3.4.1. $\mathcal{D}$ does:

Phase 1: $\mathcal{D}$ is given quantum access to an ideal cipher E. It samples a uniform $R \leftarrow \mathcal{P}_n$ and then runs $\mathcal{A}$, answering its quantum queries with E and its classical queries with R (in the appropriate directions), until $\mathcal{A}$ submits its $(j+1)^{\text{st}}$ classical query x_{j+1}. At that point, $\mathcal{D}$ has a list $T_j = \{(x_1, y_1), \cdots, (x_j, y_j)\}$ of the queries/answers $\mathcal{A}$ has made to its classical oracle thus far. Next, $\mathcal{D}$ constructs a distribution D on $\{0,1\}^{m+2n}$ and its sampling algorithm Π. To sample a tuple $(a_0, z_0, z_1) \leftarrow D$, Π does the following:

1 : Sample uniform $a_0 \in \{0,1\}^m$ and $z_0 \in \{0,1\}^n$.

2 : Construct $S = \{0,1\}^n \backslash \{z_0 \oplus x_1 \oplus x_{j+1}, \cdots, z_0 \oplus x_j \oplus x_{j+1}\}$.

3 : Sample uniform $z_1 \in S$, and output (a_0, z_0, z_1).

Phase 2: $\mathcal{D}$ is given $(k_0, s_0, s_1) \leftarrow D$ and quantum oracle access to a cipher $E^{(b)}$. Then $\mathcal{D}$ sets $k_1 = x_{j+1} \oplus s_0$, uniform $k_2 \in \{0,1\}^n$ and $K = (k_0, k_1, k_2)$. It then continues running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{FX}_K[(E^{(b)})^{T_j, K}]$, and its remaining quantum queries using $(E^{(b)})^{T_j, K}$. $\mathcal{D}$ outputs whatever $\mathcal{A}$ does.

Defining $S^{\perp} = \{x_1 \oplus k_1, \cdots x_j \oplus k_1\}$, we have $s_1 \in \{0,1\}^n \setminus S^{\perp}$ from algorithm Π. In phase 1, distinguisher $\mathcal{D}$ perfectly simulates experiments $\mathbf{H}_j$ and $\mathbf{H}_j^1$ for $\mathcal{A}$ until the point where $\mathcal{A}$ makes its

$(j+1)^{\text{st}}$ classical query. In phase 2, we first note that k_1 is uniform since s_0 is uniform and independent of x_{j+1}. If $b = 0$, $\mathcal{D}$ gets access to $E^{(0)} = E$ in phase 2. Since $\mathcal{D}$ answers all quantum queries using $(E^{(0)})^{T_j,K}$ and all classical queries using $\mathsf{FX}_K[(E^{(0)})^{T_j,K}]$, we see that $\mathcal{D}$ perfectly simulates $\mathbf{H}_j$ for $\mathcal{A}$ in that case. If, on the other hand, $b = 1$ in phase 2, then $\mathcal{D}$ gets access to $E^{(1)}$, where

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k_0 \\ E_{k_0} \circ \mathsf{swap}_{s_0,\, s_1}(x) & \text{if } k^* = k_0\,. \end{cases}$$

Since $k_1 \coloneqq s_0 \oplus x_{j+1}$, it holds that

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k_0 \\ E_{k_0} \circ \mathsf{swap}_{k_1 \oplus x_{j+1},\, s_1}(x) & \text{if } k^* = k_0\,. \end{cases}$$

Moreover, the fact that s_0 (and hence $s_0 \oplus x_{j+1}$), k_0 and k_2 are uniform implies that $\mathcal{D}$ perfectly simulates $\mathbf{H}^1_j$ for $\mathcal{A}$. Applying Lemma 3.4.1 thus gives

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}^1_j) = 1\right]\right| \leq 4\sqrt{q_Q \cdot \varepsilon \cdot 2^n},$$

where,

$$\begin{aligned} \epsilon &= \max_{(k_0^*, s_0^*, s_1^*) \in \{0,1\}^{m+2n}} D(k_0, s_0, s_1) \\ &= \frac{1}{2^{m+n}(2^n - j)}. \end{aligned}$$

Therefore, we have

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1\right]\right| \leq 4\sqrt{\frac{q_Q}{2^m(2^n - j)}}\,.$$

□

Lemma 3.5.4. *For* $j = 0, \ldots, q_C$, $\mathbf{H}_j^1 = \mathbf{H}_j^2$.

Proof.

$$\mathbf{H}_j^1 : \ \left|E\right\rangle, R, \left|E\right\rangle, \cdots, R, \left|E\right\rangle, \mathsf{FX}_K[(E^{(1)})^j], \left|(E^{(1)})^j\right\rangle \ , \mathsf{FX}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots$$

$$\mathbf{H}_j^2 : \ \left|E\right\rangle, R, \left|E\right\rangle, \cdots, R, \left|E\right\rangle, \mathsf{FX}_K[(E^{(1)})^j], \left|E^{j+1}\right\rangle \quad , \mathsf{FX}_K[E^{j+1}], \left|E^{j+1}\right\rangle, \cdots.$$

We first prove the following proposition.

Proposition 2. *For* $K = (k_0, k_1, k_2) \in \{0,1\}^{m+2n}$, $j \in \{1, \ldots, q_C\}$, $i \in \{1, \ldots, j\}$, *and all* $r \in \{0, \ldots, j\}$

$$(E_{k_0}^{(1)})^{T_r, K}(x_i \oplus k_1) \ = \ E_{k_0}^{T_r, K}(x_i \oplus k_1),$$

when $s \notin \{x_1 \oplus k_1, \cdots x_j \oplus k_1\}$.

Proof. We will do a proof by induction on r. We start by the base case $r = 0$. We note that since the classical queries are not repeated, $x_{j+1} \oplus k_1 \notin \{x_1 \oplus k_1, \cdots x_j \oplus k_1\}$. Additionally since $s \notin$

$\{x_1 \oplus k_1, \cdots x_j \oplus k_1\}$, we have that for all $i \in \{1, \cdots, j\}$

$$\begin{aligned}
(E_{k_0}^{(1)})^{T_0,K}(x_i \oplus k_1) &:= E_{k_0}^{(1)}(x_i \oplus k_1) \\
&= E_{k_0} \circ \mathsf{swap}_{x_{j+1}\oplus k_1,\, s}(x_i \oplus k_1) \\
&= E_{k_0}(x_i \oplus k_1).
\end{aligned}$$

Assume for some $r - 1 \geq 0$ that

$$(E_{k_0}^{(1)})^{T_{r-1},K}(x_i \oplus k_1) = E_{k_0}^{T_{r-1},K}(x_i \oplus k_1).$$

Then by the definition,

$$\begin{aligned}
(E_{k_0}^{(1)})^{T_r,K}(x_i \oplus k_1) &= \mathsf{swap}_{(E_{k_0}^{(1)})^{T_{r-1},K}(x_r \oplus k_1),\, y_r \oplus k_2} \circ (E_{k_0}^{(1)})^{T_{r-1},K}(x_i \oplus k_1) \\
&= \mathsf{swap}_{E_{k_0}^{T_{r-1},K}(x_r \oplus k_1),\, y_r \oplus k_2} \circ E_{k_0}^{T_{r-1},K}(x_i \oplus k_1) \\
&= E_{k_0}^{T_r,K}(x_i \oplus k_1).
\end{aligned}$$

By induction, the proposition holds for all $r \in \{0, \ldots, j\}$. $\square$

In particular, for all $i \in \{1, \ldots, j\}$,

$$(E_{k_0}^{(1)})^{T_{i-1},K}(x_i \oplus k_1) = E_{k_0}^{T_{i-1},K}(x_i \oplus k_1).$$

To prove $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$ are identical, it suffices to show that the quantum oracles $(E^{(1)})^j$ and E^{j+1} are identical. Note that $(E_{k^*}^{(1)})^j = E_{k^*}^{j+1}$ for all $k^* \neq k_0$, we only need to consider the case where

$k^* = k_0$.

In both $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$, the response to the $(j+1)^{\text{st}}$ classical query is:

$$
\begin{aligned}
y_{j+1} &\stackrel{\text{def}}{=} \mathsf{FX}_K[(E^{(1)})^{T_j,K}](x_{j+1}) \\
&= (E_{k_0}^{(1)})^{T_j,K}(x_{j+1} \oplus k_1) \oplus k_2 \\
&= \left(\prod_{i=j}^{1} \mathsf{swap}_{(E_{k_0}^{(1)})^{T_{i-1},K}(x_i \oplus k_1),\, y_i \oplus k_2} \right) \circ E_{k_0}^{(1)}(x_{j+1} \oplus k_1) \oplus k_2 \\
&= \left(\prod_{i=j}^{1} \mathsf{swap}_{(E_{k_0})^{T_{i-1},K}(x_i \oplus k_1),\, y_i \oplus k_2} \right) \circ E_{k_0}(s) \oplus k_2 \\
&= E_{k_0}^{T_j,K}(s) \oplus k_2,
\end{aligned}
$$

where the fourth equality comes from Proposition 2. We use "$\prod$" to denote sequential composition of operations, i.e., $\prod_{i=1}^{n} f_i = f_1 \circ \cdots \circ f_n$. By rearranging $s = \left(E_{k_0}^{T_j,K}\right)^{-1}(y_{j+1} \oplus k_2)$, it follows that for any $x \in \{0,1\}^n$,

$$
\begin{aligned}
(E_{k_0}^{(1)})^j(x) &= \mathsf{swap}_{(E_{k_0}^{(1)})^{j-1}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \cdots \circ \mathsf{swap}_{E_{k_0}^{(1)}(x_1 \oplus k_1),\, y_1 \oplus k_2} \circ E_{k_0} \circ \mathsf{swap}_{x_{j+1} \oplus k_1,\, s}(x) \\
&= \mathsf{swap}_{E_{k_0}^{j-1}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \cdots \circ \mathsf{swap}_{E_{k_0}(x_1 \oplus k_1),\, y_1 \oplus k_2} \circ E_{k_0} \circ \mathsf{swap}_{x_{j+1} \oplus k_1,\, s}(x) \\
&= E_{k_0}^j \circ \mathsf{swap}_{x_{j+1} \oplus k_1,\, (E_{k_0}^j)^{-1}(y_{j+1} \oplus k_2)}(x) \\
&= \mathsf{swap}_{E_{k_0}^j(x_{j+1} \oplus k_1),\, y_{j+1} \oplus k_2} \circ E_{k_0}^j(x) \\
&= E_{k_0}^{j+1}(x),
\end{aligned}
$$

where the second equality comes from Proposition 2. This concludes the proof that $(E^{(1)})^j \equiv E^{j+1}$. It follows that $\mathbf{H}_j^1 = \mathbf{H}_j^2$. $\square$

Lemma 3.5.5. *For $j = 0, \dots, q_C$, $\mathbf{H}^2_j = \mathbf{H}^3_j$.*

Proof.

$$\begin{aligned}
\mathbf{H}^2_j :\ & |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |E^{j+1}\rangle \quad, \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}^3_j :\ & |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \qquad R \qquad, |E^{j+1}\rangle \quad, \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots .
\end{aligned}$$

We observe that $\mathbf{H}^2_j$ and $\mathbf{H}^3_j$ differ only in the response to the $(j+1)^{\text{st}}$ classical query. In $\mathbf{H}^2_j$, this query is answered using

$$y_{j+1} \;=\; \mathsf{FX}_K\big[(E^{(1)})^{T_j,K}\big](x_{j+1}) \;=\; E^{T_j,K}_{k_0}(s) \oplus k_2,$$

as shown in Lemma 3.5.4. In contrast, in $\mathbf{H}^3_j$ the same query is answered with

$$y_{j+1} \;=\; R(x_{j+1}).$$

Next, we prove that y_{j+1} is distributed the same in both $\mathbf{H}^2_j$ and $\mathbf{H}^3_j$. Recall Proposition 1, we have

$$y_i \;=\; E^{T_j,K}_{k_0}(x_i \oplus k_1) \oplus k_2, \quad \forall i \in [1, j],$$

and since $s \in \{0,1\}^n \setminus \{x_1 \oplus k_1, \dots, x_j \oplus k_1\}$, it follows that in $\mathbf{H}^2_j$,

$$y_{j+1} = E^{T_j,K}_{k_0}(s) \oplus k_2 \in \{0,1\}^n \setminus \{y_1, \dots, y_j\}.$$

Moreover, because $E_{k_0}^{T_j,K}$ is a permutation, the mapping

$$x_i \oplus k_1 \;\mapsto\; y_i = E_{k_0}^{T_j,K}(x_i \oplus k_1) \oplus k_2$$

is injective. Since s is chosen uniformly from $\{0,1\}^n \setminus \{x_1 \oplus k_1, \ldots, x_j \oplus k_1\}$ and k_2 is uniform, the output y_{j+1} is uniformly distributed over $\{0,1\}^n \setminus \{y_1, \ldots, y_j\}$ in $\mathbf{H}_j^2$. In $\mathbf{H}_j^3$, since classical queries are not repeated, $y_{j+1} = R(x_{j+1})$ is also uniformly distributed over $\{0,1\}^n \setminus \{y_1, \ldots, y_j\}$. Thus, the distribution of y_{j+1} conditioned on all previous queries is identical in the two hybrids.

Moreover, in both $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$, the construction of E^{j+1} follows exactly the same procedure, i.e., from the first $j+1$ classical input–output pairs and E as specified in Equation 3.22. Consequently, the two hybrids yield identical distributions, and hence $\mathbf{H}_j^2 = \mathbf{H}_j^3$. □

Lemma 3.5.6. *For $j = 0, \ldots, q_C - 1$,*

$$\Pr\big[\mathcal{A}(\mathbf{H}_j^3) = 1\big] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{m+n}}}\,,$$

where $q_{Q,j+1}$ is the expected number of queries $\mathcal{A}$ makes to P in the $(j+1)^{st}$ stage in the ideal world (i.e., in $\mathbf{H}_{q_C}$).

Proof. Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j^3$ and $\mathbf{H}_{j+1}$. We construct a distinguisher $\mathcal{D}$ for the experiment from Lemma 3.3.1:

Phase 1: $\mathcal{D}$ samples uniform $E \in \mathcal{E}(m,n)$ and $R \in \mathcal{P}(n)$. It then runs $\mathcal{A}$, answering its quantum queries using R and its classical queries using E, until after it responds to $\mathcal{A}$'s $(j+1)^{\text{st}}$ classical query. Let $T_{j+1} = \{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}$ be the list of classical queries by $\mathcal{A}$ thus far. $\mathcal{D}$ defines $F(a, k_0, x) := E_{k_0}^a(x)$ for $a \in \{1, -1\}$.

It also defines the following randomized algorithm $\mathcal{B}$: sample $k \leftarrow \{0,1\}^{m+2n}$ and write $K = (k_0^*, k_1^*, k_2^*)$. Then it computes the set B of input/output pairs to be reprogrammed so that $F^{(B)}(a, k_0^*, x) = \left(E_{k_0^*}^{T_{j+1},K}\right)^a(x)$ for all a, k_0^*, x. Finally, $\mathcal{D}$ outputs $(F, \mathcal{B})$.

Phase 2: $\mathcal{B}$ is run to generate B, and $\mathcal{D}$ is given quantum access to an oracle F_b. $\mathcal{D}$ resumes running $\mathcal{A}$, answering its quantum queries using F_b. Phase 2 ends before $\mathcal{A}$ makes its next (i.e., $(j+2)^{\text{nd}}$) classical query.

Phase 3: $\mathcal{D}$ is given the randomness used by $\mathcal{B}$ to generate k. It resumes running $\mathcal{A}$, answering its classical queries using $\mathsf{FX}_K[E^{T_{j+1},K}]$ and its quantum queries using $E^{T_{j+1},K}$. Finally, it outputs whatever $\mathcal{A}$ outputs.

It is immediate that if $b = 0$ (i.e., $\mathcal{D}$'s oracle in phase 2 is $F_0 = F$), then $\mathcal{A}$'s output is identically distributed to its output in $\mathbf{H}_{j+1}$, whereas if $b = 1$ (i.e., $\mathcal{D}$'s oracle in phase 2 is $F_1 = F^{(B)}$), then $\mathcal{A}$'s output is identically distributed to its output in $\mathbf{H}_j^3$. It follows that $|\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]|$ is equal to the distinguishing advantage of $\mathcal{D}$ in the reprogramming experiment of Lemma 3.3.1. To bound this quantity, we bound the parameter ϵ and the expected number of queries made by $\mathcal{D}$ in phase 2 (when $F = F_0$).

The value of ϵ can be bounded using the definition of $E_{k_0^*}^{T_{j+1},K}$ and the fact that $F^{(B)}(a, k_0^*, x) = \left(E_{k_0^*}^{T_{j+1},K}\right)^a(x)$. Fixing E and T_{j+1}, the probability that any particular input (a, k_0^*, x) is reprogrammed is at most the probability (over k) that it lies in the set

$$\left\{ \begin{array}{c} (1, k_0, x_i \oplus k_1), \left(1, k_0, E_{k_0}^{-1}(y_i \oplus k_2)\right), \\ (-1, k_0, E_{k_0}(x_i \oplus k_1)), (-1, k_0, y_i \oplus k_2) \end{array} \right\}_{i=1}^{j+1}.$$

We compute the probability that $(a, k_0^*, x) = (1, k_0, x_i \oplus k_1)$ for some fixed i. As k_0 and k_1 are uniform,

$$\Pr_k[(a, k_0^*, x) = (1, k_0, x_i \oplus k_1)] = \begin{cases} 2^{-(m+n)} & a = 1 \\ \\ 0 & a = -1 \end{cases}.$$

A similar bound holds for the other possibilities. By distinguishing the cases $a = 1$ and $a = -1$ and applying a union bound, we get $\varepsilon \leq 2(j+1)/2^{m+n}$.

The expected number of queries made by $\mathcal{D}$ in phase 2 when $F = F_0$ is equal to the expected number of queries made by $\mathcal{A}$ in its $(j+1)^{\text{st}}$ stage in $\mathbf{H}_{j+1}$. Since $\mathbf{H}_{j+1}$ and $\mathbf{H}_{q_C}$ are identical until after the $(j+1)^{\text{st}}$ stage is complete, this is precisely $q_{Q,j+1}$. □

3.5.3 Tightness

Let q_C represent the number of online classical queries and q_Q denote the number of offline quantum computations. Different attacks with corresponding trade-off are presented in Table 3.1.

Typical attacks involve full-key recovery, which classically requires 2^{m+n} offline queries, while quantumly, it requires only $2^{(m+n)/2}$ offline queries using Grover's algorithm. The Grover+BHT attack employs Grover's algorithm to recover the cipher key k_0 and uses the BHT/Kuwakado–Morii [KM12] to determine the whitening keys k_1 and k_2. It requires $q_Q = 2^{m/2+n/3}$ quantum queries and $q_C = 2^{n/3}$ classical queries. Details of this attack is similar to that on LRW, which is provided in Appendix 3.8.3. The resources used align with $\frac{2}{\sqrt{2^{m+n}}} q_Q \sqrt{q_C}$ in the security bound established in Theorem 3.5.2.

In Meet-in-the-Middle attack [HS18], $q_C \cdot q_Q^6 = 2^{3(m+n)}$ classical online queries are required under the condition that only polynomially many qubits are available, by using the multi-target preimage search [CNPS17]. This attack was later improved by the offline-Simon attack [BHNP+19], which

Reference	Tradeoff between q_C and q_Q
Classical [Din15]	$q = 2^{m+n}$
Grover	$q_Q = 2^{\frac{m+n}{2}}, q_c =$ constant
Grover + BHT [KM12]	$q_Q = 2^{\frac{m}{2}+\frac{n}{3}}, q_C = 2^{\frac{n}{3}}$
Meet-in-the-Middle [HS18]	$q_C \cdot q_Q^6 = 2^{3(m+n)}, q_C \leq \{2^n, 2^{3(m+n)/7}\}$
Offline-Simon [BHNP+19]	$q_C \cdot q_Q^2 = 2^{m+n}, q_C \leq 2^n$

Table 3.1: Tradeoffs for quantum attacks on the FX construction.

combines quantum search with quantum period finding, with a trade-off $q_C \cdot q_Q^2 = 2^{m+n}$, which also matches $\frac{2}{\sqrt{2^{m+n}}} q_Q \sqrt{q_C}$ term in the security bound.

3.5.4 Practical Interpretation

In practice, one typically aims for at least 128-bit security for symmetric primitives. According to Theorem 3.5.2, when $q_C \ll 2^n$, the FX construction achieves 128-bit security provided

$$\frac{q_C\sqrt{q_Q} + q_Q\sqrt{q_C}}{2^{(m+n)/2}} \leq 2^{-128},$$

up to small constant factors.

Denote $q_C = 2^c$ and $q_Q = 2^q$. Then the two terms in the numerator scale as

$$2^{c+\frac{q}{2}-\frac{m+n}{2}} \qquad \text{and} \qquad 2^{q+\frac{c}{2}-\frac{m+n}{2}}.$$

Therefore, to obtain 128-bit security, we want both terms to be at most about 2^{-128}. Equivalently,

$$2c + q - (m+n) \lesssim -256 \qquad \text{and} \qquad 2q + c - (m+n) \lesssim -256.$$

This gives a bound on $m+n$:

$$m + n \gtrsim 256 + \max(2c+q,\ c+2q).$$

We can now instantiate this condition with concrete block ciphers. If the underlying cipher used in FX is AES-128, then $m = n = 128$, which gives $m + n = 256$. Substituting this into the above condition yields

$$\max(2c+q,\ c+2q) \lesssim 0.$$

Thus, this theorem does not provide nontrivial 128-bit security for AES-128-based FX against mixed classical/quantum query attacks.

If the underlying cipher used in FX is replaced by AES-256 which is a stronger block cipher, then $m + n = 256 + 128 = 384$. The same analysis gives

$$\max(2c+q,\ c+2q) \lesssim 128.$$

This yields a meaningful range of query bounds. For example:

- $q_C = 2^{32},\ q_Q = 2^{32}$: yes, we are safe.

- $q_C = 2^{40},\ q_Q = 2^{40}$: still approximately yes.

- $q_C = 2^{48},\ q_Q = 2^{48}$: too large for a 128-bit target under this bound.

Note that in these examples the classical queries considered above are at most 2^{48}. Hence the assumption $q_C \ll 2^{128}$ is well satisfied, and the approximation used in this discussion is reasonable.

3.5.5 Application

In this section, we present post-quantum security analyses of a lightweight block cipher that instantiate the FX construction, namely PRINCE, as an application of our framework.

PRINCE [BCG+12] is a lightweight block cipher designed for ultra-low latency applications, particularly in pervasive and embedded computing. Its distinguishing feature is the α-reflection property, which makes encryption and decryption nearly identical, thereby enabling very efficient hardware implementations. Concretely, PRINCE is based on $\mathsf{PRINCE}_{\text{core}}$, which can be viewed as a $\widetilde{\mathsf{FX}}$ construction, namely the α-reflection variant of the classical FX design.

The α-reflection property ensures that this permutation can be inverted with minimal overhead. The classical security of this design is analyzed in Section 4.1 of [BCG+12].

The classical security of PRINCE is analyzed in the ideal cipher model via the primitive $\widetilde{\mathsf{FX}}$, the α-reflection variant of the standard FX construction. Let $\mathbb{F}_2^m$ denote the m-dimensional vector space over $\mathbb{F}_2$. Let H be an $(m-1)$-dimensional linear subspace of $\mathbb{F}_2^m$, and let $\alpha \in \mathbb{F}_2^m$ be a fixed nonzero element with $\alpha \notin H$. Thus, $\mathbb{F}_2^m$ is partitioned as $H \cup (\alpha \oplus H)$. Define

$$\widetilde{\mathsf{FX}}_{k_0,k_1,k_2}(x) \;=\; \begin{cases} \mathsf{FX}_{k_0,k_1,k_2}(x), & \text{if } k_0 \in H, \\[2ex] \mathsf{FX}^{-1}_{k_0\oplus\alpha,\,k_1,\,k_2}(x), & \text{if } k_0 \in \alpha \oplus H. \end{cases}$$

[BCG+12, Corollary 1] shows that $\widetilde{\mathsf{FX}}$ achieves the same level of security as the original FX construction, in the pure classical setting.

Corollary 3.5.7 ([BCG+12]). $Adv_{\mathsf{FX}}^{\mathsf{SPRP\text{-}IC}}(q_{\mathsf{FX}}, q_E) = Adv_{\widetilde{\mathsf{FX}}}^{\mathsf{SPRP\text{-}IC}}(q_{\widetilde{\mathsf{FX}}}, q_E)$.

The classical proof of Corollary 3.5.7 carries over directly to the post-quantum setting. Together with Theorem 3.5.2, this implies the post-quantum security of $\widetilde{\mathsf{FX}}$, and hence the post-quantum security of PRINCE in the QICM, i.e., where $\mathsf{PRINCE}_{\text{core}}$ is replaced by an ideal cipher to which everyone has quantum access.

3.6 Tweakable Block Ciphers

3.6.1 Definitions

Definition 3.6.1 (Tweakable Permutation). *Let $\mathcal{T}$ be a tweak space. $\widetilde{\Pi} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is a* **tweakable permutation***, where for each tweak $\tau \in \mathcal{T}$, $\widetilde{\Pi}(\tau, \cdot) \xleftarrow{\$} \mathcal{P}_n$ is an independently and randomly chosen permutation on $\{0,1\}^n$. We also define $\mathcal{E}(\mathcal{T}, n)$ to be the set of all such tweakable permutations. Additionally, we denote the "inverse" oracle as $\widetilde{\Pi}^{-1}(\tau, \cdot) := \widetilde{\Pi}_\tau^{-1}(\cdot)$ for some $\tau \in \mathcal{T}$.*

Definition 3.6.2 (Distinguishing Advantage). *Let $G : \{0,1\}^m \times \mathcal{T} \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a family of efficient, keyed permutations and $G^{-1} : \{0,1\}^m \times \mathcal{T} \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be its corresponding decryption oracle. Consider an adversary $\mathcal{A}$ (which can be either quantum or classical) with time complexity $T_{\mathcal{A}}$. The advantage of $\mathcal{A}$ in attacking G is measured by*

$$\mathsf{Adv}_G(\mathcal{A}, q) := \left| \Pr_{k \leftarrow \{0,1\}^m}\left[\mathcal{A}^{G_k(\cdot,\cdot), G_k^{-1}(\cdot,\cdot)} = 1\right] - \Pr_{\widetilde{\Pi} \leftarrow \mathcal{E}(\mathcal{T}, n)}\left[\mathcal{A}^{\widetilde{\Pi}(\cdot,\cdot), \widetilde{\Pi}^{-1}(\cdot,\cdot)} = 1\right] \right|,$$

*where the probabilities are also taken over the randomness of $\mathcal{A}$, and $\mathcal{A}$ is allowed to make q **classical** queries to the oracles within a time bound of t.*

Definition 3.6.3 (SPRP(-PQ) Security of Tweakable Block Cipher)**.** *Let $G : \{0,1\}^m \times \mathcal{T} \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a tweakable block cipher. The (Post-Quantum) strong pseudorandom permutation security of G is measured by the maximum advantage over all (quantum) adversaries $\mathcal{A}$:*

$$\begin{aligned}\mathit{Adv}_G^{\mathit{SPRP(\text{-}PQ)}}(q,t) &= \max_{\mathcal{A}:T_{\mathcal{A}}\leq t} \mathit{Adv}_G(\mathcal{A},q)\\ &= \max_{\mathcal{A}:T_{\mathcal{A}}\leq t}\left|\Pr_{k\leftarrow\{0,1\}^m}\left[\mathcal{A}^{G_k(\cdot,\cdot),G_k^{-1}(\cdot,\cdot)}=1\right] - \Pr_{\widetilde{\Pi}\leftarrow\mathcal{E}(\mathcal{T},n)}\left[\mathcal{A}^{\widetilde{\Pi}(\cdot,\cdot),\widetilde{\Pi}^{-1}(\cdot,\cdot)}=1\right]\right|.\end{aligned}$$

Here, the probabilities are additionally over the randomness of $\mathcal{A}$, and the adversary $\mathcal{A}$ is allowed up to q classical queries to the oracles within a time bound t. $\widetilde{\Pi}$ is an ideal tweakable block cipher, where $\widetilde{\Pi}(\tau,\cdot)$ is a random permutation, and $\tau \xleftarrow{\$} \mathcal{T}$.

Remark. Definition 3.6.3 provides a general definition of distinguishing advantage. However, when considering the SPRP-PQ-security of a construction in the QICM, where the underlying block cipher E is an ideal cipher, which can only be accessed through oracle queries rather than a public description; we provide adversaries with these additional oracles, $|E(\cdot,\cdot)\rangle$ and $|E^{-1}(\cdot,\cdot)\rangle$. Formally, if we consider the SPRP-PQ-security of a construction $G[E_{\_}]$ where the cipher E is accessible exclusively through oracle queries, then the security is measured as below:

$$\mathsf{Adv}_{G[E_{\_}]}^{\mathsf{SPRP\text{-}PQ}}(q,t) = \max_{\mathcal{A}}\left|\Pr_{k\leftarrow\{0,1\}^m}[\mathcal{A}^{G[E_k],G^{-1}\left[E_k^{-1}\right],|E\rangle,|E^{-1}\rangle}=1] - \Pr_{\widetilde{\Pi}\leftarrow\mathcal{E}(\mathcal{T},n)}\left[\mathcal{A}^{\widetilde{\Pi},\widetilde{\Pi}^{-1},|E\rangle,|E^{-1}\rangle}=1\right]\right|.$$

Definition 3.6.4 (XOR-Universality)**.** *A family of functions $h = \{h_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\mathcal{K}}$ is called*

ε-XOR universal if for a randomly drawn key $k \in \mathcal{K}$, $\forall x, y, z \in \{0,1\}^n$ and $x \neq y$,

$$\Pr_{k \xleftarrow{\$} \mathcal{K}} [h_k(x) \oplus h_k(y) = z] \leq \varepsilon.$$

If $\varepsilon = \frac{1}{2^n}$, h is simply called XOR universal.

Definition 3.6.5 (Uniformity)**.** *A family of functions h is uniform if $\forall x, y \in \{0,1\}^n$,*

$$\Pr_{k \xleftarrow{\$} \mathcal{K}} [h_k(x) = y] = \frac{1}{2^n}.$$

Remark. We note that, for these properties to hold, it is necessary that $|k| \geq n$. This further implies that for all x, y, there exists at least one k such that $h_k(x) = y$.

Definition 3.6.6 (LRW construction [LRW02])**.** *Let m and n be positive integers. Let $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let h be a hash function. The LRW construction is defined as*

$$\mathsf{LRW}_{k,k'}^{E,h}(\tau, x) = E_k(x \oplus h_{k'}(\tau)) \oplus h_{k'}(\tau),$$

where $k \xleftarrow{\$} \{0,1\}^m$ is the block cipher key and $k' \xleftarrow{\$} \{0,1\}^\kappa$ is the tweak key, $x \in \{0,1\}^n$ is the input, $\tau \in \{0,1\}^*$ is the tweak, and h is a hash function. For simpler notation, we write $\mathsf{LRW}_{k,k'}^{E,h}(\tau, x)$ as $\mathsf{LRW}_{k,k'}(\tau, x)$, when E and h are clear from the context.

Theorem 3.6.7 (Classical security of LRW [LRW02, Min06])**.** *Let h be an ε-XOR-universal hash function. Then for $\mathsf{LRW}^{E,h}$ construction,*

$$\mathsf{Adv}_{\mathsf{LRW}}^{\mathsf{SPRP}}(q, t) \leq \mathsf{Adv}_{E}^{\mathsf{SPRP}}(q, t) + q^2 \varepsilon,$$

where q is the number of queries to $\mathsf{LRW}_{k,k'}(\cdot,\cdot)$ within a time bound t.

We note that the security of XEX2 is not analyzed via the hybrid argument in this section. Instead, its bound is derived later using the more general theorem presented in Theorem 3.7.8.

Definition 3.6.8 (XEX2 construction [Rog13])**.** *Let m and n be positive integers. Let $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher, and let $\alpha \in \mathbb{F}^*_{2^n}$. The tweakable block cipher construction XEX2 is*

$$\mathsf{XEX2}^{E,\alpha}_{k,k'}(x,i,j) = E_k(x \oplus \Delta_{i,j,k'}) \oplus \Delta_{i,j,k'},$$

where $\Delta_{i,j,k'} = \alpha^j \times_* E_{k'}(i)$. Here, $\times_*$ denotes multiplication in the finite field $\mathbb{F}_{2^n} \setminus \{0^n\}$. The key of XEX2 consists of the block cipher keys $k, k' \xleftarrow{\$} \{0,1\}^m$. Furthermore, $x, i \in \{0,1\}^n$, and $j \in [0, 2^{20}-1]$.

We include a classical security analysis for completeness here. To facilitate the (classical and post-quantum) security analysis of XEX2, we define an inefficient idealized variant of XEX2, where $E_{k'}$ is replaced with a random permutation π, which is then considered part of the key.

Definition 3.6.9 (XEX2 with idealized hash)**.** *The tweakable block cipher construction* $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ *is*

$$\mathsf{XEX2}^{E,\alpha}_{k,\pi}(x,i,j) = E_k(x \oplus \Delta_{i,j}) \oplus \Delta_{i,j},$$

where $\Delta_{i,j} = \alpha^j \times_* \pi(i)$. The key of $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ *consists of the block cipher key $k \xleftarrow{\$} \{0,1\}^m$ and the permutation $\pi \in S_{2^n}$, while the remaining parameters are as in Definition 3.6.8.*

Observe that $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ is the LRW construction with the hash function family

$$h_{\pi}^{\mathsf{XEX2}^{\mathrm{ideal-hash}}}(i,j) = \Delta_{i,j} = \alpha^{j} \times_{*} \pi(i).$$

Clearly, $\mathsf{XEX2}$ and $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ are indistinguishable for an SPRP adversary, up to the SPRP security of E.

Proposition 3. *The security of* $\mathsf{XEX2}$ *and* $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ *is related as follows.*

$$\mathsf{Adv}_{\mathsf{XEX2}}^{\mathsf{SPRP}}(q,t) \leq \mathsf{Adv}_{E}^{\mathsf{SPRP}}(q,t) + \mathsf{Adv}_{\mathsf{XEX2}^{\mathrm{ideal-hash}}}^{\mathsf{SPRP}}(q,t).$$

This inequality holds for various adversarial models, including classical, post-quantum and quantum-access.

Next, we show that the hash function in $\mathsf{XEX2}^{\mathrm{ideal-hash}}$ is ε-XOR universal for negligible ε.

Lemma 3.6.10. *The hash function family* $h^{\mathsf{XEX2}^{\mathrm{ideal-hash}}}$ *is* $\frac{1}{2^n-1}$*-XOR universal.*

Proof. We bound

$$\max_{z\in\{0,1\}^n,(i_1,j_1)\neq(i_2,j_2)} \Pr_{\pi}[\pi(i_1) \times_{*} \alpha^{j_1} \oplus \pi(i_2) \times_{*} \alpha^{j_2} = z]$$

If $i_1 \neq i_2$, since π is a random permutation, the values $\pi(i_1)$ and $\pi(i_2)$ are distinct random elements from GF(2^n), implying that the XOR expression is uniformly distributed over the set $\mathsf{GF}(2^n) \setminus \{\alpha^{j_1} \oplus \alpha^{j_2}\}$. In this case, the probability expression above us thus bounded by $1/(2^n-1)$. When $i_1 = i_2$ and

$j_1 \neq j_2$, the probability expression becomes:

$$\Pr_\pi[\pi(i_1) \times_* \alpha^{j_1} \oplus \pi(i_1) \times_* \alpha^{j_2} = z]$$
$$= \Pr_\pi[\pi(i_1) \times_* (\alpha^{j_1} \oplus \alpha^{j_2}) = z]$$
$$= \Pr_\pi[\pi(i_1) = z \times_* (\alpha^{j_1} \oplus \alpha^{j_2})^{-1}].$$

Since π is a random permutation, the value $\pi(i_1)$ is uniformly distributed, and thus $\varepsilon \leq \frac{1}{2^n-1}$. □

We now obtain the classical security of XEX2.

Theorem 3.6.11 (Classical security of XEX2 [Rog13]). *Fix $n \geq 1$ and let $\alpha \in \mathbb{F}_{2^n}^*$ be base elements. Fix a block cipher $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$. Then*

$$\mathit{Adv}^{\mathit{SPRP}}_{\mathit{XEX2}}(q,t) \leq 2\mathit{Adv}^{\mathit{SPRP}}_{E}(q,t) + \frac{q^2}{2^n-1},$$

where q is the number of queries to XEX2, within a time bound t.

Proof. Immediate by combining Theorem 3.6.7, Proposition 3 and Lemma 3.6.10. □

The tweakable block cipher XEX2 is employed in the NIST-standardized XTS-AES [P1619], with AES as the underlying block cipher. However, the application-level security goals of XTS remain poorly understood [CSR+08, Rog13]. Another variant, which we denote by XEX, uses a single key ($k = k'$). Discussions on whether to employ a single key or two separate keys are explored in [CSR+08, LM08], with no definitive conclusion reached.

Applications of LRW and XEX2. LRW is implemented as LRW-AES Encryption in earlier drafts of IEEE P1619 [P1606], utilizing AES as the block cipher E and the hash function used is $h_{k'}(\tau) =$

$k' \times_* \tau$, where $\times_*$ represents the group multiplication in $\mathbb{F}^*_{2^n}$. For a single 128-bit block, the block cipher key k is 128, 192, or 256 bits, while the hash key k' is 128 bits. The tweak τ is 128 bits, and E operates on 128-bit (tweakable) blocks.

This LRW-AES was later replaced by XTS-AES in the final IEEE P1619 standard [P1619], with a constraint that the data unit length for any XTS-AES implementation must not exceed 2^{20} AES blocks [Dwo10a]. Initially, LRW was considered slower than XTS due to the full multiplication over $\mathsf{GF}(2^n)$ required for each tweak update, but this is no longer the case [IM19]. The input or output data may consist of multiple 128-bit blocks followed by a partial block of less than 128 bits.

In the XTS-AES Encryption procedure [P1619] for a single 128-bit block, the keys k and k' are either 128 bits or 256 bits. The tweak is represented by i, a 128-bit value, and j, a sequential number of the 128-bit block within the data unit.

3.6.2 Post-quantum security of LRW using the hybrid technique

Theorem 3.6.12. *Let LRW be as in Definition 3.6.6 and let $\mathcal{A}$ be an adversary making q_C classical queries to its first oracle and $q_Q \geq 1$ quantum queries to its second oracle. Assuming h is XOR-universal and uniform, it holds that in the ideal cipher model,*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^m; k' \leftarrow \{0,1\}^\kappa \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \mathsf{LRW}_{k,k'}[E_\cdot], |\pm E\rangle} = 1 \right] - \Pr_{\substack{\widetilde{\Pi} \leftarrow \mathcal{E}(\mathcal{T}, n); \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \widetilde{\Pi}, |\pm E\rangle} = 1 \right] \right|$$
$$\leq \frac{6 q_C^2}{2^n} + \frac{4}{2^{(m+n)/2}} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

Proof. The proof follows a procedure similar to that of the post-quantum security proof for FX (Theorem 3.5.2). However, in LRW, we introduce bad events related to collisions, defined as $\mathsf{Bad}_j =$

$\mathsf{Bad}_{j,1} \wedge \mathsf{Bad}_{j,2}$, where j indicates the j^{th} stage of the hybrid:

- $\mathsf{Bad}_{j,1}$: A collision occurs in the set $\{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$,
- $\mathsf{Bad}_{j,2}$: A collision occurs in the set $\{y_1 \oplus h_{k'}(\tau_1), \ldots, y_j \oplus h_{k'}(\tau_j)\}$.

Using the hybrid technique, we analyze under the condition that these bad events do not occur:

$$
\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1]| \\
&= |\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \neg\mathsf{Bad}_{q_C}] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \mathsf{Bad}_{q_C}]| \\
&\leq |\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \neg\mathsf{Bad}_{q_C}]| + \Pr[\mathsf{Bad}_{q_C}] \\
&\leq \sum_{j=0}^{q_C-1} |\Pr[\mathcal{A}(\mathbf{H}_j) = 1 \wedge \neg\mathsf{Bad}_j] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \wedge \neg\mathsf{Bad}_{j+1}]| + \Pr[\mathsf{Bad}_{q_C}].
\end{aligned}
$$

This way to introducing bad events was first employed in [Hos25]. Detailed proof for LRW is in Section 3.8.4. □

3.6.3 Tightness

We summarize the best-known attacks on standardized LRW and XEX2, with detailed presentations provided in Section 3.8.3:

- Classical Distinguishing Attack: The best-known distinguishing attacks on LRW and XEX2, which aim to differentiate them from a block cipher, exploit either birthday collisions or Even-Mansour-related structures. These attacks generally require only $O(2^{n/2})$ online queries. This matches the $q_C^2 \cdot 2^{-n}$ term in the security bound.
- Classical Complete Key Recovery Attack: Distinguishing attacks are first employed to iden-

tify the tweak key, followed by an exhaustive key search for the cipher key. This process requires a total of $O(2^{n/2} + 2^m)$ online queries and $O(2^m)$ offline queries.

- Quantum Complete Key Recovery Attack (shorter cipher key): The offline-Simon attack [BHNP+19] satisfies the trade-off $q_C q_Q^2 = 2^{m+n}$. For block ciphers with shorter key lengths ($m \leq 2n$), this attack, recognized as the best known attack, recovers the secret key using approximately $\widetilde{O}(2^{(m+n)/3})$ online (classical) and $\widetilde{O}(2^{(m+n)/3})$ offline (quantum) queries. Up to poly-logarithmic factors, this aligns with the term $\sqrt{q_C}\, q_Q \cdot 2^{(m+n)/2}$ in our security bound.

- Quantum Complete Key Recovery Attack (longer cipher key): For longer key lengths ($m > 2n$), we can combine Grover's algorithm with the Kuwakado-Morii attack [KM12]. The query complexity is $q_Q = O(2^{m/2+n/3})$ and $q_C = O(2^{n/3})$. Notably, this matches the offline-Simon attack's trade-off, satisfying $q_Q^2 q_C = O(2^{(m+n)/2})$. Alternatively, we can apply Grover's algorithm alone to recover both the cipher key and tweak key, resulting in $q_Q = O(2^{(m+n)/2})$. The choice of attack depends on whether we prioritize minimizing quantum or classical queries. Similarly, it aligns with the term $\sqrt{q_C}\, q_Q \cdot 2^{(m+n)/2}$ in our security bound.

3.6.4 Practical Interpretation

Similar to FX, we aim for at least 128-bit security for LRW. According to Theorem 3.6.12, the LRW construction achieves this level security provided

$$\frac{q_C^2}{2^n} + \frac{q_C\sqrt{q_Q} + q_Q\sqrt{q_C}}{2^{(m+n)/2}} \leq 2^{-128}.$$

up to small constant factors. Similar analysis would give that to achieve 128-bit security, we need

$$n \gtrsim 128 + 2c,$$

$$m + n \gtrsim 256 + \max(2c + q,\ c + 2q).$$

A key difference from the FX theorem is the additional birthday-type term $q_C^2/2^n$. This term depends only on the block length n, and therefore imposes an intrinsic limitation on the achievable security level in practice. Since AES has block length $n = 128$ for AES-128, 192, 256, which gives $c \lesssim 0$, this theorem should be interpreted as supporting at most birthday-style security in the number of classical queries.

Instead, if one asks for 64-bit security, then the birthday condition becomes $2c - n \lesssim -64$. When $n = 128$, this gives $c \lesssim 32$, that is, roughly $q_C \lesssim 2^{32}$. This matches the usual birthday-barrier intuition for tweakable block-cipher constructions with 128-bit blocks.

3.7 Block cipher modes

3.7.1 Definitions

Definition 3.7.1 (Distinguishing Advantage)**.** *Let $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be an efficient, keyed permutation. Consider an adversary $\mathcal{A}$ (which can be either quantum or classical). The advantage of $\mathcal{A}$ in attacking E is measured by*

$$\mathsf{Adv}_E(\mathcal{A}) := \left| \Pr_{k \leftarrow \{0,1\}^m}[\mathcal{A}^{E_k(\cdot), E_k^{-1}(\cdot)} = 1] - \Pr_{P \leftarrow \mathcal{P}_n}[\mathcal{A}^{P(\cdot), P^{-1}(\cdot)} = 1] \right|,$$

where the probabilities are also taken over the randomness of $\mathcal{A}$.

For the rest of the paper, we define $T_{\mathcal{A}}$ as the time an adversary $\mathcal{A}$ spends and $Q_{\mathcal{A}}$ as the number of **classical** queries that $\mathcal{A}$ makes to the oracles.

Definition 3.7.2 (SPRP(-PQ) Security)**.** *Let $E : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ be an efficient keyed permutation. The (PQ) strong pseudorandom permutation security of E is measured by the maximum advantage over all (quantum) adversaries $\mathcal{A}$:*

$$\begin{aligned} \mathsf{Adv}_E^{\mathsf{SPRP(\text{-}PQ)}}(q,t) &:= \max_{\substack{\mathcal{A}:\, T_{\mathcal{A}}\le t;\\ Q_{\mathcal{A}}\le q}} \mathsf{Adv}_E(\mathcal{A}) \\ &= \max_{\substack{\mathcal{A}:\, T_{\mathcal{A}}\le t;\\ Q_{\mathcal{A}}\le q}} \left| \Pr_{k\leftarrow\{0,1\}^m}[\mathcal{A}^{E_k(\cdot),E_k^{-1}(\cdot)} = 1] - \Pr_{P\leftarrow\mathcal{P}_n}[\mathcal{A}^{P(\cdot),P^{-1}(\cdot)} = 1] \right|, \end{aligned}$$

where the probabilities are also taken over the randomness of $\mathcal{A}$.

Definition 3.7.3 (PRF(-PQ) Security)**.** *Let $E : \{0,1\}^m \times \{0,1\}^* \to \{0,1\}^*$ be an efficient keyed function. The (PQ) pseudorandom function security of E is measured by the maximum advantage over all (quantum) adversaries $\mathcal{A}$:*

$$\mathsf{Adv}_E^{\mathsf{PRF(\text{-}PQ)}}(q,t) := \max_{\substack{\mathcal{A}:\, T_{\mathcal{A}}\le t;\\ Q_{\mathcal{A}}\le q}} \left| \Pr_{k\leftarrow\{0,1\}^m}[\mathcal{A}^{E_k(\cdot)} = 1] - \Pr_{f}[\mathcal{A}^{f(\cdot)} = 1] \right|,$$

where f is chosen uniformly from the set of functions mapping $\{0,1\}^*$ to $\{0,1\}^*$; the probabilities are also taken over the randomness of $\mathcal{A}$.

3.7.2 SPRP-PQ security of an ideal cipher

We consider quantum information-theoretic adversaries that are not constrained by computational resources, such as time or the number of available qubits. The only restriction is on the number of queries that the adversary can make to its oracles.

Definition 3.7.4. *Let E be the ideal cipher in the quantum ideal cipher model, and let $\mathcal{A}$ be a quantum adversary. The advantage of $\mathcal{A}$ in attacking E is measured by:*

$$\mathsf{Adv}_E(\mathcal{A}) := \left| \Pr_{k \leftarrow \{0,1\}^m} \left[\mathcal{A}^{E_k(\cdot), E_k^{-1}(\cdot), |E(\cdot,\cdot)\rangle, |E^{-1}(\cdot,\cdot)\rangle} = 1 \right] - \Pr_{P \leftarrow \mathcal{P}_n} \left[\mathcal{A}^{P, P^{-1}, |E\rangle, |E^{-1}\rangle} = 1 \right] \right|.$$

For simplicity of presentation, we sometimes later will omit the inverse oracles when considering SPRP(-PQ)-security. Similarly, SPRP-PQ security of an ideal cipher E is the maximum advantage taken over all possible quantum adversaries $\mathcal{A}$:

$$\mathsf{Adv}_E^{\mathsf{SPRP\text{-}PQ}}(q_1, q_2) := \max_{\substack{\mathcal{A}:\, Q_{1,\mathcal{A}} \le q_1; \\ Q_{2,\mathcal{A}} \le q_2}} \mathsf{Adv}_E(\mathcal{A}),$$

where $Q_{1,\mathcal{A}}$ and $Q_{2,\mathcal{A}}$ represent the queries that $\mathcal{A}$ makes to $E_k^{\pm}(\cdot)$ and $|E^{\pm}(\cdot,\cdot)\rangle$, respectively.

We emphasize that no block cipher can offer greater security than an ideal cipher. Now we start examining the SPRP-PQ security of an ideal cipher. To establish a bound on the distinguishing probability, we reduce to the UNIQUE-SEARCH problem from the distinguishing problem.

Definition 3.7.5. *(UNIQUE-SEARCH$_n$) Given a function $f : \{0,1\}^n \to \{0,1\}$, such that f maps at most one element to 1, output YES if $f^{-1}(1)$ is non-empty and NO otherwise.*

Theorem 3.7.6. *Let E be an ideal cipher (in the quantum ideal cipher model) and assume an adversary*

$\mathcal{A}$ making q_C classical queries to E_k and q_Q quantum queries to E. Then it holds that

$$\textsf{\textit{Adv}}_E^{\textsf{\textit{SPRP-PQ-IC}}}(q_C, q_Q) = \max_{\mathcal{A}} \left| \Pr_{k \leftarrow \{0,1\}^m}[\mathcal{A}^{E_k, |E\rangle} = 1] - \Pr_{P \leftarrow \mathcal{P}_n}[\mathcal{A}^{P, |E\rangle} = 1] \right|$$
$$\leq \max_{\mathcal{A}'} \textsf{\textit{Adv}}_f^{\textsf{\textit{UNIQUE-SEARCH}}}(\mathcal{A}').$$

Proof. Given an adversary $\mathcal{A}$ that attacks E, we now construct an adversary $\mathcal{A}'^{|f\rangle}$ attacking UNIQUE-SEARCH problem:

1. $\mathcal{A}'$ generates a permutation $P \in \mathcal{P}_n$ and an ideal block cipher $E \in \mathcal{E}(m, n)$.

2. $\mathcal{A}'$ then constructs a function E' by querying its own oracle f:

$$E'(k, \tau) = \begin{cases} P(\tau) & \text{if } f(k) = 1 \\ E(k, \tau) & \text{if } f(k) \neq 1 \end{cases}.$$

   $\mathcal{A}'$ subsequently provides $\mathcal{A}$ with classical oracle access to P, P^{-1}, as well as quantum oracle access to E', E'^{-1}. The construction of these quantum oracles is detailed in Section 3.8.5.

3. $\mathcal{A}'$ outputs what $\mathcal{A}$ outputs.

We now argue that if $\mathcal{A}$ successfully distinguishes between having oracle access to $(P, |E\rangle)$ or $(E_k, |E\rangle)$, then $\mathcal{A}'$ can successfully attack the UNIQUE-SEARCH problem, specifically determining whether f^{-1} is empty or not. We will present our argument by considering the following cases:

YES case. There exists a k^* such that $f(k^*) = 1$, which implies that $E'(k^*, \tau) = P(\tau)$ for any t. From the perspective of $\mathcal{A}$, for this specific k^*, the behavior of the second oracle E' when queried with $(k^*, \cdot)$ matches the behavior of the first oracle P when queried with $(\cdot)$. This situation is equivalent to what $\mathcal{A}$ expects to observe in the interaction with the oracles $(E_k, |E\rangle)$, i.e. $\mathcal{A}^{E_k,|E\rangle}$.

NO case. There does not exist any k such that $f(k) = 1$, which implies that for all k and any τ, we have $E'(k, \tau) = E(k, \tau)$. Consequently, from the perspective of $\mathcal{A}$, the interaction with the oracles $(P, |E'\rangle)$ is just $(P, |E\rangle)$, which gives $\mathcal{A}^{P,|E\rangle}$.

Therefore, if $\mathcal{A}$ can distinguish the real world $\left(\mathcal{A}^{E_k,|E\rangle}\right)$ and the ideal world $\left(\mathcal{A}^{P,|E\rangle}\right)$, then $\mathcal{A}'$ can distinguish YES and NO case in UNIQUE-SEARCH:

$$\mathsf{Adv}_f^{\mathsf{UNIQUE\text{-}SEARCH}}(\mathcal{A}') \geq \mathsf{Adv}_E(\mathcal{A}).$$

We note that the number of queries $\mathcal{A}'$ makes to f is determined only by q_Q, as specified by the construction of E'. $\square$

The UNIQUE-SEARCH bound against quantum adversaries [BBBV97] yields:

Corollary 3.7.7. *Let E be an ideal cipher (in QICM) and assume an adversary $\mathcal{A}$ making q_C queries to E_k and q_Q **quantum** queries to E. Then it holds that*

$$\mathit{Adv}_E^{\mathit{SPRP\text{-}PQ\text{-}IC}}(q_C, q_Q) \leq \frac{q_Q^2}{2^m}.$$

We note that the bound is optimal, and Grover's search is the matching algorithm, which involves querying E $2^{m/2}$ times to find the key. It is evident from the bound that only q_Q, i.e., the number of queries to E, is involved. This implies that the security lower bound stays the same in the Q2 model, in which the adversary can query both of its oracles quantumly.

3.7.3 Security of block cipher modes

General Theorem **Construction Con.** We begin by defining a polynomial-time algorithm, denoted as Con, which uses a (black-box) permutation as input to produce a specific construction. This permutation can take on different forms: when the construction is instantiated with a specific block cipher E, we denote it as $\mathsf{Con}[E_{\_}]$. Then with a key $k \leftarrow \{0,1\}^m$ is uniformly sampled first, E_k then serves as the (black-box) permutation, denoted as $\mathsf{Con}[E_k]$. For Con that is invertible, we denote its inverse as DeCon and with different underlying permutations, it is denoted as $\mathsf{DeCon}\left[E_k^{-1}\right]$ and $\mathsf{DeCon}\left[P^{-1}\right]$.

To illustrate, consider Con as the $\mathsf{CBC\text{-}MAC}$ construction. When the permutation is instantiated with a block cipher E, the construction first samples a uniformly random key $k \leftarrow \{0,1\}^m$ and uses it as the block cipher key for E, which gives $\mathsf{Con}[E_k] = \mathsf{CBC\text{-}MAC}[E_k]$. The output of this construction is given by:

$$\mathsf{CBC\text{-}MAC}[E_k](x_1, \ldots, x_\ell) = E_k(E_k(\ldots E_k(E_k(x_1) \oplus x_2) \oplus \ldots) \oplus x_\ell).$$

Alternatively, when the permutation-like oracle is instantiated as a true random permutation P, the construction is denoted by $\mathsf{Con}[P] = \mathsf{CBC\text{-}MAC}[P]$. In this scenario, the block cipher E_k is replaced entirely by the permutation P within the construction, resulting in the following definition:

$$\mathsf{CBC\text{-}MAC}[P](x_1, \ldots, x_\ell) = P(P(\ldots P(P(x_1) \oplus x_2) \oplus \ldots) \oplus x_\ell).$$

Experiment Exp. The security of a cryptographic scheme Con is typically analyzed through a well-defined probabilistic experiment, denoted as Exp. This experiment captures the interaction between an adversary $\mathcal{A}$ and a challenger, modeled as a classical polynomial-time algorithm. The challenger has access to certain resources, such as a black-box permutation oracle, which it can query as needed during the experiment. The goal of the adversary $\mathcal{A}$ is to exploit these interactions and the oracle queries to either compromise a specific security property of Con or extract secret information. The experiment $\mathsf{Exp}(\mathcal{A})$ proceeds as follows:

1. **Initialization and Oracle Access**: The challenger Chall is possible to be given access to one or more oracles, such as black-box permutation oracles, which it may use to process information or respond to the adversary's queries. Chall may also grant the adversary direct access to some oracles under predefined rules. Con can also obtain oracle access to these oracles. We assume that the number of queries to these oracles only depends on the number of queries the adversary makes to the construction Con.

2. **Interaction with the Adversary**: The challenger Chall and the adversary $\mathcal{A}$ engage in a protocol where information is exchanged. The adversary may send queries or responses to the challenger, aiming to gather information or manipulate the interaction to achieve its goals. The interaction is always classical.

3. **Output of the Experiment**: After a sequence of interactions and oracle queries, the challenger outputs a single bit $b \in \{0, 1\}$. This bit represents the outcome of the experiment, where its distribution reflects the adversary's success in breaking the scheme Con or achieving a specific objective.

The success of the adversary $\mathcal{A}$ is quantified by the probability that it achieves a favorable outcome in Exp. This is measured in terms of the adversary's **advantage**, denoted as $\mathsf{Adv}^{\mathsf{Exp}}_{\mathsf{Con}}(\mathcal{A})$. The advantage reflects the extent to which the adversary outperforms random guessing in achieving its goals. Finally, the security of the scheme Con is determined by evaluating the maximum advantage achievable by any adversary $\mathcal{A}$. This is expressed as:

$$\mathsf{Adv}_{\mathsf{Con}}^{\mathsf{Exp}}(\cdot) = \max_{\mathcal{A}} \mathsf{Adv}_{\mathsf{Con}}^{\mathsf{Exp}}(\mathcal{A}).$$

Here are some example experiments. The PRF experiment is a standard framework to evaluate the pseudorandom function PRF security of a construction Con. This experiment proceeds as follows:

1. **Initialization and Oracle Access**: The challenger initializes the experiment by setting up the construction $\mathsf{Con}_k(\cdot)$, where $k \in \{0,1\}^\kappa$ is a secret key sampled uniformly at random. A bit $d \xleftarrow{\$} \{0,1\}$ is sampled uniformly at random. This bit determines the behavior of the oracle provided to the adversary $\mathcal{A}$:

   - If $d = 0$, the challenger grants $\mathcal{A}$ access to the oracle implementing the function $\mathsf{Con}_k(\cdot)$.
   - If $d = 1$, for each query m from $\mathcal{A}$, the challenger responds with a value $y \xleftarrow{\$} \mathcal{Y}$ sampled uniformly at random from the output range $\mathcal{Y}$, independent of m, mimicking a random oracle.

2. **Interaction with the Adversary**: The adversary $\mathcal{A}$ is allowed to interact with the oracle provided by the challenger. It may issue a sequence of queries m and receive corresponding responses, either from $\mathsf{Con}_k(\cdot)$ (if $d = 0$) or the random oracle (if $d = 1$).

3. **Output of the Experiment**: After a sequence of interactions, $\mathcal{A}$ outputs a bit d', representing its guess about whether the oracle is $\mathsf{Con}_k(\cdot)$ ($d = 0$) or a random oracle ($d = 1$). The challenger then computes the output of the experiment as a bit $b = \overline{d \oplus d'}$, where $b = 1$ indicates that $\mathcal{A}$ correctly distinguished the oracle type and $b = 0$ indicates that $\mathcal{A}$ guessed incorrectly.

The adversary's goal is to maximize the probability of outputting the correct guess, $d' = d$, thus making $b = 1$. The PRF security of the construction Con is defined as the maximum distinguishing advantage over all adversaries $\mathcal{A}$ running within the given resource constraints (q), denoted as $\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{Con}}(q)$.

The distinguishing advantage of the adversary $\mathcal{A}$ in the PRF experiment is defined as:

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{Con}}(\mathcal{A}) = \left| \Pr[b = 1] - \frac{1}{2} \right|.$$

The PRF security of the construction Con is then defined as the maximum distinguishing advantage over all adversaries $\mathcal{A}$ running within the given resource constraints (q, t):

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{Con}}(q, t) = \max_{\mathcal{A}} \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{Con}}(\mathcal{A}).$$

Consider another example, the $\mathsf{Forgery}$ experiment. The $\mathsf{Forgery}$ experiment evaluates the unforgeability of a construction Con, demonstrating that Con is a MAC (Message Authentication Code). It is structured as follows:

1. **Initialization and Oracle Access**: The challenger initializes the experiment by setting up the construction $\mathsf{Con}_k(\cdot)$, where $k \in \{0, 1\}^\kappa$ is a secret key sampled uniformly at random. The oracle $\mathsf{Con}_k(\cdot)$ is made available to the adversary $\mathcal{A}$.

2. **Interaction with the Adversary**: The adversary $\mathcal{A}$ interacts with the oracle $\mathsf{Con}_k(\cdot)$. On a query m, the oracle computes the tag $\mathsf{tag} = \mathsf{Con}_k(m)$ and returns tag to $\mathcal{A}$.

3. **Output of the Experiment**: After a sequence of interactions, $\mathcal{A}$ outputs a pair (m, tag). Let $\mathcal{Q}$ denote the set of all queries that $\mathcal{A}$ asked its oracle. The challenger outputs $b = 1$ if and only if $\mathsf{Con}_k(m) = \mathsf{tag}$ and $m \notin \mathcal{Q}$, the adversary produces a valid tag for a message m that it has not

queried during the interaction. Otherwise, the challenger outputs 0.

The adversary's goal is to maximize the probability of producing a valid, unqueried message-tag pair (m, tag) such that the experiment outputs 1. The unforgeability of the construction Con is quantified by the adversary's advantage:

$$\mathsf{Adv}_{\mathsf{Con}}^{\mathsf{Forgery}}(\mathcal{A}) = \Pr[b = 1].$$

The unforgeability of Con is then defined as the maximum advantage over all adversaries $\mathcal{A}$ constrained by a certain number of queries q and time t:

$$\mathsf{Adv}_{\mathsf{Con}}^{\mathsf{Forgery}}(q, t) = \max_{\mathcal{A}} \mathsf{Adv}_{\mathsf{Con}}^{\mathsf{Forgery}}(\mathcal{A}).$$

For $\mathsf{Forgery}$ experiment with decryption oracle access, denoted as $\mathsf{Forgery}\pm$, it is structured as follows for a construction Con:

1. **Initialization and Oracle Access**: The challenger Chall initializes the experiment by setting up the construction $\mathsf{Con}_k(\cdot)$ and $\mathsf{DeCon}_k(\cdot)$, where $k \in \{0, 1\}^\kappa$ is a secret key sampled uniformly at random. Both oracles are made available to the adversary $\mathcal{A}$.

2. **Interaction with the Adversary**: The adversary $\mathcal{A}$ interacts with the oracle $\mathsf{Con}_k(\cdot)$ and $\mathsf{DeCon}_k(\cdot)$. When $\mathsf{Con}_k(\cdot)$ is queried on a query m, the oracle computes the tag $\mathsf{tag} = \mathsf{Con}_k(m)$ and returns t to $\mathcal{A}$. When $\mathsf{DeCon}_k(\cdot)$ is queried on a query t, the oracle computes the tag $m = \mathsf{DeCon}_k(t)$ and returns m to $\mathcal{A}$.

3. **Output of the Experiment**: Same as in $\mathsf{Forgery}$ experiment.

General Theorem. We now focus on a collection of constructions whose security is analyzed in a typical reduction-based approach. The classical security of the construction $\mathsf{Con}[E_{\_}]$, for a particular block cipher E in a specific experiment Exp, is established through the following steps:

1. **Reduction to the Block Cipher SPRP Security:** Replace the block cipher E_k with a truly random permutation $P \xleftarrow{\$} \mathcal{P}_n$, incurring a loss equivalent to the SPRP security of E.

2. **Information-Theoretic Security of** $\mathsf{Con[P]}$**:** When E is replaced by P, it can be shown that the construction $\mathsf{Con}[P]$ achieves a certain level of security in an information-theoretic sense. This means that the security is determined solely by the number of queries q made to the construction oracle $\mathsf{Con}[P]$ and its inverse, independent of the adversary's computational power.

Therefore, the final security of $\mathsf{Con}[E_{\_}]$ in a given experiment Exp against a classical probabilistic adversary making q queries in time t is given by:

$$\mathsf{Adv}^{\mathsf{Exp}}_{\mathsf{Con}[E_{\_}]}(q,t) \leq \mathsf{Adv}^{\mathsf{SPRP}}_{E}(q',t) + \delta(q),$$

Here, $q' = f(q)$, a function of q, represents the number of queries made by Con and Chall to $E_{\_}(\cdot)$. Additionally, t denotes the computational resources required for the reduction to the security of the underlying block cipher.

This approach combines the underlying computational security of the block cipher E and the inherent information-theoretic security of the construction when using a random permutation P. When Con is instantiated with a specific block cipher E, we can extend the classical security analysis of Con in a given experiment Exp to establish its post-quantum security in the same experiment Exp as follows:

Theorem 3.7.8. *Let Exp be a security experiment, and let Con be a construction as defined above, instantiated with a block cipher E. Then the post-quantum security of Con is bounded as follows:*

$$\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\text{-}}]}(q,t) \leq \mathsf{Adv}^{\mathsf{SPRP\text{-}PQ}}_{E}(q',t) + \delta(q),$$

where q is the number of queries made to the keyed construction and its inverses, $\mathsf{Con}[E_k](\cdot)$ and $\mathsf{DeCon}\left[E_k^{-1}\right](\cdot)$; $q' = f(q)$ represents the number of queries made by Con and the challenger Chall to $E_{\text{-}}(\cdot)$; t denote the time resources available to quantum adversaries in the experiments Exp. $\delta(q)$ is the classical-model information-theoretic security of Con instantiated with an ideal block cipher.

Proof. Suppose $\mathcal{A}$ is a quantum adversary attacking $\mathsf{Con}[E_{\text{-}}]$ in a specified experiment Exp. The corresponding security is:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\text{-}}]}(q,t) &= \max_{\mathcal{A}} \mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\text{-}}]}(\mathcal{A}) \\ &= \max_{\mathcal{A}} \left| \Pr_{k \leftarrow \{0,1\}^m}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[E_k],\mathsf{DeCon}\left[E_k^{-1}\right]}\right) = 1\right] - \Pr_{P \leftarrow \mathcal{P}_n}\left[\mathsf{Exp}\left(\mathcal{A}^{P,P^{-1}}\right) = 1\right]\right|.\end{aligned}$$

We then consider a special class of quantum adversaries, denoted as $\mathcal{B}$. As defined in Definition 3.7.1, $\mathcal{B}$ has classical oracle access to $\mathcal{O}$ and $\mathcal{O}^{-1}$, where $\mathcal{O}$ represents either $E_k(\cdot)$ or $P(\cdot)$. The goal of $\mathcal{B}$ is to simulate a complete interaction between a specific adversary $\mathcal{A}$ and the challenger in the experiment Exp as follows:

1. **Construct Oracles:** $\mathcal{B}$ constructs the oracles $\mathsf{Con}[\mathcal{O}]$ and $\mathsf{DeCon}\left[\mathcal{O}^{-1}\right]$ based on its own access to $\mathcal{O}$ and $\mathcal{O}^{-1}$.

2. **Simulate Interaction:** $\mathcal{B}$ invokes the specific adversary $\mathcal{A}$ and runs it in the experiment Exp while acting as the challenger; $\mathcal{B}$ also provides $\mathcal{A}$ with the constructed oracles $\mathsf{Con}[\mathcal{O}]$ and

$\mathsf{DeCon}\left[\mathcal{O}^{-1}\right]$ when necessary.

3. **Output Result:** $\mathcal{B}$ outputs a bit: 1 if $\mathcal{A}$ succeeds in the experiment Exp, and 0 otherwise.

We note that for certain specific experiments Exp, where the adversary $\mathcal{A}$ is tasked with distinguishing whether the construction is based on E or P, in the final step, $\mathcal{B}$ simply outputs whatever $\mathcal{A}$ outputs. From the reduction, it is clear that

$$\begin{aligned}\mathsf{Adv}_E(\mathcal{B}) &= \left| \Pr_{k \leftarrow \{0,1\}^m}\left[\mathcal{B}^{E_k(\cdot),E_k^{-1}(\cdot)} = 1\right] - \Pr_{P \leftarrow \mathcal{P}_n}\left[\mathcal{B}^{P(\cdot),P^{-1}(\cdot)} = 1\right] \right| \\ &\geq \left| \Pr_{k \leftarrow \{0,1\}^m}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[E_k],\mathsf{DeCon}\left[E_k^{-1}\right]}\right) = 1\right] - \Pr_{k \leftarrow \{0,1\}^m}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P],\mathsf{DeCon}\left[P^{-1}\right]}\right) = 1\right] \right|.\end{aligned} \tag{3.26}$$

We also note that after replacing E_k with P, the distinguishing advantage of $\mathcal{A}$ attacking $\mathsf{Con}[E_k]$ with classical oracle access is given by

$$\left| \Pr_{k \leftarrow \{0,1\}^m}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P],\mathsf{DeCon}\left[P^{-1}\right]}\right) = 1\right] - \Pr_{P \leftarrow \mathcal{P}_n}\left[\mathsf{Exp}\left(\mathcal{A}^{P,P^{-1}}\right) = 1\right] \right| \leq \delta(q), \tag{3.27}$$

which is information-theoretic. This indicates that even if $\mathcal{A}$ possesses local quantum capabilities, as long as it has only classical oracle access, the distinguishing advantage remains bounded by $\delta(q)$.

By combining the above three equations, we obtain:

$$
\begin{aligned}
&\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\_}]}(\mathcal{A}) \\
&\leq \left| \Pr_{k \leftarrow \{0,1\}^m} \left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[E_k], \mathsf{DeCon}\left[E_k^{-1}\right]}\right) = 1 \right] - \Pr_{k \leftarrow \{0,1\}^m} \left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P], \mathsf{DeCon}\left[P^{-1}\right]}\right) = 1 \right] \right| \\
&\quad + \left| \Pr_{k \leftarrow \{0,1\}^m} \left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P], \mathsf{DeCon}\left[P^{-1}\right]}\right) = 1 \right] - \Pr_{P \leftarrow \mathcal{P}_n} \left[\mathsf{Exp}\left(\mathcal{A}^{P, P^{-1}}\right) = 1 \right] \right| \\
&\leq \mathsf{Adv}_E(\mathcal{B}) + \delta(q).
\end{aligned} \tag{3.28}
$$

By taking the maximum over adversaries $\mathcal{A}$, which is equivalent to taking the maximum over adversaries $\mathcal{B}$, it follows that:

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\_}]}(q, t) &\leq \max_{\mathcal{B}} \mathsf{Adv}_E(\mathcal{B}) + \delta(q) \\
&= \mathsf{Adv}^{\mathsf{SPRP\text{-}PQ}}_E(q', t) + \delta(q).
\end{aligned}
$$

□

Remark. The current statement applies directly only to constructions Con based on a block cipher E with a single key. For constructions where E uses multiple (say c) independently sampled keys, the argument can be extended with minor modifications:

$$
\mathsf{Adv}^{\mathsf{Exp}}_{\mathsf{Con}[E_{\_}]}(q, t) \leq c \cdot \mathsf{Adv}^{\mathsf{SPRP}}_E(q', t) + \delta(q) \Longrightarrow \mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_{\_}]}(q, t) \leq c \cdot \mathsf{Adv}^{\mathsf{SPRP\text{-}PQ}}_E(q', t) + \delta(q).
$$

We also examine the security of $\mathsf{Con}[E_{\_}]$ in the QICM. As discussed in Section 3.4, in this model, E is treated as an ideal cipher, and $\mathsf{Con}[E_{\_}]$ is constructed using such E. Unlike the previous version, here, the challenger Chall and Con are only granted **classical** oracle access to the *ideal cipher* $E(\cdot, \cdot)$, rather than access to the full scripts as in the previous version. The number of queries they make is recorded as q'_C. The adversary $\mathcal{A}$, however, is granted **quantum** oracle access to $E(\cdot,\cdot)$, recorded as q_Q, in addition to **classical** oracle access to $\mathsf{Con}[E_k]$ and $\mathsf{DeCon}\left[E_k^{-1}\right]$, recorded as q_C. It is important to note that, in this model, we consider adversaries with unbounded computational power. Thus, we are effectively analyzing the security of $\mathsf{Con}[E_\_]$ in the post-quantum model.

Theorem 3.7.9. *Let* Exp *be a security experiment, and let* Con *be a construction as defined before. Then the post-quantum security of* $\mathsf{Con}[E_\_]$ *in the ideal cipher model is given by:*

$$\begin{aligned}\mathit{Adv}^{\textit{Exp-PQ-IC}}_{\textit{Con}[E_\_]}(q_C, q_Q) &\leq \mathit{Adv}^{\textit{SPRP-PQ-IC}}_{E}(q'_C, q_Q) + \delta(q_C) \\ &= \frac{q_Q^2}{2^m} + \delta(q_C)\end{aligned}$$

where q_C, q'_C *and* q_Q *is defined as above;* t *denote the time resources available to quantum adversaries in the experiments* Exp*.* $\delta(q_C)$ *is the classical-model info-theoretic security of* Con *instantiated with an ideal block cipher.*

Proof. Suppose $\mathcal{A}$ is a quantum adversary attacking $\mathsf{Con}[E_\_]$ in a specified experiment Exp in the quantum ideal cipher model. The corresponding security is:

$$\begin{aligned}&\mathsf{Adv}_{\mathsf{Con}[E_\_]}(q_C, q_Q) \\ &= \max_{\mathcal{A}} \left| \Pr_{\substack{k \leftarrow \{0,1\}^m \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[E_k], \mathsf{DeCon}\left[E_k^{-1}\right], E}\right) = 1 \right] - \Pr_{\substack{P \leftarrow \mathcal{P}_n \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathsf{Exp}\left(\mathcal{A}^{P, P^{-1}, E}\right) = 1 \right] \right|.\end{aligned}$$

We note that the presence of an additional oracle $E(\cdot,\cdot)$ does not affect the validity of Equation 3.26. Additionally, Equation 3.27 also remains valid in the presence of the extra oracle E, as the oracles $\mathsf{Con}[P]$, $\mathsf{DeCon}\left[P^{-1}\right]$, P, P^{-1} operate independently from E:

$$\Pr_{\substack{k\leftarrow\{0,1\}^m\\E\leftarrow\mathcal{E}(m,n)}}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P],\mathsf{DeCon}\left[P^{-1}\right],E}\right)=1\right]=\Pr_{k\leftarrow\{0,1\}^m}\left[\mathsf{Exp}\left(\mathcal{A}^{\mathsf{Con}[P],\mathsf{DeCon}\left[P^{-1}\right]}\right)=1\right]$$

$$\Pr_{\substack{P\leftarrow\mathcal{P}_n\\E\leftarrow\mathcal{E}(m,n)}}\left[\mathsf{Exp}\left(\mathcal{A}^{P,P^{-1},E}\right)=1\right]=\Pr_{P\leftarrow\mathcal{P}_n}\left[\mathsf{Exp}\left(\mathcal{A}^{P,P^{-1}}\right)=1\right].$$

Therefore Equation 3.28 holds in the quantum ideal cipher model. Then combining with the security of the ideal cipher in the quantum ideal cipher model as stated in Theorem 3.7.7:

$$\begin{aligned}\mathsf{Adv}^{\mathsf{Exp\text{-}PQ\text{-}IC}}_{\mathsf{Con}[E_\cdot]}(q_C,q_Q)&=\mathsf{Adv}^{\mathsf{SPRP\text{-}PQ\text{-}IC}}_{E}(q'_C,q_Q)+\delta(q_C)\\&\le\frac{q_Q^2}{2^n}+\delta(q_C).\end{aligned}$$

□

The theorem also applies to constructions Con whose classical security can be reduced to the block cipher's PRP security. This yields the following bound:

$$\mathsf{Adv}^{\mathsf{Exp\text{-}PQ}}_{\mathsf{Con}[E_\cdot]}(q,t)\le\mathsf{Adv}^{\mathsf{PRP\text{-}PQ}}_{E}(q',t')+\delta(q).$$

Similarly, in QICM, we can derive:

$$\mathsf{Adv}^{\mathsf{Exp\text{-}PQ\text{-}IC}}_{\mathsf{Con}[E_\cdot]}(q_Q,q_C)\le\frac{q_Q^2}{2^n}+\delta(q_C).$$

Applications of Theorem 3.7.8 and Theorem 3.7.9. Let m and n be positive integers. Let $E:\{0,1\}^m\times\{0,1\}^n\to\{0,1\}^n$ be a block cipher. A summary of applications of the general theorem for

Block Cipher Modes	Post-Quantum Security Bound
CBC	$\frac{q_Q^2\ell^2}{2^m} + \frac{q_C^2\ell^2}{2^n}$
ECBC	$\frac{2q_Q^2\ell^2}{2^m} + \frac{4q_C^2\ell^2}{2^n}$
CMAC	$\frac{(q_Q\ell+1)^2}{2^m} + \frac{5(\ell^2+1)q^2}{2^n}$
GCM	$\frac{q_Q^2\ell^2}{2^m} + \frac{(\sigma+q_C+q_C'+1)^2}{2^{n+1}} + \frac{\sigma+q_C+q_C'}{2^{n-1}} + \frac{q_C'(\ell+1)}{2^s}$
GCM-SST	$\frac{q_Q^2\ell^2}{2^m} + \frac{(\sigma+3(q_C+q_C'))^2}{2^{n+1}} + \frac{q_C'\ell}{2^n} + \frac{q_C'}{2^s}$
LRW	$\frac{q_Q^2}{2^m} + \frac{q_C^2}{2^n}$
XEX2	$\frac{q_Q^2}{2^m} + \frac{3q_C^2}{2^n}$

Table 3.2: Post-quantum security bound of block cipher modes in the QICM. ℓ is the block length for every query.

block cipher mode post-quantum security is given in Table 3.2.

CBC [EMST78].

- Construction: $\mathsf{Con}[E_k] = \mathsf{CBC}_k^E(x_1, \ldots, x_\ell) = E_k(E_k(\ldots E_k(E_k(x_1) \oplus x_2) \oplus \ldots) \oplus x_\ell)$

- Classical security [KL07]: For all probabilistic adversaries $\mathcal{A}$, it holds that:

$$\begin{aligned}\mathsf{Adv}_{\mathsf{CBC}}^{\mathsf{PRF}}(q,t) &:= \max_{\mathcal{A}} \left| \Pr_{k \leftarrow \{0,1\}^m}[\mathcal{A}^{\mathsf{CBC}[E_k]} = 1] - \Pr_f[\mathcal{A}^f = 1] \right| \\ &\leq \mathsf{Adv}_E^{\mathsf{PRP}}(q\ell, t) + \frac{q^2\ell^2}{2^n},\end{aligned}$$

  where f is chosen uniformly from the set of functions mapping $(\{0,1\}^n)^\ell$ to $\{0,1\}^n$, q is the number of queries to CBC, t is the time complexity and ℓ is the number of blocks of the longest query input.

- Post-quantum security: The post-quantum security of a construction based on another concrete block cipher, such as AES, in the PQ model is closely related to its classical security since no second oracle is available to the adversary. The equivalence holds with $\mathsf{Adv}_E^{\mathsf{PRF}}$ replaced by $\mathsf{Adv}_E^{\mathsf{PRP\text{-}PQ}}$ as stated in Theorem 3.7.8:

$$\mathsf{Adv}_{\mathsf{CBC}}^{\mathsf{PRF\text{-}PQ}}(q,t) \leq \mathsf{Adv}_E^{\mathsf{PRP\text{-}PQ}}(q\ell,t) + \frac{q^2\ell^2}{2^n}.$$

- Post-quantum security in QICM: By applying Theorem 3.7.9, we get

$$\begin{aligned}\mathsf{Adv}_{\mathsf{CBC}}^{\mathsf{PRF\text{-}PQ\text{-}IC}}(q_C,q_Q) &= \max_{\mathcal{A}}\left|\Pr_{k\leftarrow\{0,1\}^m}\left[\mathcal{A}^{\mathsf{CBC}[E_k],E}=1\right]-\Pr_f\left[\mathcal{A}^{P,E}=1\right]\right|\\ &\leq \frac{q_Q^2\ell^2}{2^m}+\frac{q_C^2\ell^2}{2^n}.\end{aligned}$$

Similarly as explained in [KL07], when the tag is set to be $\mathsf{CBC}_k(m)$, this is exactly CBC -MAC, and post-quantum security of CBC implies that basic CBC -MAC is post-quantum secure for messages of any fixed length. However, it is not inherently secure for variable-length messages. To address this limitation, *encrypt-last-block* offers a method to extend CBC -MAC for use with variable-length inputs. It is formally called Encrypt-last-block CBC-MAC, i.e., ECBC -MAC.

ECBC [PR00].

- Construction: $\mathsf{Con}[E_k](x) = \mathsf{ECBC}_{k_1,k_2}^E(x) = E_{k_2}(\mathsf{CBC}_{k_1}^E(x))$

- Classical security [PR00, Vau00]: For all probabilistic adversaries $\mathcal{A}$, it holds that:

$$\begin{aligned}\mathsf{Adv}_{\mathsf{ECBC}}^{\mathsf{PRF}}(q,t) &:= \max_{\mathcal{A}}\left|\Pr_{k\leftarrow\{0,1\}^m}[\mathcal{A}^{\mathsf{ECBC}[E_k]}=1]-\Pr_f[\mathcal{A}^f=1]\right|\\ &\leq 2\mathsf{Adv}_E^{\mathsf{PRP}}(q\ell,t)+\frac{4q^2\ell^2}{2^n}.\end{aligned}$$

where f is chosen uniformly from the set of functions mapping $(\{0,1\}^n)^\ell$ to $\{0,1\}^n$, q is the number of queries to CBC, t is the time complexity and ℓ is the number of blocks of the longest query input.

- Post-quantum security: By applying Theorem 3.7.8, we get

$$\mathsf{Adv}^{\mathsf{PRF\text{-}PQ}}_{\mathsf{CBC}}(q,t) \le 2\mathsf{Adv}^{\mathsf{PRP\text{-}PQ}}_{E}(q\ell,t) + \frac{4q^2\ell^2}{2^n}.$$

- Post-quantum security in QICM: By applying Theorem 3.7.9, we get

$$\begin{aligned}\mathsf{Adv}^{\mathsf{PRF\text{-}PQ\text{-}IC}}_{\mathsf{CBC}}(q_C,q_Q) &= \max_{\mathcal{A}}\left|\Pr_{k\leftarrow\{0,1\}^m}\left[\mathcal{A}^{\mathsf{CBC}[E_k],E}=1\right] - \Pr_{f}\left[\mathcal{A}^{P,E}=1\right]\right| \\ &\le \frac{2q_Q^2\ell^2}{2^m} + \frac{4q_C^2\ell^2}{2^n}.\end{aligned}$$

CMAC [IK03].

- Construction: $\mathsf{Con}[E_k](x) = \mathsf{CMAC}^E_k(x) = E_{k_1}(\mathsf{CBC}^E_{k_1}(x_1\|\ldots\|x_{\ell-1}) \oplus x_\ell \oplus k_2)$

- Classical security [IK03]: For all probabilistic adversaries $\mathcal{A}$, it holds that:

$$\mathsf{Adv}^{\mathsf{Forgery}}_{\mathsf{CMAC}}(q,t) \le \mathsf{Adv}^{\mathsf{PRP}}_{E}(q\ell+1, t+O(q\ell)) + \frac{5(\ell^2+1)q^2}{2^n}.$$

where q is the number of queries in the Forgery experiment, t is the time complexity and ℓ is the number of blocks of the longest query input.

- Post-quantum security: By applying Theorem 3.7.8, we get

$$\mathsf{Adv}^{\mathsf{Forgery\text{-}PQ}}_{\mathsf{CMAC}}(q,t) \leq \mathsf{Adv}^{\mathsf{PRP\text{-}PQ}}_{E}(q\ell+1, t+O(q\ell)) + \frac{5(\ell^2+1)q^2}{2^n}.$$

- Post-quantum security in QICM: By applying Theorem 3.7.9, we get

$$\mathsf{Adv}^{\mathsf{Forgery\text{-}PQ\text{-}IC}}_{\mathsf{CMAC}}(q_C, q_Q) \leq \frac{(q_Q\ell+1)^2}{2^m} + \frac{5(\ell^2+1)q^2}{2^n}.$$

GCM [MV04].

- Classical security [MV04,IOM12,NOMI15]: We begin by noting that the oracle in the Forgery experiment for GCM also includes a decryption oracle, denoted as GCM$\pm$. For all probabilistic adversaries $\mathcal{A}$, it holds:

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{GCM}}(q,t) &\leq \mathsf{Adv}^{\mathsf{PRP}}_{E}(q,t) + \frac{(\sigma+q+1)^2}{2^{n+1}} + \frac{\sigma+q}{2^{n-1}} \\
\mathsf{Adv}^{\pm\mathsf{Forgery}}_{\mathsf{GCM}}(q,t) &\leq \mathsf{Adv}^{\mathsf{PRP}}_{E}(q,t) + \frac{(\sigma+q+q'+1)^2}{2^{n+1}} + \frac{\sigma+q+q'}{2^{n-1}} + \frac{q'(\ell+1)}{2^s}.
\end{aligned}$$

where q, q' are the number of queries made to the encryption and decryption oracles, respectively; t is the time complexity and s is the authentication tag size. Additionally ℓ is the maximum number of blocks in the input-data for any individual query, and σ is the total number of blocks of input-data across all queries.

From [Rog02], an AEAD-scheme Π is said to be "secure" if $\mathsf{Adv}^{\mathsf{PRF}}_{\Pi}$ and $\mathsf{Adv}^{\mathsf{Forgery}}_{\Pi}$ are "small" for any "reasonable" adversary $\mathcal{A}$. GCM is a widely used secure AEAD-scheme.

- Post-quantum security: By applying Theorem 3.7.8, we get

$$\mathsf{Adv}^{\mathsf{PRF\text{-}PQ}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\mathsf{PRP\text{-}PQ}}_{E}(q,t) + \frac{(\sigma+q+1)^2}{2^{n+1}} + \frac{\sigma+q}{2^{n-1}}$$
$$\mathsf{Adv}^{\pm\mathsf{Forgery\text{-}PQ}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\mathsf{PRP\text{-}PQ}}_{E}(q,t) + \frac{(\sigma+q+q'+1)^2}{2^{n+1}} + \frac{\sigma+q+q'}{2^{n-1}} + \frac{q'(\ell+1)}{2^s}.$$

- Post-quantum security in QICM: By applying Theorem 3.7.9, we get

$$\mathsf{Adv}^{\mathsf{PRF\text{-}PQ\text{-}IC}}_{\mathsf{GCM}}(q_C,q_Q) \leq \frac{q_Q^2\ell^2}{2^m} + \frac{(\sigma+q_C+1)^2}{2^{n+1}} + \frac{\sigma+q_C}{2^{n-1}}$$
$$\mathsf{Adv}^{\pm\mathsf{Forgery\text{-}PQ\text{-}IC}}_{\mathsf{GCM}}(q_C,q_Q) \leq \frac{q_Q^2\ell^2}{2^m} + \frac{(\sigma+q_C+q'_C+1)^2}{2^{n+1}} + \frac{\sigma+q_C+q'_C}{2^{n-1}} + \frac{q'_C(\ell+1)}{2^s}.$$

GCM-SST [CMM23, IJMM24]. When analyzing security of GCM-SST, the experiment considers Nonce-Misuse Resilience for both privacy and authenticity, denoted as nml-PRF and nml-Forgery, respectively. The nml security notions give stronger notions of security, where GCM fails to meet the requirements, but GCM-SST satisfies them.

- Classical security [IJMM24]: For all probabilistic adversaries $\mathcal{A}$, it holds that:

$$\mathsf{Adv}^{\mathsf{nml\text{-}PRF}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\mathsf{PRP}}_{E}(q,t) + \frac{(\sigma+3q)^2}{2^{n+1}}$$
$$\mathsf{Adv}^{\mathsf{nml\text{-}Forgery}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\mathsf{PRP}}_{E}(q,t) + \frac{(\sigma+3(q+q'))^2}{2^{n+1}} + \frac{q'\ell}{2^n} + \frac{q'}{2^s}.$$

  where q, q' are the number of queries made to the encryption and decryption oracles, respectively; t is the time complexity and s is the authentication tag size. Additionally ℓ is the maximum number of blocks in the input-data for any individual query, and σ is the total number of blocks of input-data across all queries.

- Post-quantum security: By applying Theorem 3.7.8, we get

$$\mathsf{Adv}^{\text{nml-PRF-PQ}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\text{PRP-PQ}}_{E}(q,t) + \frac{(\sigma+3q)^2}{2^{n+1}}$$
$$\mathsf{Adv}^{\text{nml-Forgery-PQ}}_{\mathsf{GCM}}(q,t) \leq \mathsf{Adv}^{\text{PRP-PQ}}_{E}(q,t) + \frac{(\sigma+3(q+q'))^2}{2^{n+1}} + \frac{q'\ell}{2^n} + \frac{q'}{2^s}.$$

- Post-quantum security in QICM: By applying Theorem 3.7.9, we get

$$\mathsf{Adv}^{\text{nml-PRF-PQ-IC}}_{\mathsf{GCM}}(q_C,q_Q) \leq \frac{q_Q^2\ell^2}{2^m} + \frac{(\sigma+3q_C)^2}{2^{n+1}}$$
$$\mathsf{Adv}^{\text{nml-Forgery-PQ-IC}}_{\mathsf{GCM}}(q_C,q_Q) \leq \frac{q_Q^2\ell^2}{2^m} + + \frac{(\sigma+3(q_C+q'_C))^2}{2^{n+1}} + \frac{q'_C\ell}{2^n} + \frac{q'_C}{2^s}.$$

3.7.4 Security of LRW and XEX2 using the general theorem

The result from Theorem 3.7.8 extends to tweakable block cipher constructions. As a direct result, the SPRP-PQ security of LRW also follows from its classical security bound (Theorem 3.6.7).

Corollary 3.7.10. *Let LRW be as defined in Definition 3.6.6, constructed from a block cipher E and an ε-XOR universal hash function family h. Then the post-quantum security of LRW is given by:*

$$\mathit{Adv}^{\mathit{SPRP\text{-}PQ}}_{\mathit{LRW}}(q,t) \leq \mathit{Adv}^{\mathit{SPRP\text{-}PQ}}_{E}(q,t) + q^2\varepsilon.$$

We analyze the SPRP-PQ security of LRW. The LRW construction is built upon such E, and consequently, the adversary $\mathcal{A}$ is granted quantum oracle access to $E(\cdot,\cdot)$.

Corollary 3.7.11. *Let LRW be as defined in Definition 3.6.6. Assuming h is XOR-universal, the post-*

quantum security of LRW *in QICM is given by:*

$$\mathit{Adv}^{\text{SPRP-PQ-IC}}_{\mathsf{LRW}}(q_C, q_Q) = \left| \Pr_{\substack{k \leftarrow \{0,1\}^m; k' \leftarrow \{0,1\}^\kappa \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \mathsf{LRW}_{k,k'}[E], |\pm E\rangle} = 1\right] - \Pr_{\substack{\widetilde{\Pi} \leftarrow \mathcal{E}(T,n); \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \widetilde{\Pi}, |\pm E\rangle} = 1\right] \right|$$
$$\leq \frac{q_Q^2}{2^m} + \frac{q_C^2}{2^n}.$$

Similarly to the classical case, the $\mathsf{SPRP\text{-}PQ}$ security of $\mathsf{XEX2}$ reduces to that of LRW.

Corollary 3.7.12. *Let* $\mathsf{XEX2}$ *be as defined in Definition 3.6.8. Then the post-quantum security of* $\mathsf{XEX2}$ *in QICM is given by:*

$$\mathit{Adv}^{\text{SPRP-PQ}-\text{IC}}_{\mathsf{XEX2}}(q_C, q_Q) \leq \frac{2q_Q^2}{2^m} + \frac{q_C^2}{2^n - 1}.$$

Proof. Immediate consequence of Proposition 3, Lemma 3.6.10 and Theorem 3.7.10. □

3.7.5 Comparison of LRW security bound

Recall that the bound of LRW proved in Theorem 3.6.12 is:

$$\frac{6q_C^2}{2^n} + \frac{4 \cdot \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right)}{2^{(m+n)/2}}.$$

The comparison between the bounds from Corollary 3.7.11 and Theorem 3.6.12 is summarized earlier in Table 3.3. A detailed discussion follows below, omitting multiplicative constants. Note that in applying the general theorem, we do not need to assume that h is uniform.

- When $m \gg n$: The two bounds converge to $\frac{q_C^2}{2^n}$, and it aligns with the classical optimal bound (and classical matching attack) in Theorem 3.6.7.

- When $q_C \gg q_Q$: The bounds also align at $\frac{q_C^2}{2^n}$, and it aligns with the classical optimal bound (and classical matching attack) in Theorem 3.6.7.

- When $q_C \ll q_Q$: This scenario is more practical since online queries (q_C) are typically more scarce resources. Under these conditions, the bound in Theorem 3.6.12 simplifies to approximately $\frac{q_Q\sqrt{q_C}}{2^{(m+n)/2}}$, whereas the bound in Corollary 3.7.11 becomes $\frac{q_Q^2}{2^m}$. The bound from Theorem 3.6.12 is more favorable in this case.

  Notably, this bound aligns with two quantum complete key recovery attacks: the offline-Simon attack with the trade-off $q_Q^2 q_C = 2^{(m+n)/2}$, and the Grover+Kuwakado–Morii attack with $q_Q = 2^{m/2+n/3}$ and $q_C = 2^{n/3}$.

- When $q_C \approx q_Q \approx q$: The bound in Theorem 3.6.12 simplifies to approximately $\frac{q^2}{2^n} + \frac{q^{3/2}}{2^{(m+n)/2}}$, where the second term corresponds to the offline-Simon attack described in Section 3.6.3, assuming $m \leq 2n$. Furthermore, when $m \approx n$, both bounds reduce to $\frac{q^2}{2^n}$, aligning with the classical collision attack.

A summary is provided in Table 3.3. From the case analysis, the bound in Theorem 3.6.12 is typically tighter, with matching attacks. Nonetheless, the analysis in Theorem 3.7.10 and Corollary 3.7.11 provides a simpler and more general method for deriving a post-quantum security bound.

3.8 Additional proofs and constructions

3.8.1 Proof of the new resampling lemma

Lemma 3.4.1. *Let $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ be a quantum distinguisher in the following experiment:*

Scenarios	Classical	Q1 from GT	Q1 from HT
$m \gg n$	$\frac{q_C^2}{2^n}$	$\frac{q_C^2}{2^n}$	$\frac{q_C^2}{2^n}$
$q_C \gg q_Q$	$\frac{q_C^2}{2^n}$	$\frac{q_C^2}{2^n}$	$\frac{q_C^2}{2^n}$
$q_C \ll q_Q$	$\frac{q_Q}{2^m}$	$\frac{q_Q^2}{2^m}$	$\frac{q_Q\sqrt{q_C}}{2^{(m+n)/2}}$
$q_C \approx q_Q = q$	$\frac{q^2}{2^n} + \frac{q}{2^m}$	$\frac{q^2}{2^n} + \frac{q^2}{2^m}$	$\frac{q^2}{2^n} + \frac{q^{3/2}}{2^{(m+n)/2}}$

Table 3.3: Comparison of the classical bound (Theorem 3.6.7), the post-quantum security bounds of LRW using the General Theorem (GT, Corollary 3.7.11) and the Hybrid Technique (HT, Theorem 3.6.12). In the classical case, q_Q denotes the number of classical queries made to E, denoted for straightforward comparison.

Phase 1: *Choose uniform $E \in \mathcal{E}(m,n)$ and give $\mathcal{D}$ quantum access to E and E^{-1}. Then $\mathcal{D}_0$ chooses and outputs a distribution D over $\{0,1\}^{m+2n}$.*

Phase 2: *Choose $k_0 \in \{0,1\}^m$ and $s_0, s_1 \in \{0,1\}^n$ according to D. Let $E^{(0)} = E$ and define $E^{(1)} : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ by*

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \textit{if } k^* \neq k_0 \\ E_{k^*} \circ \mathsf{swap}_{s_0, s_1}(x) & \textit{if } k^* = k_0\,. \end{cases}$$

A uniform bit $b \in \{0,1\}$ is chosen, and $\mathcal{D}$ is given k_0, s_0, s_1, and quantum access to $E^{(b)}$. Then $\mathcal{D}$ outputs a guess b'.

For a probability distribution D on $\{0,1\}^{m+2n}$, we define

$$\epsilon = \max_{(k_0^*, s_0^*, s_1^*) \in \{0,1\}^{m+2n}} D(k_0, s_0, s_1)$$

Then for any $\mathcal{D}$ making at most q queries to E in phase 1,

$$|\Pr[\mathcal{D} \textit{ outputs 1} \mid b = 1] - \Pr[\mathcal{D} \textit{ outputs 1} \mid b = 0]| \leq 4\sqrt{2^n \cdot q \cdot \epsilon}\,.$$

Proof. The proof of this lemma generalizes Lemma 3 from [Hos25], which in turn builds on techniques from [ABKM22]. Thus, part of the proof is borrowed from these two works.

Let $F = F_{0^m} F_{0^{m-1}1}, \ldots, F_{1^m}$ be the internal register of a superposition oracle for an ideal cipher, where each $F_k = F_{k,0^n}, \ldots, F_{k,1^n}$ is a database register for a random permutation. Each F_k is initialized in the initial state $|\phi_0\rangle$ for a random permutation π, namely,

$$|\phi_0\rangle_{F_k} = (2^n!)^{-1/2} \sum_{\pi \in \mathcal{P}(n)} |\pi\rangle_{F_k}\,.$$

Moreover, define $|\Phi_0\rangle_F = |\phi_0\rangle^{\otimes 2^m}$.

Then, quantum queries to the ideal cipher can be evaluated via the unitary operators O and O^{inv}:

$$\begin{aligned} O_{KXYF}\ |k\rangle_K |x\rangle_X |y\rangle_Y |\pi\rangle_{F_k} &= |k\rangle_K |x\rangle_X |y \oplus \pi(x)\rangle_Y |\pi\rangle_{F_k} \\ O^{\mathrm{inv}}_{KXYF} |k\rangle_K |x\rangle_X |y\rangle_Y |\pi\rangle_{F_k} &= |k\rangle_K \left|x \oplus \left(\oplus_{x':\pi(x')=y}\, x'\right)\right\rangle_X |y\rangle_Y |\pi\rangle_{F_k}\,. \end{aligned}$$

Note that O and O^{inv} act only on the register F_k, and act as the identity on all other F-registers. By analogy to the proof of [ABKM22, Hos25], define the projectors

$$(P_{k_0,s_0,s_1})_{KX} = \begin{cases} \mathbb{1} & s_0 = s_1 \\ \mathbb{1} - |k_0\rangle\langle k_0| \otimes (|s_0\rangle\langle s_0| + |s_1\rangle\langle s_1|)_X & s_0 \neq s_1 \end{cases}$$

and

$$\left(P^{\mathrm{inv}}_{k_0,s_0,s_1}\right)_{KYF}$$
$$= \begin{cases} \mathbb{1} & s_0 = s_1 \\ |k_0\rangle\langle k_0|_K \otimes \sum_{y\in\{0,1\}^n} |y\rangle\langle y|_Y \otimes (\mathbb{1} - |y\rangle\langle y|)^{\otimes 2}_{F_{k_0,s_0}F_{k_0,s_1}} & s_0 \neq s_1. \end{cases}$$

Moreover, let $\bar{P}_{k_0,s_0,s_1} = \mathbb{1} - P_{k_0,s_0,s_1}$ and $\bar{P}^{\mathrm{inv}}_{k_0,s_0,s_1} = \mathbb{1} - P^{\mathrm{inv}}_{k_0,s_0,s_1}$. Define the swap operator

$$\S_{AB} |x\rangle_A |y\rangle_B = |y\rangle_A |x\rangle_B$$

.

Without loss of generality, we assume (as in [Hos25]) that $\mathcal{D}_0$ makes exactly q quantum queries and does not perform intermediate measurements. The distribution D is chosen by $\mathcal{D}_0$ as follows.

1. At the beginning, $\mathcal{D}_0$ sets a specific part of the offline register E, which we refer to as register Z, to be $|0\rangle_Z$. Register Z is untouched until $\mathcal{D}_0$ has made all its q queries.
2. After making q quantum queries, $\mathcal{D}_0$ applies a unitary to all its internal registers, including Z.
3. Then, it measures register Z to obtain the distribution D.

We denote the projector $|D\rangle\langle D|_Z$ as P_D. Let $|\psi_0\rangle = |\psi_0'\rangle_{KXYE} |\phi_0\rangle_F$ be the initial state of $\mathcal{D}_0$. Let $|\psi\rangle$ be the quantum state right before the measurement to choose the distribution D. Then,

$$|\psi\rangle = U_q O_q \cdots U_1 O_1 |\psi_0\rangle , \tag{3.29}$$

holds, where each O_i (which is either O_{KXYF} or O^{inv}_{KXYF}) is the unitary operator corresponding to the i^{th} quantum query, and U_i is a unitary operator acting on registers XYE that corresponds to some offline computation by $\mathcal{D}_0$. In addition, for arbitrary $k_0 \in \{0,1\}^m$ and $s_0, s_1 \in \{0,1\}^n$, let

$$|\psi_{\text{good}}(k_0, s_0, s_1)\rangle := z \cdot U_q O_q P^q_{k_0,s_0,s_1} \cdots U_1 O_1 P^1_{k_0,s_0,s_1} |\psi_0\rangle$$
$$|\psi_{\text{bad}}(k_0, s_0, s_1)\rangle := |\psi\rangle - |\psi_{\text{good}}(k_0, s_0, s_1)\rangle ,$$

where

$$P^i_{k_0,s_0,s_1} := \begin{cases} (P_{k_0,s_0,s_1})_X & \text{if } O_i = O_{KXYF} \\ (P^{\text{inv}}_{k_0,s_0,s_1})_{KYF} & \text{if } O_i = O^{\text{inv}}_{KXYF}, \end{cases}$$

and z is a complex number such that $|z| = 1$ and $\langle \psi \mid \psi_{\text{good}}(k_0, s_0, s_1)\rangle \in \mathbb{R}_{\geq 0}$. As in [ABKM22, Hos25], the vector $|\psi_{\text{good}}(k_0, s_0, s_1)\rangle$ is invariant under the action of $\S_{F_{k_0,s_0},F_{k_0,s_1}}$, that is,

$$\S_{F_{k_0,s_0},F_{k_0,s_1}} |\psi_{\text{good}}(k_0, s_0, s_1)\rangle = |\psi_{\text{good}}(k_0, s_0, s_1)\rangle . \tag{3.30}$$

Moreover, let

$$\epsilon_i(k_0, s_0, s_1) := \left\| \bar{P}^i_{k_0,s_0,s_1} U_{i-1} O_{i-1} \cdots U_1 O_1 |\psi_0\rangle \right\|_2^2 ,$$

where $\bar{P}^i_{k_0,s_0,s_1} := \mathbb{1} - P^i_{k_0,s_0,s_1}$.

Recall that, at the end of $\mathcal{D}_0$, the register Z (which is a part of register E for offline computation) is measured to choose a distribution D. The (pure) state just before the measurement is $|\psi\rangle$ of

Equation 3.29. By the measurement, $|\psi\rangle$ changes to the mixed state

$$\sum_{D \in \mathcal{S}} P_D \left|\psi\right\rangle \left\langle\psi\right| P_D.$$

After that, k_0, s_0 and s_1 are sampled, and the state changes to

$$\rho_0 := \sum_{\substack{D \in \mathcal{S} \\ (k_0,s_0,s_1) \sim D}} D(k_0, s_0, s_1)\ P_D \left|\psi\right\rangle \left\langle\psi\right|_{KXYEF} P_D \otimes \left|k_0, s_0, s_1\right\rangle \left\langle k_0, s_0, s_1\right|_C ,$$

where C is an additional register.

If $b = 1$ and the values of the permutation are swapped on s_0 and s_1, the state further changes to

$$\rho_1 := \sum_{\substack{D \in \mathcal{S} \\ (k_0,s_0,s_1) \sim D}} D(k_0, s_0, s_1)\ \S_{F_{k_0,s_0}, F_{k_0,s_1}} P_D \left|\psi\right\rangle \left\langle\psi\right| P_D \S_{F_{k_0,s_0}, F_{k_0,s_1}}$$
$$\otimes \left|k_0, s_0, s_1\right\rangle \left\langle k_0, s_0, s_1\right| .$$

From the standard argument for distinguishing advantages,

$$\left|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]\right| \leq \mathsf{TD}(\rho_0, \rho_1),$$

where TD is the trace distance.

We observe that the state ρ_0 can also be produced from $|\psi\rangle$ as follows.

1. Without measuring register Z (on which the information of D is written), create the superposi-

tion of k_0, s_0 and s_1 of the form

$$\sum_{(k_0,s_0,s_1)\sim D} \sqrt{D(k_0, s_0, s_1)}\ \ |D\rangle_Z\, |k_0, s_0, s_1\rangle_C\, .$$

For each D, obtaining the (pure) state

$$|\xi_0\rangle := \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\sim D}} \sqrt{D(k_0, s_0, s_1)}\, P_D\, |\psi\rangle_{KXYEF} \otimes |k_0, s_0, s_1\rangle_C\, .$$

2. Measure registers Z and C to obtain the classical information of D, k_0, s_0, s_1.

Similarly, the state ρ_1 can be produced by measuring the registers Z and C of the pure state

$$|\xi_1\rangle := \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\sim D}} \sqrt{D(k_0, s_0, s_1)}\ \S_{F_{k_0,s_0},F_{k_0,s_1}} P_D\, |\psi\rangle_{KXYEF} \otimes |k_0, s_0, s_1\rangle_C\, .$$

Since measurements decrease the trace distance between two states, and the trace distance between the two pure states $|\xi_0\rangle\langle\xi_0|$ and $|\xi_1\rangle\langle\xi_1|$ is upper bounded by the norm of the difference $(|\xi_0\rangle - |\xi_1\rangle)$, it holds that

$$\mathsf{TD}(\rho_0, \rho_1) \ \leq\ \mathsf{TD}(|\xi_0\rangle\langle\xi_0|\,, |\xi_1\rangle\langle\xi_1|) \ \leq\ \|\, |\xi_0\rangle - |\xi_1\rangle\, \|_2.$$

In addition, we have

$$
\begin{aligned}
\|\,|\xi_0\rangle - |\xi_1\rangle\,\|_2^2 &\overset{(i)}{=} \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\sim D}} D(k_0,s_0,s_1)\,\left\|P_D\big(1-\mathsf{Swap}_{F_{k_0,s_0},F_{k_0,s_1}}\big)\,|\psi\rangle\right\|_2^2 \\
&\overset{(ii)}{=} \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\sim D}} D(k_0,s_0,s_1)\,\Big\|P_D\Big(|\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle \\
&\qquad\qquad - \mathsf{Swap}_{F_{k_0,s_0},F_{k_0,s_1}}\,|\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\Big)\Big\|_2^2 \\
&\overset{(iii)}{\leq} 4 \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\in\{0,1\}^{m+2n}}} D(k_0,s_0,s_1)\,\|P_D\,|\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\|_2^2 \\
&\overset{(iv)}{\leq} 4 \sum_{\substack{D\in\mathcal{S}\\(k_0,s_0,s_1)\in\{0,1\}^{m+2n}}} \varepsilon\,\|P_D\,|\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\|_2^2 \\
&\overset{(v)}{=} 4\varepsilon\cdot 2^{m+2n}\,\mathbb{E}_{(k_0,s_0,s_1)\leftarrow\{0,1\}^{m+2n}}\left[\|\,|\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\,\|_2^2\right].
\end{aligned} \tag{3.31}
$$

where we used the fact that $\left[P_D, \S_{F_{k_0,s_0},F_{k_0,s_1}}\right] = 0$ [4] for (i) and (iii), Equation 3.30 for (ii), and the fact that $\sum_D P_D = 1$ for (v).

Next, we bound $\mathbb{E}_{(k_0,s_0,s_1)\leftarrow\{0,1\}^{m+2n}}\left[\||\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\|_2^2\right]$. First, using the gentle measurement lemma as in [ABKM22, Section 4.2], we get

$$
\||\psi_{\mathrm{bad}}(k_0,s_0,s_1)\rangle\|_2^2 \;\leq\; 2\sum_{i=1}^{q}\epsilon_i(k_0,s_0,s_1). \tag{3.32}
$$

[4] P_D acts only on (a part of) register E while $\mathsf{Swap}_{F_{k_0,s_0},F_{k_0,s_1}}$ acts only on register F.

For any normalized state $|\psi\rangle_{KXE}$, let

$$|\psi\rangle_{KXE} = \sum_{\substack{k\in\{0,1\}^m \\ x\in\{0,1\}^n}} |k\rangle_K |x\rangle_X \otimes |\psi_{kx}\rangle_E$$

be its expansion in the computational basis on X. By the definition of ϵ, following [ABKM22, Section 4.2], we obtain

$$\begin{aligned}
&\mathbb{E}_{(k_0,s_0,s_1)\leftarrow\{0,1\}^{m+2n}} \left[\| \left(P_{k_0,s_0,s_1}\right)_{KX} |\psi\rangle_{KXE} \|_2^2\right] \\
&= \sum_{\substack{k\in\{0,1\}^m \\ x\in\{0,1\}^n}} \| |\psi_{kx}\rangle \|_2^2 \mathbb{E}_{(k_0,s_0,s_1)\leftarrow\{0,1\}^{m+2n}} \left[\| \left(P_{k_0,s_0,s_1}\right)_{KX} |k\rangle_K |x\rangle_X \|_2^2\right] \\
&= \sum_{\substack{k\in\{0,1\}^m \\ x\in\{0,1\}^n}} \| |\psi_{kx}\rangle \|_2^2 \Pr_{(k_0,s_0,s_1)\leftarrow\{0,1\}^{m+2n}} \left[(k,x) \in \{(k_0,s_0),(k_0,s_1)\}\right] \\
&\leq 2\cdot 2^{-(n+m)}.
\end{aligned}$$

Similarly, again following [ABKM22, Section 4.2]

$$\mathop{\mathbb{E}}_{(k_0,s_0,s_1)\sim D}\left[\left\|\left(\bar{P}^{\mathrm{inv}}_{k_0,s_0,s_1}\right)_{KYF} |\psi\rangle_{KYEF}\right\|_2^2\right] \leq 2\cdot 2^{-(n+m)}.$$

Then we have

$$\mathbb{E}_{(k_0,s_0,s_1)\sim D}\left[\epsilon_i(k_0,s_0,s_1)\right] \;\leq\; 2\cdot 2^{-(n+m)}. \tag{3.33}$$

Combining Equation 3.31, Equation 3.32 and Equation 3.33, we obtain the desired bound. $\square$

3.8.2 More details on post-quantum security of FX

Proof of Proposition 1.

Proposition 4 (Restatement)**.** *For any $E \in \mathcal{E}(m,n)$, $K = (k_0, k_1, k_2) \in \{0,1\}^{m+2n}$, $j \in [1, q_C]$, transcript $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ without repetition, and any $i \in \{1, \ldots, j\}$, it holds that*

$$E_{k_0}^{T_j,K}(x_i \oplus k_1) \oplus k_2 = y_i.$$

Proof. We prove by strong induction on i. We start by the base case $i = j$.

$$E_{k_0}^{T_j,K}(x_j \oplus k_1) = \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ E_{k_0}^{T_{j-1},K}(x_j \oplus k_1) = y_j \oplus k_2.$$

Assume for all $i \in \{r, \ldots, j\}$ that $E_{k_0}^{T_j,K}(x_i \oplus k_1) \oplus k_2 = y_i$. Expanding it, we obtain

$$\begin{aligned}
&E_{k_0}^{T_j,K}(x_i \oplus k_1) \\
&= \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \ldots \circ \mathsf{swap}_{E_{k_0}^{T_{i-1},K}(x_i \oplus k_1),\, y_i \oplus k_2} \circ E_{k_0}^{T_{i-1},K}(x_i \oplus k_1) \\
&= F_i \circ E_{k_0}^{T_{i-1},K}(x_i \oplus k_1) = y_i \oplus k_2,
\end{aligned}$$

where we define

$$F_i = \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \ldots \circ \mathsf{swap}_{E_{k_0}^{T_{i-1},K}(x_i \oplus k_1),\, y_i \oplus k_2}.$$

By rearranging, we get for all $i \in \{r, \ldots, j\}$,

$$E_{k_0}^{T_{i-1},K}(x_i \oplus k_1) = F_i^{-1}(y_i \oplus k_2).$$

We now work on case $r-1$,

$$\begin{aligned}
&E_{k_0}^{T_j,K}(x_{r-1} \oplus k_1)\\
&= \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \ldots \circ \mathsf{swap}_{E_{k_0}^{T_{r-2},K}(x_{r-1} \oplus k_1),\, y_{r-1} \oplus k_2} \circ E_{k_0}^{T_{r-2},K}(x_{r-1} \oplus k_1)\\
&= \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \ldots \circ \mathsf{swap}_{E_{k_0}^{T_{r-1},K}(x_r \oplus k_1),\, y_r \oplus k_2}(y_{r-1} \oplus k_2)\\
&= F_r(y_{r-1} \oplus k_2).
\end{aligned}$$

The last equality holds if the following two properties hold.

Property 1**:** $y_{r-1} \notin \{y_r, \ldots, y_j\}$**.** Since we don't allow repeated queries, then $x_{r-1} \notin \{x_r, \ldots, x_j\}$. Since R is a permutation, $R(x_{r-1}) \notin \{R(x_r), \ldots, R(x_j)\}$ which gives the desired property.

Property 2**:** $y_{r-1} \oplus k_2 \notin \left\{E_{k_0}^{T_{r-1},K}(x_r \oplus k_1), \ldots, E_{k_0}^{T_{j-1},K}(x_j \oplus k_1)\right\}$**.** We prove step by step. First by contradiction, suppose $y_{r-1} \oplus k_2 = E_{k_0}^{T_{r-1},K}(x_r \oplus k_1)$. By the inductive assumption, we get

$$\begin{aligned}
y_{r-1} \oplus k_2 &= F_r^{-1}(y_r \oplus k_2)\\
F_r(y_{r-1} \oplus k_2) &= y_r \oplus k_2\\
E_{k_0}^{T_j,K}(x_{r-1} \oplus k_1) &= E_{k_0}^{T_j,K}(x_r \oplus k_1).
\end{aligned}$$

Since $E_{k_0}^{T_j,K}$ is a permutation, this contradicts the fact that $x_{r-1} \neq x_r$. Therefore we have $y_{r-1} \oplus k_2 \neq$

$E_{k_0}^{T_{r-1},K}(x_r \oplus k_1)$. Now, we can write

$$\begin{aligned}
&E_{k_0}^{T_j,K}(x_{r-1} \oplus k_1) \\
&= \mathsf{swap}_{E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),\, y_j \oplus k_2} \circ \ldots \circ \mathsf{swap}_{E_{k_0}^{T_r,K}(x_{r+1} \oplus k_1),\, y_{r+1} \oplus k_2}(y_{r-1} \oplus k_2) \\
&= F_{r+1}(y_{r-1} \oplus k_2).
\end{aligned}$$

By a similar prove-by-contradiction argument, using the inductive assumption on $r+1$, we obtain

$$y_{r-1} \oplus k_2 \neq E_{k_0}^{T_r,K}(x_{r+1} \oplus k_1).$$

Repeating this proof by contradiction step by step for $r+2, r+3, \ldots, j$, we eventually reach

$$y_{r-1} \oplus k_2 \neq E_{k_0}^{T_{j-1},K}(x_j \oplus k_1),$$

which establishes Property 2.

By these two properties, we can get

$$E_{k_0}^{T_j,K}(x_{r-1} \oplus k_1) = y_{r-1} \oplus k_2.$$

Then by strong induction, the proposition holds for all $i \in \{1, \ldots, j\}$. $\square$

Handling inverse Queries. We also need to extend our analysis to the setting where the distinguisher may issue classical queries in the inverse direction. Conceptually, the overall hybrid structure remains unchanged, i.e., the sequence of hybrids $(\mathbf{H}_0, \ldots, \mathbf{H}_{q_C})$ is identical in both the forward and inverse query settings. This is because each hybrid depends only on the transcript $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$, which records the pairs of classical queries made so far. The transcript itself is invariant under the query direction: whether the adversary queried $\mathsf{FX}_K(x_i) = y_i$ or $\mathsf{FX}_K^{-1}(y_i) = x_i$, the resulting set of pairs (x_i, y_i) is the same.

Consequently, the hybrids $\mathbf{H}_j$ remain identical in both settings. The only difference arises in the definition of the intermediate hybrids, where the classical queries are handled in different directions. In particular, the query direction affects how the reprogramming is applied within each step, and thus the intermediate transitions between $\mathbf{H}_j$ and $\mathbf{H}_{j+1}$ must be redefined accordingly.

We now show that these inverse queries can be handled in a fully symmetric manner. Specifically, for each pair of adjacent hybrids $\mathbf{H}_j$ and $\mathbf{H}_{j+1}$, we consider the $(j+1)^{\text{st}}$ classical query as an inverse query, which is answered by the inverse oracle. To formalize this, we first introduce the notation for inverse queries. Using the fact

$$E_{k_0}^{-1} \circ \mathsf{swap}_{a,\,b} = \mathsf{swap}_{E_{k_0}^{-1}(a),\, E_{k_0}^{-1}(b)} \circ E_{k_0}^{-1},$$

we obtain

$$\begin{aligned}(E_{k_0}^{T_1,k})^{-1} &= (\mathsf{swap}_{E_{k_0}(x_1 \oplus k_1),\, y_1 \oplus k_2} \circ E_{k_0})^{-1} \\ &= E_{k_0}^{-1} \circ \mathsf{swap}_{E_{k_0}(x_1 \oplus k_1),\, y_1 \oplus k_2} \\ &= \mathsf{swap}_{x_1 \oplus k_1,\, E_{k_0}^{-1}(y_1 \oplus k_2)} \circ E_{k_0}^{-1}.\end{aligned}$$

Extending to j swaps, we get

$$\begin{aligned}(E_{k_0}^{T_j,k})^{-1}(y) &= E_{k_0}^{-1} \circ \mathsf{swap}_{E_{k_0}(x_1\oplus k_1),\, y_1\oplus k_2} \circ \cdots \circ \mathsf{swap}_{E_{k_0}^{T_{j-1},k}(x_j\oplus k_1),\, y_j\oplus k_2}(y) \\ &= (E_{k_0}^{T_{j-1},k})^{-1} \circ \mathsf{swap}_{E_{k_0}^{T_{j-1},k}(x_j\oplus k_1),\, y_j\oplus k_2}(y) \\ &= \mathsf{swap}_{x_j\oplus k_1,\, (E_{k_0}^{T_{j-1},k})^{-1}(y_j\oplus k_2)} \circ (E_{k_0}^{T_{j-1},k})^{-1}(y) \\ &= \prod_{i=j}^{1} \mathsf{swap}_{x_i\oplus k_1,\, (E_{k_0}^{T_{i-1},k})^{-1}(y_i\oplus k_2)} \circ E_{k_0}^{-1}(y).\end{aligned}$$

Moreover, define

$$(E_{k^*}^{(1)})^{-1}(y) = \begin{cases} E_{k^*}^{-1}(x) & \text{if } k^* \neq k_0 \\ \left(E_{k_0}^{-1} \circ \mathsf{swap}_{s_0,\, s_1}\right)(y) & \text{if } k^* = k_0. \end{cases}$$

Analogous to the forward case, we first define the intermediate hybrids $\mathbf{H}_j^{1,\text{inv}}$ and $\mathbf{H}_j^{2,\text{inv}}$ ($\mathbf{H}_3^j$ is the same in both cases). We then show that the distinguishing advantage between $\mathbf{H}_j$ and $\mathbf{H}_{j+1}$ remains the same in the inverse case. Specifically, for all $0 \leq j < q_C$, we establish the following bounds:

Lemma 3.8.1: $\quad |\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^{1,\text{inv}}) = 1\right]| \leq 4\sqrt{\dfrac{q_Q}{2^m(2^n - j)}}$

Lemma 3.8.2: $\quad \mathbf{H}_j^{1,\text{inv}} = \mathbf{H}_j^{2,\text{inv}}$

Lemma 3.8.3: $\quad \mathbf{H}_j^{2,\text{inv}} = \mathbf{H}_j^3$

Lemma 3.5.6: $\quad |\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{Q,j+1}\sqrt{\dfrac{2(j+1)}{2^{m+n}}},$

where $q_{Q,j+1}$ is the expected number of queries $\mathcal{A}$ makes to P in the $(j+1)^{\text{st}}$ stage, i.e., the stage between the $(j+1)^{\text{st}}$ and $(j+2)^{\text{nd}}$ classical queries. By Equation 3.25, we obtain the same bound as in the forward case. Hence, inverse queries can be handled in a fully symmetric manner, yielding the same distinguishing advantage.

Experiment $\mathbf{H}_j^{1,\mathbf{inv}}$.

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using E, until $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query y_{j+1}, which we assume for concreteness to be in the inverse direction. Let T_j be the list of the classical queries so far.

2. Define set $S = \{0,1\}^n \backslash \{y_1 \oplus k_2, \cdots, y_j \oplus k_2\}$. Choose uniform $s \in S$, and define $(E^{(1)})^{-1}$ as

$$(E_{k^*}^{(1)})^{-1}(y) = \begin{cases} E_{k^*}^{-1}(y) & \text{if } k^* \neq k_0 \\ \left(E_{k_0}^{-1} \circ \mathsf{swap}_{y_{j+1}\oplus k_2,\, s}\right)(y) & \text{if } k^* = k_0. \end{cases}$$

Then

$$(E_{k^*}^{(1)})(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k_0 \\ \left(E_{k_0} \circ \mathsf{swap}_{E_{k_0}^{-1}(y_{j+1}\oplus k_2),\, E_{k_0}^{-1}(s)}\right)(x) & \text{if } k^* = k_0. \end{cases}$$

Continue running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{FX}_K[(E^{(1)})^{T_j,K}]$, and its quantum queries using $(E^{(1)})^{T_j,K}$.

Experiment $\mathbf{H}_j^{2,\mathbf{inv}}$.

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using E, until $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query y_{j+1}. Let T_j be the list of classical queries so far.

2. Define $E^{(1)}$ as in $\mathbf{H}_j^{1,\text{inv}}$ and answer the $(j+1)^{\text{st}}$ classical query using $\mathsf{FX}_K[(E^{(1)})^{T_j,K}]$. Denote the response as x_{j+1}, i.e.,

$$x_{j+1} = (\mathsf{FX}_K[(E^{(1)})^j])^{-1}(y_{j+1}).$$

Then the challenger constructs E^{j+1} adaptively as in Equation 3.22. Continue running $\mathcal{A}$, answering its remaining classical queries using $\mathsf{FX}_K[E^{T_{j+1},K}]$, and its quantum queries using $E^{T_{j+1},K}$.

Next, we prove Lemma 3.8.1, Lemma 3.8.2 and Lemma 3.8.3. The proofs are proceeded in an entirely analogous manner to those of Lemma 3.5.3, Lemma 3.5.4 and Lemma 3.5.5.

Lemma 3.8.1. *For $j = 0, \ldots, q_C$,*

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^{1,inv}) = 1\right]\right| \leq 4\sqrt{\frac{q_Q}{2^m(2^n - j)}}.$$

Proof. In this lemma, we bound the distinguishability of $\mathbf{H}_j$ and $\mathbf{H}_j^{1,\text{inv}}$.

$$\begin{aligned} \mathbf{H}_j:&\quad |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[\ E^j\ ], |E^j\rangle \quad\quad , \mathsf{FX}_K\left[E^j\right], |E^j\rangle, \cdots \\ \mathbf{H}_j^{1,\text{inv}}:&\quad |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle\ , \mathsf{FX}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots. \end{aligned}$$

Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j$ and $\mathbf{H}_j^{1,\text{inv}}$. We construct from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for the resampling experiment of Lemma 3.4.1. $\mathcal{D}$ does:

Phase 1: $\mathcal{D}$ is given quantum access to an ideal cipher E. It samples a uniform $R \leftarrow \mathcal{P}_n$ and then runs $\mathcal{A}$, answering its quantum queries with E and its classical queries with R (in the appropriate directions), until $\mathcal{A}$ submits its $(j+1)^{\text{st}}$ classical query y_{j+1}. At that point, $\mathcal{D}$ has a list $T_j = \{(x_1, y_1), \cdots, (x_j, y_j)\}$ of the queries/answers $\mathcal{A}$ has made to its classical oracle thus far. Next, $\mathcal{D}$ constructs a distribution D on $\{0,1\}^{m+2n}$ and it's sampling algorithm Π. To sample a tuple $(a_0, z_0, z_1) \leftarrow D$, Π does the following:

1 : Sample uniform $a_0 \in \{0,1\}^m$ and $z_0 \in \{0,1\}^n$.

2 : Construct $S = \{0,1\}^n \backslash \{z_0 \oplus y_1 \oplus y_{j+1}, \cdots, z_0 \oplus y_j \oplus y_{j+1}\} = \{0,1\}^n \backslash S^{\perp}$.

3 : Sample uniform $z_1 \in S$, and output (a_0, z_0, z_1).

Phase 2: $\mathcal{D}$ is given $(k_0, s_0, s_1) \leftarrow D$ and quantum oracle access to a cipher $E^{(b)}$. Then $\mathcal{D}$ sets $k_2 = y_{j+1} \oplus s_0$, uniform $k_1 \in \{0,1\}^n$ and $k = (k_0, k_1, k_2)$. It then continues running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{FX}_K[(E^{(b)})^{T_j,k}]$, and its remaining quantum queries using $(E^{(b)})^{T_j,k}$. $\mathcal{D}$ outputs whatever $\mathcal{A}$ does.

Note that $s_1 \notin S^{\perp}$ where $S^{\perp} = \{y_1 \oplus k_2, \cdots y_j \oplus k_2\}$, by algorithm Π. In phase 1, distinguisher $\mathcal{D}$ perfectly simulates experiments $\mathbf{H}_j$ and $\mathbf{H}_j^{1,\text{inv}}$ for $\mathcal{A}$ until the point where $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query. If $b = 0$, $\mathcal{D}$ gets access to $E^{(0)} = E$ in phase 2. Since $\mathcal{D}$ answers all quantum queries using $(E^{(0)})^{T_j,k}$ and all classical queries using $\mathsf{FX}_K[(E^{(0)})^{T_j,k}]$, we see that $\mathcal{D}$ perfectly simulates $\mathbf{H}_j$ for $\mathcal{A}$ in that case. If, on the other hand, $b = 1$ in phase 2, then $\mathcal{D}$ gets access to $E^{(1)}$, where

$$(E_{k^*}^{(1)})^{-1}(y) = \begin{cases} E_{k^*}^{-1}(x) & \text{if } k^* \neq k_0 \\ \left(E_{k_0}^{-1} \circ \mathsf{swap}_{s_0, s_1}\right)(y) & \text{if } k^* = k_0. \end{cases}$$

Since $k_2 \coloneqq s_0 \oplus y_{j+1}$, it holds that

$$(E_{k^*}^{(1)})^{-1}(y) = \begin{cases} E_{k^*}^{-1}(x) & \text{if } k^* \neq k_0 \\ \left(E_{k_0}^{-1} \circ \mathsf{swap}_{k_2 \oplus y_{j+1}, s_1}\right)(y) & \text{if } k^* = k_0. \end{cases}$$

Moreover, the fact that s_0 (and hence $s_0 \oplus y_{j+1}$), k_0 and k_1 are uniform implies that $\mathcal{D}$ perfectly simulates $\mathbf{H}_j^{1,\text{inv}}$ for $\mathcal{A}$. Applying Lemma 3.4.1 thus gives

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^{1,\text{inv}}) = 1\right]\right| \leq 4\sqrt{q_Q \cdot \varepsilon \cdot 2^n},$$

where ϵ is the same as in Lemma 3.5.3. Therefore, we have

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr\left[\mathcal{A}(\mathbf{H}_j^{1,\text{inv}}) = 1\right]\right| \leq 4\sqrt{\frac{q_Q}{2^m(2^n - j)}}\,.$$

□

Lemma 3.8.2. *For $j = 0, \ldots, q_C$,*

$$\mathbf{H}_j^{1,inv} = \mathbf{H}_j^{2,inv}\;.$$

Proof. We bound the distinguishability of $\mathbf{H}_j^{1,\text{inv}}$ and $\mathbf{H}_j^{2,\text{inv}}$.

$$\mathbf{H}_j^{1,\text{inv}}: \;\; |E\rangle\,, R, |E\rangle\,, \cdots, R, |E\rangle\,, \mathsf{FX}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle\;\;, \mathsf{FX}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots$$

$$\mathbf{H}_j^{2,\text{inv}}: \;\; |E\rangle\,, R, |E\rangle\,, \cdots, R, |E\rangle\,, \mathsf{FX}_K[(E^{(1)})^j], |E^{j+1}\rangle\;\;, \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle\,, \cdots.$$

To prove $\mathbf{H}_j^{1,\text{inv}}$ and $\mathbf{H}_j^{2,\text{inv}}$ are indistinguishable, it suffices to show that the quantum oracles $(E^{(1)})^j$ and E^{j+1} are identical.

In both $\mathbf{H}_j^{1,\text{inv}}$ and $\mathbf{H}_j^{2,\text{inv}}$, the response to the $(j+1)^{\text{st}}$ classical query is:

$$
\begin{aligned}
x_{j+1} &\stackrel{\text{def}}{=} (\mathsf{FX}_K[(E^{(1)})^{T_j,k}])^{-1}(y_{j+1}) \\
&= ((E_{k_0}^{(1)})^{T_j,k})^{-1}(y_{j+1} \oplus k_2) \oplus k_1 \\
&= \prod_{i=j}^{1} \mathsf{swap}_{x_i \oplus k_1,\, ((E_{k_0}^{(1)})^{T_{i-1},k})^{-1}(y_i \oplus k_2)} \circ (E_{k_0}^{(1)})^{-1}(y_{j+1} \oplus k_2) \oplus k_1 \\
&= \prod_{i=j}^{1} \mathsf{swap}_{x_i \oplus k_1,\, (E_{k_0}^{T_{i-1},k})^{-1}(y_i \oplus k_2)} \circ (E_{k_0}^{-1} \circ \mathsf{swap}_{y_{j+1} \oplus k_2,\, s}(y_{j+1} \oplus k_2)) \oplus k_1 \\
&= \prod_{i=j}^{1} \mathsf{swap}_{x_i \oplus k_1,\, (E_{k_0}^{T_{i-1},k})^{-1}(y_i \oplus k_2)} \circ E_{k_0}^{-1}(s) \oplus k_1 \\
&= (E_{k_0}^{T_j,k})^{-1}(s) \oplus k_1,
\end{aligned}
$$

where the fourth equality comes from Proposition 2. By rearranging $s = E_{k_0}^{T_j,k}(x_{j+1} \oplus k_1)$, it follows that for any $y \in \{0,1\}^n$,

$$
\begin{aligned}
((E^{(1)})^j)^{-1}(y) &= \prod_{i=j}^{1} \mathsf{swap}_{x_i \oplus k_1,\, ((E_{k_0}^{(1)})^{j-1})^{-1}(y_i \oplus k_2)} \circ E_{k_0}^{-1} \circ \mathsf{swap}_{y_{j+1} \oplus k_2,\, s}(y) \\
&= \prod_{i=j}^{1} \mathsf{swap}_{x_i \oplus k_1,\, (E_{k_0}^{j-1})^{-1}(y_i \oplus k_2)} \circ E_{k_0}^{-1} \circ \mathsf{swap}_{y_{j+1} \oplus k_2,\, s}(y) \\
&= (E^j)^{-1} \circ \mathsf{swap}_{y_{j+1} \oplus k_2,\, E_{k_0}^j(x_{j+1} \oplus k_1)}(y) \\
&= \mathsf{swap}_{x_{j+1} \oplus k_1,\, (E^j)^{-1}(y_{j+1} \oplus k_2)} \circ (E^j)^{-1}(y) \\
&= (E^{j+1})^{-1}(y)
\end{aligned}
$$

where the second equality comes from Proposition 2. This concludes the proof that $(E^{(1)})^j \equiv E^{j+1}$.

Therefore, hybrids $\mathbf{H}_j^{1,\text{inv}}$ and $\mathbf{H}_j^{2,\text{inv}}$ are perfectly indistinguishable:

$$\mathbf{H}_j^{1,\text{inv}} = \mathbf{H}_j^{2,\text{inv}} \ .$$

□

Lemma 3.8.3. *For $j = 0, \dots, q_C$, $\mathbf{H}_j^{2,inv} = \mathbf{H}_j^3$.*

Proof.

$$\begin{aligned}
\mathbf{H}_j^{2,\text{inv}}: \ & |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \mathsf{FX}_K[(E^{(1)})^j], |E^{j+1}\rangle \quad, \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}_j^3: \ & |E\rangle, R, |E\rangle, \cdots, R, |E\rangle, \qquad R \qquad, |E^{j+1}\rangle \quad, \mathsf{FX}_K[E^{j+1}], |E^{j+1}\rangle, \cdots .
\end{aligned}$$

We observe that $\mathbf{H}_j^{2,\text{inv}}$ and $\mathbf{H}_j^3$ differ only in the response to the $(j+1)^{\text{st}}$ classical query. In $\mathbf{H}_j^{2,\text{inv}}$, this query is answered using

$$x_{j+1} \ = \ (\mathsf{FX}_K\big[(E^{(1)})^{T_j,K}\big])^{-1}(y_{j+1}) \ = \ (E_{k_0}^{T_j,K})^{-1}(s) \oplus k_1,$$

as shown in Lemma 3.5.4. In contrast, in $\mathbf{H}_j^3$ the same query is answered with

$$x_{j+1} \ = \ R^{-1}(y_{j+1}).$$

Using Proposition 1 and following the same strategy as in Lemma 3.5.5, we conclude that the two hybrids induce identical distributions on x_{j+1}. Hence, $\mathbf{H}_j^{2,\text{inv}} = \mathbf{H}_j^3$. □

3.8.3 Detailed attacks on LRW and XEX2

We provide a detailed description of the attacks on standardized LRW and XEX2, as outlined in Section 3.6.3.

Classical Distinguishing Attacks.

Birthday Collision Attack. As mentioned in Section 3.2.4, $q = O(2^{n/2})$ queries are required to distinguish LRW (Definition 3.6.6) from an ideal tweakable block cipher. The attack is outlined below, given access to a classical oracle:

1. Randomly choose $\tau \in \{0,1\}^n$.
2. Query $\widetilde{O}$ with (m, τ) for $2^{n/2}$ distinct values of m, obtaining pairs (m, c), where c is the response, and store them.
3. Randomly choose $\tau' \in \{0,1\}^n$.
4. Query $\widetilde{O}$ with (m', τ') for $2^{n/2}$ distinct values of m', obtaining pairs (m', c'), where c' is the response, and store them.
5. Find a pair (c, c') such that $m \oplus c = m' \oplus c'$.
6. Based on the construction of h, use the identified pair (c, c') (or multiple such pairs) to distinguish between an ideal tweakable cipher and LRW.

Example. Consider the LRW construction as established in [P1606], where $h_{k'}(\tau) = k' \times_* \tau$. For this construction, the attack proceeds as follows:

1. Perform steps 1–5 twice to obtain two pairs (c_1, c_1') and (c_2, c_2') corresponding to two pairs of tweaks (τ_1, τ_1') and (τ_2, τ_2').

2. Compute $k_1' = (c_1 \oplus c_1') \times_* (\tau_1 \oplus \tau_1')^{-1}$ and similarly $k_2' = (c_2 \oplus c_2') \times_* (\tau_2 \oplus \tau_2')^{-1}$.

3. If $k_1' = k_2'$, output LRW. Otherwise output ideal tweakable cipher.

Analysis of the Algorithm. If $\widetilde{O}$ is LRW, i.e. $\widetilde{O} = \mathsf{LRW}_{k,k'}$, we define $H_{k'}(m, \tau) := m \oplus h_{k'}(\tau)$. Since h is a universal hash function, for all m, τ with a uniformly selected k', $H_{k'}(m, \tau)$ is uniformly distributed. Now, recall that a birthday attack can be applied to $H_{k'}$. Specifically, with $2^{n/2}$ randomly selected inputs, there exists (with overwhelming probability) a pair (m, τ) and (m', τ') such that

$$H_{k'}(m, \tau) = H_{k'}(m', \tau') \quad \Rightarrow \quad m \oplus h_{k'}(\tau) = m' \oplus h_{k'}(\tau').$$

This implies:

$$c \oplus c' = m \oplus m' = h_{k'}(\tau) \oplus h_{k'}(\tau').$$

Substituting $h_{k'}(\tau) = k' \times_* \tau$, we get:

$$c \oplus c' = k' \times_* \tau \oplus k' \times_* \tau'.$$

Using this, we compute the hash key:

$$k' = (c \oplus c') \times_* \tau \oplus \tau')^{-1}.$$

The key k' will always be the same for all such collisions.

If $\widetilde{O}$ is an ideal tweakable cipher, then similarly, by the birthday attack, among $2^{n/2}$ randomly selected inputs, there will exist a collision where:

$$c = \widetilde{P}(m, \tau) = \widetilde{P}(m', \tau') = c'$$

i.e. a collision on the independent random permutations $f_0(\tau) = f_1(\tau')$. However, these collisions do not reveal any structure of k'; instead, the two pairs of ciphertexts and tweaks provide a random guess of k'. Thus, with overwhelming probability, $k'_1 \neq k'_2$.

In conclusion, this classical attack requires $O(2^{n/2})$ queries. This attack bound has been mentioned in several related works [LRW02].

Remark. If $\widetilde{O}$ is XEX2, i.e. $h_{k'}(\tau) = h_{k'}(i, j) = \alpha^j \otimes E_{k'}(2^i)$, we also follow step 1-5 twice for two pairs of tweaks (i, j_1, i, j'_1) and (i, j_2, i, j'_2), obtaining the corresponding ciphertext pairs (c_1, c'_1) and (c_2, c'_2). Using these, we compute $(c_1 \oplus c'_1) \times_* (\alpha^{j_1} \oplus \alpha^{j'_1})^{-1}$ and $(c_2 \oplus c'_2) \times_* (\alpha^{j_2} \oplus \alpha^{j'_2})^{-1}$ and check whether they match. It they do, this value is the candidate for $E_{k'}(2^i)$. Similar arguements work for XEX2.

Even-Mansour Attack. Here, we use the fact that LRW with one tweak is an Even-Mansour to construct the distinguishing attack:

1. Randomly choose two tweaks $t, \tau' \in \{0, 1\}^n$, and define two oracles

$$P(\cdot) := \widetilde{O}(\cdot, \tau), \quad R(\cdot) := \widetilde{O}(\cdot, \tau').$$

2. Define $f(x) = P(x) \oplus R(x)$. Query $f(\cdot)$ $2^{n/2}$ times and find a collision, i.e., $f(x) = f(x')$.

3. Compute $\widetilde{k} = x \oplus x'$.

4. Randomly select $x^* \in \{0,1\}^n$. Compute $R(x^* \oplus \widetilde{k}) \oplus \widetilde{k}$ and $P(x^*)$.

5. If the two results are equal, output LRW; otherwise, output $\mathsf{ideal\ tweakable\ cipher}$.

<u>Analysis of the Algorithm.</u> If $\widetilde{O}$ is LRW, i.e. $\widetilde{O} = \mathsf{LRW}_{k,k'}$, then

$$
\begin{aligned}
P(x) &= \widetilde{O}(x, \tau) = E_k(x \oplus h_{k'}(\tau)) \oplus h_{k'}(\tau) \\
R(x) &= \widetilde{O}(x, \widetilde{\tau}) = E_k(x \oplus h_{k'}(\widetilde{\tau})) \oplus h_{k'}(\widetilde{\tau}).
\end{aligned}
$$

We can rewrite $P(x)$ as follows:

$$
\begin{aligned}
P(x) &= E_k(x \oplus \underline{h_{k'}(\tau) \oplus h_{k'}(\widetilde{\tau})} \oplus h_{k'}(\widetilde{\tau})) \oplus \underline{h_{k'}(\tau) \oplus h_{k'}(\widetilde{\tau})} \oplus h_{k'}(\widetilde{\tau}) \\
&= E_k(x \oplus \widetilde{k} \oplus h_{k'}(\widetilde{\tau})) \oplus \widetilde{k} \oplus h_{k'}(\widetilde{\tau}) \\
&= R(x \oplus \widetilde{k}) \oplus \widetilde{k},
\end{aligned}
$$

This shows that P is an Even-Mansour of R. Next, consider

$$\begin{aligned}
f(x) &= P(x) \oplus R(x) \\
&= R(x \oplus \widetilde{k}) \oplus \widetilde{k} \oplus R(x) \\
&= R(x \oplus \widetilde{k}) \oplus \widetilde{k} \oplus R(x \oplus \widetilde{k} \oplus \widetilde{k}) \\
&= R(x \oplus \widetilde{k}) \oplus P(x \oplus \widetilde{k}) \\
&= f(x \oplus \widetilde{k}).
\end{aligned}$$

This periodicity allows us to use $2^{n/2}$ queries to find a collision on $f(x) = f(x')$. The Even-Mansour key is then computed as $\widetilde{k} = x \oplus x'$. For any $x^* \in \{0,1\}^n$, the Even-Mansour relation between P and R would hold.

If $\widetilde{O}$ is an ideal tweakable cipher, then $P(x)$ and $R(x)$ are independent random permutations of input x. In this case, $f(x) = P(x) \oplus R(x)$ is the XOR of random permutations. While collisions can still be found using $2^{n/2}$ queries, the computed $\widetilde{k}$ is meaningless. For a random x^*, $P(x^*)$ and $R(x^* \oplus \widetilde{k}) \oplus \widetilde{k}$ are independently and uniformly distributed and so will not be equal for overwhelming probability.

We note that this attack works similarly for XEX2.

Classical Complete Key-Search Attack.

1. Randomly select $\tau \in \{0,1\}^n$.

2. Use the classical distinguishing attack described above to recover k' (assuming an appropriate hash function construction).

3. Construct an oracle $E_k(\cdot)$ such that, for an input x, it replies

$$E_k(x) = \mathsf{LRW}_{k,k'}(x \oplus h_{k'}(\tau), \tau) \oplus h_{k'}(\tau).$$

4. Perform an exhaustive search to get key k.

Analysis of the Algorithm. If the tweakable block cipher under investigation is standardized LRW, then in step 2, k' can be computed directly. We then perform an exhaustive search to recover k. Specifically, we first query LRW using a constant number of inputs, such as $(x \oplus h_{k'}(\tau), \tau)$. Then, for each candidate k, we query $E_k(\cdot)$ with the same x values and verify whether the equation from step 3 holds. This process requires a total of $O(2^{n/2})$ online queries and $O(2^m)$ offline queries for complete key recovery.

If the tweakable block cipher under investigation is $\mathsf{XEX2}$, then in step 2, using the ciphertext pair (c, c') with corresponding tweaks $(i, j), (i, j')$, we can compute $E_{k'}(2^i) = (c \oplus c') \times_* (\alpha^j \oplus \alpha^{j'})^{-1}$. Next, we query $E(\cdot, 2^i)$ 2^m times with different candidate values for k'. With high probability $(1 - \frac{1}{2^n})$, we identify the correct tweak key k'^* by verifying that $E(k'^*, 2^i) = (c \oplus c') \times_* (\alpha^j \oplus \alpha^{j'})^{-1}$. Once k' is obtained, we follow a similar process as for LRW to recover k.

Thus, the total number of queries required for complete key recovery remains $O(2^{n/2})$ online queries (plus $O(2^m)$ offline queries).

Quantum Complete Key Recovery Attack (shorter cipher key). The best-known attack for quantum complete key search is the Offline-Simon attack [BHNP+19]. Let u be an integer such that

$0 \le u \le n$. Define the function $F : \{0,1\}^{m+(n-u)} \times \{0,1\}^u \to \{0,1\}^n$ as follows:

$$F(i \| j, x) = E_i(x \| j) \ (i \in \{0,1\}^m, j \in \{0,1\}^{n-u}).$$

Additionally, define the function $g : \{0,1\}^u \to \{0,1\}^n$ as

$$g(x) = \mathsf{LRW}_{k,k'}(x \| 0^{n-u}, \tau).$$

Note that $F(k \| h_{k'}^{(2)}(\tau), x) \oplus g(x)$ has the period $h_{k'}^{(1)}(\tau)$ since

$$F(k \| h_{k'}^{(2)}(\tau), x) \oplus g(x) = E_k(x \| h_{k'}^{(2)}(\tau)) \oplus E_k\left((x \oplus h_{k'}^{(1)}(\tau)) \| h_{k'}^{(2)}(\tau)\right) \oplus h_{k'(\tau)}.$$

The procedure for recovering k and k' is as follows:

1. Randomly choose $\tau \in \{0,1\}^n$.
2. Run $\mathsf{Alg\text{-}ExpQ1}$ on the defined F and g to recover k and $h_{k'}^{(2)}(\tau)$.
3. Use $\mathsf{SimQ1}$ on $f_{k \| h_{k'}^{(2)}(\tau)}$ to recover $h_{k'}^{(1)}(\tau)$.
4. (Verification) Check if $E(k,k')(0^n, \tau) \oplus E_k(h_{k'}(\tau)) = h_{k'}(\tau)$. If the check fails, repeat steps 2–4.
5. Based on construction of hash function h, compute k'.

Analysis of the Algorithm. Similar to the classical case, if the tweakable block cipher under investigation is standardized LRW, k' can be easily computed from $h_{k'}(\tau)$ using $k' = h_{k'}(\tau) \times_* t^{-1}$.

So $\widetilde{O}(2^{(n+m)/3})$ queries (with $m \leq 2n$) are required for a complete key search, same as the query complexity of the attack on the FX construction [BHNP+19].

If the tweakable block cipher under investigation is $\mathsf{XEX2}$, then $h_{k'}(\tau) := h_{k'}(i,j) = E_{k'}(2^i) \times \alpha^j$ is recovered from step 2 and 3. As in the classical case, we can query $E(\cdot, 2^i)$ 2^m times with different k' to identify the true key that satisfies the matching condition. Thus, $D = O(2^u)$ classical queries are made to $\mathsf{LRW}_{k,k'}(\cdot,\cdot)$ and $T = O(n^3 2^{(m+n-u)/2} + 2^{m/2}) = O(n^3 2^{(m+n-u)/2})$ offline quantum queries are performed. When $D \leq 2n$, the balance is achieved at at $T = D = \widetilde{O}(2^{(n+m)/3})$.

Quantum Complete Key Recovery Attack (longer cipher key). Another possible quantum attack involves combining Grover's algorithm with the Kuwakado-Morii attack [KM12] by treating the Kuwakado-Morii method as a predicate and using Grover's search to find the cipher key k. Detailedly,

1. Fix a key $k \in \{0,1\}^m$.
2. Select $\tau \in \{0,1\}^n$ at random.
3. Choose x_i ($i = 1, \ldots, s$) from $\{0,1\}^n$ at random, ensuring $x_i \neq x_j$ for $i \neq j$ and $\overline{x_i} \notin \{x_i \mid i = 1, 2, \ldots, s\}$ for any i.
4. For $i = 1, 2, \ldots, s$, compute

$$d_i = \mathsf{LRW}_{k,k'}(\tau, x_i) \oplus \mathsf{LRW}_{k,k'}(\tau, \overline{x_i})$$

by querying $\mathsf{LRW}_{k,k'}(\cdot,\cdot)$ classically. Then define a set

$$D = \{d_i | i = 1, 2, \ldots, s\}.$$

Assume all elements in D are distinct for simplicity.

5. Create a table S containing pairs (d_i, x_i), sorted by d_i.

6. Find z such that $L_D(z) = 1$ using the Grover algorithm with the unitary operator V_{L_D}, where $L_D : \{0,1\}^n \to \{0,1\}$ is defined as

$$L_{\mathcal{D}}(x) = \begin{cases} 0 & \text{if } E(k,x) \oplus E(k,\bar{x}) \notin D \\ 1 & \text{if } E(k,x) \oplus E(k,\bar{x}) \in D, \end{cases},$$

and

$$V_{L_D}|x\rangle = (-1)^{L_D(x)}|x\rangle.$$

7. Compute

$$d = E(k,z) \oplus E(k,\bar{z}),$$

then find an x from table S such that

$$d = \mathsf{LRW}_{k,k'}(\tau, x) \oplus \mathsf{LRW}_{k,k'}(\tau, \bar{x}).$$

8. Compute $h_{k'}(\tau) = x \oplus z$.

9. Select $x' \in \{0,1\}^n$ at random and obtain $\mathsf{LRW}_{k,k'}(\tau, x')$ by querying $\mathsf{LRW}_{k,k'}(\cdot,\cdot)$ classically.

10. If $E(k,x') \oplus h_{k'}(\tau) \neq \mathsf{LRW}_{k,k'}(\tau, x')$, then set $h_{k'} = \bar{x} \oplus z$.

11. Repeat step 2–10 three times to obtain three pairs: $(\tau_1, h_{k'}(\tau_1))$, $(\tau_2, h_{k'}(\tau_2))$, and $(\tau_3, h_{k'}(\tau_3))$. If $h_{k'}(\tau_1), h_{k'}(\tau_2), h_{k'}(\tau_3)$ are not uniformly distributed, output 1; otherwise, output 0.

12. Treat steps 2–11 as a predicate f, where $f(k) \in \{0,1\}$, and find k such that $f(k) = 1$ using the Grover algorithm with the unitary operator U_f.

13. (Verification) If a valid k is found, compute k' based on the construction of the hash function h and output 1 or 0 accordingly.

Analysis of the Algorithm. This combined attack employs BHT to recover the tweak term $h_{k'}(\tau)$ and Grover's algorithm to find the cipher key k. Therefore, it requires $q_Q = O(2^{m/2} \cdot 2^{n/3}) = O(2^{m/2+n/3})$ quantum queries and $q_C = O(2^{n/3})$ classical queries.

3.8.4 Proof of the post-quantum security of LRW

Theorem 3.8.4 (Restatement)**.** *Let* LRW *be as in Definition 3.6.6 and let $\mathcal{A}$ be an adversary making q_C classical queries to its first oracle and $q_Q \geq 1$ quantum queries to its second oracle. Assuming h is XOR-universal and uniform, it holds that in the ideal cipher model,*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^m; k' \leftarrow \{0,1\}^\kappa \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \mathsf{LRW}_{k,k'}[E_\cdot], |\pm E\rangle} = 1\right] - \Pr_{\substack{\widetilde{\Pi} \leftarrow \mathcal{E}(\mathcal{T},n); \\ E \leftarrow \mathcal{E}(m,n)}} \left[\mathcal{A}^{\pm \widetilde{\Pi}, |\pm E\rangle} = 1\right] \right| \leq \frac{6q_C^2}{2^n} + \frac{4}{2^{(m+n)/2}} \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right).$$

Proof. We adapt a similar hybrid technique to that used in proving the post-quantum security of FX (see Theorem 3.5.2) for the case of LRW. Similarly, we assume that all classical queries are made in the forward direction. The inverse queries can be handled in a similar manner as FX. We begin by setting down a way of modifying a given cipher based on a choice of keys and a list of classical queries to an LRW oracle. Let $E \in \mathcal{E}(m,n)$, $K = (k,k') \in \{0,1\}^{m+\kappa}$. Fix a list $\{(\tau_1,x_1,y_1),(\tau_2,x_2,y_2),\ldots,(\tau_{q_C},x_{q_C},y_{q_C})\}$. Each pair (τ_i,x_i,y_i) represents an input–output pair of a classical query. Repeated queries are not allowed. For any $j \leq q_C$, let T_j be the list containing the first j queries.

The "modified cipher" after j-many queries is denoted $E^{T_j,K}$, and is given by

$$E_{k^*}^{T_j,K}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k \\ \\ E_k^{T_j,K}(x) & \text{if } k^* = k. \end{cases} \tag{3.34}$$

where $E_k^{T_j,K}$ is defined as follows. First, define $E^{T_0,k} = E$, and for all $j \in [1, q_C]$,

$$E_k^{T_j,K}(x) = \mathsf{swap}_{E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ E_k^{T_{j-1},K}(x)\,. \tag{3.35}$$

Roughly speaking, $E^{T_j,K}$ is the minimal modification of E that is consistent with the forward ($\rightarrow$) and inverse ($\leftarrow$) queries from the transcript T_j. For compactness, we occasionally write E^j in place of $E^{T_j,K}$ when T_j and K are clear to the context. We also set $E^0 = E$.

We now define the bad event $\mathsf{Bad}_j = \mathsf{Bad}_{j,1} \vee \mathsf{Bad}_{j,2}$, where:

- $\mathsf{Bad}_{j,1}$: A collision occurs in the set $\{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$.
- $\mathsf{Bad}_{j,2}$: A collision occurs in the set $\{y_1 \oplus h_{k'}(\tau_1), \ldots, y_j \oplus h_{k'}(\tau_j)\}$.

The probability of the bad event occurring depends on the distributions of E_k, k and T_j. Given the XOR-universality of h and that k' is sampled uniformly at random (after T_j is fixed), the bad event occurs with probability $\Pr[\mathsf{Bad}_j] \leq \frac{j^2}{2^n}$. Conditioned on this event, we derive statements analogous to those in Proposition 1 in the proof of FX.

Proposition 5. *For any $E \in \mathcal{E}(m,n)$, $K = (k,k') \in \{0,1\}^{m+\kappa}$, $j \in \{1,\ldots,q_C\}$, transcript $T_j =$*

$\{(x_1, y_1), \ldots, (x_j, y_j)\}$ *without repetition, and any* $i \in \{1, \ldots, j\}$*, if* $\neg\textsf{\textit{Bad}}_j$*,*

$$E_k^{T_j,K}(x_i \oplus h_{k'}(\tau_i)) \oplus h_{k'}(\tau_i) = y_i.$$

Proof. We prove by strong induction on i. We start by the base case $i = j$.

$$E_k^{T_j,K}(x_j \oplus h_{k'}(\tau_j)) = \mathsf{swap}_{E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)) = y_j \oplus h_{k'}(\tau_j).$$

Assume for all $i \in \{r, \ldots, j\}$ that $E_k^{T_j,K}(x_i \oplus h_{k'}(\tau_i)) \oplus h_{k'}(\tau_i) = y_i$. Expanding it, we obtain

$$\begin{aligned}
&E_k^{T_j,K}(x_i \oplus h_{k'}(\tau_i)) \\
&= \mathsf{swap}_{E_k^{T_j,K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \ldots \circ \mathsf{swap}_{E_k^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)),\, y_i \oplus h_{k'}(\tau_i)} \circ E_k^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)) \\
&= F_i \circ E_k^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)) = y_i \oplus h_{k'}(\tau_i),
\end{aligned}$$

where we define

$$F_i = \mathsf{swap}_{E_k^{T_j,K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \ldots \circ \mathsf{swap}_{E_k^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)),\, y_i \oplus h_{k'}(\tau_i)}.$$

By rearranging, we get for all $i \in \{r, \ldots, j\}$,

$$E_k^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)) = F_i^{-1}(y_i \oplus h_{k'}(\tau_i)).$$

We now work on case $r-1$,

$$
\begin{aligned}
&E_k^{T_j,K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})) \\
&= \mathsf{swap}_{E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \ldots \circ \mathsf{swap}_{E_k^{T_{r-2},K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})),\, y_{r-1} \oplus h_{k'}(\tau_{r-1})} \\
&\qquad \circ E_k^{T_{r-2},K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})) \\
&= \mathsf{swap}_{E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \ldots \circ \mathsf{swap}_{E_k^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r)),\, y_r \oplus h_{k'}(\tau_r)}(y_{r-1} \oplus h_{k'}(\tau_{r-1})) \\
&= F_r(y_{r-1} \oplus h_{k'}(\tau_{r-1})).
\end{aligned}
$$

We now argue the following two properties:

Property 1**:** $y_{r-1} \oplus h_{k'}(\tau_{r-1}) \notin \{y_r \oplus h_{k'}(\tau_r), \ldots, y_j \oplus h_{k'}(\tau_j)\}$**.** This follows from the assumption $\neg\mathsf{Bad}_{j,2}$.

Property 2**:** $y_{r-1} \oplus h_{k'}(\tau_{r-1}) \notin \{E_k^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r)), \ldots, E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j))\}$. We prove step by step. First by contradiction, suppose $y_{r-1} \oplus h_{k'}(\tau_{r-1}) = E_k^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r))$. Then by the inductive assumption, we get

$$
\begin{aligned}
y_{r-1} \oplus h_{k'}(\tau_{r-1}) &= F_r^{-1}(y_r \oplus h_{k'}(\tau_r)) \\
F_r(y_{r-1} \oplus h_{k'}(\tau_{r-1})) &= y_r \oplus h_{k'}(\tau_r) \\
E_k^{T_j,K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})) &= E_k^{T_j,K}(x_r \oplus h_{k'}(\tau_r)).
\end{aligned}
$$

Since $E_{k_0}^{T_j,K}$ is a permutation, this contradicts the assumption ($\neg\mathsf{Bad}_{j,1}$) that $x_{r-1} \oplus h_{k'}(\tau_{r-1}) \neq$

$x_r \oplus h_{k'}(\tau_r)$. Therefore we have $y_{r-1} \oplus h_{k'}(\tau_{r-1}) \neq E_k^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r))$. Now, we can write

$$
\begin{aligned}
&E_k^{T_j,K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})) \\
&= \mathsf{swap}_{E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \ldots \circ \mathsf{swap}_{E_k^{T_r,K}(x_{r+1} \oplus h_{k'}(\tau_{r+1})),\, y_{r+1} \oplus h_{k'}(\tau_{r+1})} (y_{r-1} \oplus h_{k'}(\tau_{r-1})) \\
&= F_{r+1}(y_{r-1} \oplus h_{k'}(\tau_{r-1})).
\end{aligned}
$$

By a similar prove-by-contradiction argument, using the inductive assumption on $r+1$, we obtain

$$y_{r-1} \oplus h_{k'}(\tau_{r-1}) \neq E_k^{T_r,K}(x_{r+1} \oplus h_{k'}(\tau_{r+1})).$$

Repeating this proof by contradiction step by step for $r+2, r+3, \ldots, j$, we eventually reach

$$y_{r-1} \oplus h_{k'}(\tau_{r-1}) \neq E_k^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),$$

which establishes Property 2.

By these two properties, we can get

$$E_k^{T_j,K}(x_{r-1} \oplus h_{k'}(\tau_{r-1})) = y_{r-1} \oplus h_{k'}(\tau_{r-1}).$$

Then by strong induction, the proposition holds for all $i \in \{1, \ldots, j\}$. $\square$

We now define a sequence

$$\mathbf{H}_0, \mathbf{H}_0^1, \mathbf{H}_0^2, \mathbf{H}_0^3, \mathbf{H}_1, \mathbf{H}_1^1, \ldots, \mathbf{H}_{q_C}^3 \tag{3.36}$$

of hybrid experiments. Each experiment begins with sampling uniform $\widetilde{\Pi} \in \mathcal{E}(\mathcal{T}, n)$ and $E \in \mathcal{E}(m, n)$, and a uniform key pair $K = (k, k') \in \{0,1\}^{m+\kappa}$. The remaining steps of each hybrid are as follows.

Experiment H_j.

1. Run $\mathcal{A}$, answering its classical queries using $\widetilde{\Pi}$ and its quantum queries using E, stopping immediately *before* its $(j+1)^{\text{st}}$ classical query. Let $T_j = \{(\tau_1, x_1, y_1), \ldots, (\tau_j, x_j, y_j)\}$ be the list of classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries by $\mathsf{LRW}_K[E^{T_j,K}]$ and its quantum queries by $E^{T_j,k}$.

Experiment H_j^1.

1. Run $\mathcal{A}$, answering its classical queries using $\widetilde{\Pi}$ and its quantum queries using E, until $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query (τ_{j+1}, x_{j+1}).

2. Choose uniform $s \in \{0,1\}^n$, and define $E^{(1)}$ as

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k \\ \left(E_k \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, s}\right)(x) & \text{if } k^* = k. \end{cases}$$

   Continue running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{LRW}_K[(E^{(1)})^{T_j,K}]$, and its quantum queries using $(E^{(1)})^{T_j,K}$.

Experiment H_j^2.

1. Run $\mathcal{A}$, answering its classical queries using $\widetilde{\Pi}$ and its quantum queries using E, until $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query (τ_{j+1}, x_{j+1}).

2. Define $E^{(1)}$ as in $\mathbf{H}_j^1$ and answer the $(j+1)^{\text{st}}$ using $\mathsf{LRW}_K[(E^{(1)})^{T_j,K}]$. Denote the response as y_{j+1}, i.e.,

$$y_{j+1} = \mathsf{LRW}_K[(E^{(1)})^j](\tau_{j+1}, x_{j+1}).$$

Continue running $\mathcal{A}$, answering its remaining classical queries using $\mathsf{LRW}_K[E^{T_{j+1},K}]$, and its quantum queries using $E^{T_{j+1},K}$.

Experiment $\mathbf{H}_j^3$.

1. Run $\mathcal{A}$, answering its classical queries using $\widetilde{\Pi}$ and its quantum queries using E, stopping immediately *after* its $(j+1)^{\text{st}}$ classical query. Let $T_{j+1} = \{(\tau_1, x_1, y_1), \ldots, (\tau_{j+1}, x_{j+1}, y_{j+1})\}$ be the classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $\mathsf{LRW}_K\left[E^{T_{j+1},K}\right]$ and its quantum queries using $E^{T_{j+1},K}$, i.e. $|E^{j+1}\rangle$.

We can compactly represent $\left\{\mathbf{H}_j, \mathbf{H}_j^1, \mathbf{H}_j^2, \mathbf{H}_j^3, \mathbf{H}_{j+1}\right\}$ as the experiment in which $\mathcal{A}$'s queries are answered using the following oracle sequences. We also define $(E^{(1)})^j$ to denote $(E_k^{(1)})^{T_j,K}$.

$$\begin{aligned}
\mathbf{H}_j &: \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[\ E^j\ ], |E^j\rangle \qquad, \mathsf{LRW}_K\left[E^j\right], |E^j\rangle, \cdots \\
\mathbf{H}_j^1 &: \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle\ , \mathsf{LRW}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots \\
\mathbf{H}_j^2 &: \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |E^{j+1}\rangle \quad, \mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}_j^3 &: \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \qquad \widetilde{\Pi} \qquad, |E^{j+1}\rangle \quad, \mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots \\
\mathbf{H}_{j+1} &: \ \underbrace{|E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle}_{j \text{ classical queries}}, \underbrace{\qquad \widetilde{\Pi} \qquad, |E\rangle \quad}_{(j+1)^{\text{st}} \text{ classical query}}, \underbrace{\mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots}_{q_C - j - 1 \text{ classical queries}}.
\end{aligned}$$

Then we establish the following bounds on the distinguishability of $\mathbf{H}_j$ and $\mathbf{H}_{j+1}$, step by step, for $0 \leq j < q_C$:

Lemma 3.8.5: $\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1 \wedge \neg\mathsf{Bad}_j] - \Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1 \wedge \neg\mathsf{Bad}_j\right]\right| \leq 4\sqrt{\frac{q_Q}{2^{m+n}}}$

Lemma 3.8.6: $\left|\Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1 \wedge \neg\mathsf{Bad}_j\right] - \Pr\left[\mathcal{A}(\mathbf{H}_j^2) = 1 \wedge \neg\mathsf{Bad}_j\right]\right| \leq \frac{2j}{2^n}$

Lemma 3.8.7: $\left|\Pr\left[\mathcal{A}(\mathbf{H}_j^2) = 1 \wedge \neg\mathsf{Bad}_j\right] - \Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1 \wedge \neg\mathsf{Bad}_j\right]\right| \leq \frac{2j}{2^n}$

Lemma 3.8.8: $\left|\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1 \wedge \neg\mathsf{Bad}_j\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \wedge \neg\mathsf{Bad}_{j+1}]\right| \leq 2 \cdot q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{m+n}}}$,

where the last equation, $q_{Q,j+1}$ is the expected number of queries $\mathcal{A}$ makes to P in the $(j+1)^{\text{st}}$ stage.

Using the above, we have

$$\begin{aligned}
&\left|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1]\right| \\
&= \left|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \neg\mathsf{Bad}_{q_C}] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \mathsf{Bad}_{q_C}]\right| \\
&\leq \left|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1 \wedge \neg\mathsf{Bad}_{q_C}]\right| + \left|\Pr[\mathsf{Bad}_{q_C}]\right| \\
&\leq \sum_{j=0}^{q_C-1} \left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1 \wedge \neg\mathsf{Bad}_j] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \wedge \neg\mathsf{Bad}_{j+1}]\right| + \left|\Pr[\mathsf{Bad}_{q_C}]\right| \\
&\leq \sum_{j=0}^{q_C-1} \left(4\sqrt{\frac{q_Q}{2^{m+n}}} + \frac{4j}{2^n} + 2 \cdot q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{m+n}}}\right) + \frac{2q_C^2}{2^n} \\
&\leq \frac{6q_C^2}{2^n} + \sum_{j=0}^{q_C-1} \left(4\sqrt{\frac{q_Q}{2^{m+n}}} + 2\sqrt{2} \cdot q_{Q,j+1}\sqrt{\frac{q_C}{2^{m+n}}}\right) \\
&\leq \frac{6q_C^2}{2^n} + \frac{4}{\sqrt{2^{m+n}}} \cdot \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right),
\end{aligned}$$

which concludes the proof. $\square$

We now prove Lemma 3.8.5, Lemma 3.8.6, Lemma 3.8.7 and Lemma 3.8.8.

Lemma 3.8.5. *For* $j = 0, \ldots, q_C$*,*

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1 \wedge \textsf{\textit{Bad}}_j] - \Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1 \wedge \textsf{\textit{Bad}}_j\right]\right| \leq 4\sqrt{\frac{q_Q}{2^{n+m}}}\,.$$

Proof. In this lemma, we bound the distinguishability of $\mathbf{H}_j$ and $\mathbf{H}_j^1$.

$$\begin{aligned}
\mathbf{H}_j: &\ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[\ \ E^j\ \ ], |E^j\rangle \qquad, \mathsf{LRW}_K\left[E^j\right], |E^j\rangle, \cdots \\
\mathbf{H}_j^1: &\ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle\ , \mathsf{LRW}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots.
\end{aligned}$$

Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j$ and $\mathbf{H}_j^1$. We construct from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for the resampling experiment of Lemma 3.4.1. Fixing D to be uniform distribution over $\{0,1\}^{m+2n}$ (so $\varepsilon = 2^{-(m+2n)}$ in Lemma 3.4.1), $\mathcal{D}$ does:

Phase 1: $\mathcal{D}$ is given quantum access to an ideal cipher E. It samples a uniform $\widetilde{\Pi} \leftarrow \mathcal{E}(\mathcal{T}, n)$ and then runs $\mathcal{A}$, answering its quantum queries with E and its classical queries with $\widetilde{\Pi}$ (in the appropriate directions), until $\mathcal{A}$ submits its $(j+1)^{\text{st}}$ classical query (τ_{j+1}, x_{j+1}). At that point, $\mathcal{D}$ has a list $T_j = \{(\tau_1, x_1, y_1), \ldots, (\tau_j, x_j, y_j)\}$ of the queries/answers $\mathcal{A}$ has made to its classical oracle thus far. If Bad_j, then set flag to 1; otherwise, set flag to 0.

Phase 2: $\mathcal{D}$ is given $(k, s_0, s_1) \leftarrow D$ and quantum oracle access to a cipher $E^{(b)}$. Then $\mathcal{D}$ selects the k' conditioned on $h_{k'}(\tau_{j+1}) = x_{j+1} \oplus s_0$ (at least one such k' must exist since $\kappa \geq n$), and sets $K = (k, k')$. It then continues running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)^{\text{st}}$) using $\mathsf{LRW}_K[(E^{(b)})^{T_j,K}]$, and its remaining quantum queries using $(E^{(b)})^{T_j,K}$. If $\mathsf{flag} = 1$, $\mathcal{D}$ outputs 0; otherwise, $\mathcal{D}$ outputs whatever $\mathcal{A}$ does.

Note that in phase 1, distinguisher $\mathcal{D}$ perfectly simulates experiments $\mathbf{H}_j$ and $\mathbf{H}_j^1$ for $\mathcal{A}$ until the point where $\mathcal{A}$ makes its $(j+1)^{\text{st}}$ classical query. In phase 2, we now argue that k' is uniformly distributed. For any $k^* \in \{0,1\}^\kappa$,

$$\begin{aligned}
\Pr_{k':h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0}[k'=k^*] &= \Pr_{k'\in\{0,1\}^\kappa}[k'=k^*|h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0] \\
&= \frac{\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0|k'=k^*]\cdot\Pr_{k'\in\{0,1\}^\kappa}[k'=k^*]}{\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0]} \\
&= \frac{\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0|k'=k^*]\cdot\Pr_{k'\in\{0,1\}^\kappa}[k'=k^*]}{\sum_{\widetilde{k}\in\{0,1\}^\kappa}\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0|k'=\widetilde{k}]\Pr_{\widetilde{k}}[k'=\widetilde{k}]} \\
&= \frac{\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0|k'=k^*]}{\sum_{\widetilde{k}\in\{0,1\}^\kappa}\Pr_{k'\in\{0,1\}^\kappa}[h_{k'}(\tau_{j+1})=x_{j+1}\oplus s_0|k'=\widetilde{k}]} \\
&= \frac{\Pr[h_{k^*}(\tau_{j+1})=x_{j+1}\oplus s_0]}{\sum_{\widetilde{k}\in\{0,1\}^\kappa}\Pr\left[h_{\widetilde{k}}(\tau_{j+1})=x_{j+1}\oplus s_0\right]} \\
&= \frac{1/2^\kappa}{\sum_{\widetilde{k}}1/2^\kappa} \\
&= \frac{1}{2^\kappa}
\end{aligned}$$

where the third equality follows from $\Pr_{k'}[k'=k^*]=\Pr_{k'}[k'=\widetilde{k}]=\frac{1}{2^\kappa}$. The second last equality holds because s_0 is drawn uniformly at random and h is uniform.

If $b=0$ and $\mathsf{flag}=0$, $\mathcal{D}$ gets access to $E^{(0)}=E$ in phase 2. Since $\mathcal{D}$ answers all quantum queries using $\left(E^{(0)}\right)^{T_j,K}$ and all classical queries using $\mathsf{LRW}_K\left[\left(E^{(0)}\right)^{T_j,K}\right]$, we see that $\mathcal{D}$ perfectly simulates $\mathbf{H}_j$ for $\mathcal{A}$ in that case. If, on the other hand, $b=1$ and $\mathsf{flag}=0$ in phase 2, then $\mathcal{D}$ gets access to $E^{(1)}$, where

$$E^{(1)}_{k^*}(x)=\begin{cases}E_{k^*}(x) & \text{if } k^*\neq k\\[2em] E_k\circ\mathsf{swap}_{s_0,s_1}(x) & \text{if } k^*=k.\end{cases}$$

Since $h_{k'}(\tau_{j+1}) := s_0 \oplus x_{j+1}$, it holds that

$$E^{(1)}_{k^*}(x) = \begin{cases} E_{k^*}(x) & \text{if } k^* \neq k \\ E_k \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, s_1}(x) & \text{if } k^* = k. \end{cases}$$

Moreover, the uniformity property of h and the fact that s_0 (and then $s_0 \oplus x_{j+1}$) is uniform imply that the joint distribution of k' and $s_0 \oplus x_{j+1}$ is equal to the joint distribution of $\widetilde{k}$ and $h_{\widetilde{k}}(\tau_{j+1})$ for a uniform $\widetilde{k}$. Thus, in this case $\mathcal{D}$ perfectly simulates $\mathbf{H}^1_j$ for $\mathcal{A}$. Applying Lemma 3.4.1 thus gives

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_j) = 1 \wedge \neg\mathsf{Bad}_j] - \Pr\big[\mathcal{A}(\mathbf{H}^1_j) = 1 \wedge \neg\mathsf{Bad}_j\big]| \\
&= |\Pr[\mathcal{D} = 1 \wedge \mathsf{flag} = 0 | b = 0] - \Pr[\mathcal{D} = 1 \wedge \mathsf{flag} = 0 | b = 1]| \\
&\leq |\Pr[\mathcal{D} = 1 | b = 0] - \Pr[\mathcal{D} = 1 | b = 1]| \\
&\leq 4\sqrt{2^n \cdot q_Q \cdot \varepsilon} \\
&\leq 4\sqrt{\frac{q_Q}{2^{m+n}}},
\end{aligned}$$

where the first inequality holds due to the independence of $\mathcal{D}$'s output from $\mathcal{A}$'s output when $\mathsf{flag} = 1$, and $\Pr[\mathsf{flag} = 1]$ being the same for $b = 0$ and $b = 1$. □

Lemma 3.8.6. *For $j = 0, \ldots, q_C$,*

$$\left|\Pr\big[\mathcal{A}(\mathbf{H}^1_j) = 1 \wedge \neg \mathit{Bad}_j\big] - \Pr\big[\mathcal{A}(\mathbf{H}^2_j) = 1 \wedge \neg \mathit{Bad}_j\big]\right| \leq \frac{2j}{2^n}.$$

Proof. Next, we bound the distinguishability of $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$.

$$\mathbf{H}_j^1: \ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |(E^{(1)})^j\rangle \ \ , \mathsf{LRW}_K[(E^{(1)})^j], (E^{(1)})^j, \cdots$$

$$\mathbf{H}_j^2: \ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |E^{j+1}\rangle \quad , \mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots.$$

We first prove the following proposition.

Proposition 6. *For all $i \in \{1, \ldots, j\}$ and all $r \in \{0, \ldots, j\}$*

$$(E_k^{(1)})^{T_r,K}(x_i \oplus h_{k'}(\tau_i)) \ = \ E_k^{T_r,K}(x_i \oplus h_{k'}(\tau_i)),$$

when $s \notin \{x_1 \oplus h_{k'}(\tau_1), \cdots x_j \oplus h_{k'}(\tau_j)\}$ and $x_{j+1} \oplus h_{k'}(\tau_{j+1}) \notin \{x_1 \oplus h_{k'}(\tau_1), \cdots x_j \oplus h_{k'}(\tau_j)\}$.

Proof. We will do a proof by induction on r. We start with the base case $r = 0$. By the constraint on s and $x_{j+1} \oplus h_{k'}(\tau_{j+1})$, we have that for all $i \in \{1, \cdots, j\}$

$$\begin{aligned}(E_k^{(1)})^{T_0,K}(x_i \oplus h_{k'}(\tau_i)) &:= E_k^{(1)}(x_i \oplus h_{k'}(\tau_i)) \\ &= E_k \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, s}(x_i \oplus h_{k'}(\tau_i)) \\ &= E_k(x_i \oplus h_{k'}(\tau_{j+1})).\end{aligned}$$

Assume for some $r - 1 \geq 0$ that

$$(E_k^{(1)})^{T_{r-1},K}(x_i \oplus h_{k'}(\tau_i)) = E_k^{T_{r-1},K}(x_i \oplus h_{k'}(\tau_i)).$$

Then by the definition,

$$
\begin{aligned}
(E_k^{(1)})^{T_r,K}(x_i \oplus h_{k'}(\tau_i)) &= \mathsf{swap}_{(E_k^{(1)})^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r)),\, y_r \oplus h_{k'}(\tau_r)} \circ (E_k^{(1)})^{T_{r-1},K}(x_i \oplus h_{k'}(\tau_i)) \\
&= \mathsf{swap}_{E_k^{T_{r-1},K}(x_r \oplus h_{k'}(\tau_r)),\, y_r \oplus h_{k'}(\tau_r)} \circ E_k^{T_{r-1},K}(x_i \oplus h_{k'}(\tau_i)) \\
&= E_k^{T_r,K}(x_i \oplus h_{k'}(\tau_i)).
\end{aligned}
$$

By induction, the proposition holds for all $r \in \{0, \ldots, j\}$. $\square$

In particular, for all $i \in \{1, \ldots, j\}$,

$$
(E_k^{(1)})^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)) = E_{k_0}^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)).
$$

To analyze the distinguishability between $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$, it suffices to examine the distinguishability between the quantum oracles $(E^{(1)})^j$ and E^{j+1}. We begin under the constraint that

- $s \notin \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$
- $x_{j+1} \oplus h_{k'}(\tau_{j+1}) \notin \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$.

In this case, the response to the $(j+1)^{\text{st}}$ classical query is identical in both $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$:

$$
\begin{aligned}
y_{j+1} &\stackrel{\text{def}}{=} \mathsf{LRW}_K[(E^{(1)})^{T_j,K}](\tau_{j+1}, x_{j+1}) \\
&= (E_k^{(1)})^{T_j,K}(x_{j+1} \oplus h_{k'}(\tau_{j+1})) \oplus h_{k'}(\tau_{j+1}) \\
&= \mathsf{swap}_{(E_k^{(1)})^{T_{j-1},K}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ (E_k^{(1)})^{T_{j-1},K}(x_{j+1} \oplus h_{k'}(\tau_{j+1})) \oplus h_{k'}(\tau_{j+1}) \\
&= \left(\prod_{i=j}^{1} \mathsf{swap}_{(E_k^{(1)})^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)),\, y_i \oplus h_{k'}(\tau_i)} \right) \circ E_k^{(1)}(x_{j+1} \oplus h_{k'}(\tau_{j+1})) \oplus h_{k'}(\tau_{j+1}) \\
&= \left(\prod_{i=j}^{1} \mathsf{swap}_{(E_k)^{T_{i-1},K}(x_i \oplus h_{k'}(\tau_i)),\, y_i \oplus h_{k'}(\tau_i)} \right) \circ E_k(s) \oplus h_{k'}(\tau_{j+1}) \\
&= E_k^{T_j,K}(s) \oplus h_{k'}(\tau_{j+1}).
\end{aligned}
$$

By rearranging $s = \left(E_{k_0}^{T_j,K}\right)^{-1} (y_{j+1} \oplus h_{k'}(\tau_{j+1}))$. It follows that for any $x \in \{0,1\}^n$,

$$
\begin{aligned}
&(E^{(1)})^j(x) \\
&= \mathsf{swap}_{(E_k^{(1)})^{j-1}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \cdots \circ \mathsf{swap}_{E_k^{(1)}(x_1 \oplus h_{k'}(\tau_1)),\, y_1 \oplus h_{k'}(\tau_1)} \\
&\qquad \circ E_k \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, s}(x) \\
&= \mathsf{swap}_{E_k^{j-1}(x_j \oplus h_{k'}(\tau_j)),\, y_j \oplus h_{k'}(\tau_j)} \circ \cdots \circ \mathsf{swap}_{E_k(x_1 \oplus h_{k'}(\tau_1)),\, y_1 \oplus h_{k'}(\tau_1)} \circ E_k \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, s}(x) \\
&= E^j \circ \mathsf{swap}_{x_{j+1} \oplus h_{k'}(\tau_{j+1}),\, (E^j)^{-1}(y_{j+1} \oplus h_{k'}(\tau_{j+1}))}(x) \\
&= \mathsf{swap}_{E^j(x_{j+1} \oplus h_{k'}(\tau_{j+1})),\, y_{j+1} \oplus h_{k'}(\tau_{j+1})} \circ E^j(x) \\
&= E^{j+1}(x),
\end{aligned}
$$

where the second equality comes from Proposition 6. Under these constraints, we have $(E^{(1)})^j \equiv E^{j+1}$, making the hybrids $\mathbf{H}_j^1$ and $\mathbf{H}_j^2$ perfectly indistinguishable. We then introduce the constraints

Bad_j, which is independent of the new constraints on s and $x_{j+1} \oplus h_{k'}(\tau_{j+1})$ and has the same probability in two hybrids. Thus, these constraints do not affect the perfect indistinguishability. To bound the distinguishability of the hybrids with $\neg\mathsf{Bad}_j$, we analyze the probability of these new constraints occurring:

$$
\begin{aligned}
&\left|\Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1 \wedge \neg\mathsf{Bad}_j\right] - \Pr\left[\mathcal{A}(\mathbf{H}_j^2) = 1 \wedge \neg\mathsf{Bad}_j\right]\right| \\
&\leq \left|\Pr\left[\mathcal{A}(\mathbf{H}_j^1) = 1\right] - \Pr\left[\mathcal{A}(\mathbf{H}_j^2) = 1\right]\right| \\
&\leq \Pr[s \in S] + \Pr[x_{j+1} \oplus h_{k'}(\tau_{j+1}) \in S] \\
&= \frac{2j}{2^n},
\end{aligned}
$$

where $S = \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$. Since s is uniformly distributed over $\{0,1\}^n$, the probability of the first term is $\frac{j}{2^n}$. For the second term,

$$
x_{j+1} \oplus h_{k'}(\tau_{j+1}) \in S \;\; \Rightarrow \;\; \bigcup_{i=1}^{j} \left\{ h_{k'}(\tau_{j+1}) \oplus h_{k'}(\tau_i) = x_{j+1} \oplus x_i \right\},
$$

and by the XOR-universality of h, this occurs with probability $\frac{j}{2^n}$. □

Lemma 3.8.7. *For $j = 0, \ldots, q_C$,*

$$
\left|\Pr\left[\mathcal{A}(\mathbf{H}_j^2) = 1 \wedge \textsf{\textit{Bad}}_j\right] - \Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1 \wedge \textsf{\textit{Bad}}_j\right]\right| \leq \frac{2j}{2^n}\,.
$$

Proof. We bound the distinguishability of $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$.

$$\mathbf{H}_j^2: \ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \mathsf{LRW}_K[(E^{(1)})^j], |E^{j+1}\rangle, \mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots$$

$$\mathbf{H}_j^3: \ \ |E\rangle, \widetilde{\Pi}, |E\rangle, \cdots, \widetilde{\Pi}, |E\rangle, \qquad \widetilde{\Pi} \qquad, |E^{j+1}\rangle, \mathsf{LRW}_K[E^{j+1}], |E^{j+1}\rangle, \cdots.$$

We observe that $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$ differ only in the response to the $(j+1)^{\text{st}}$ classical query. In $\mathbf{H}_j^2$, as specified in Lemma 3.8.6, under the constraint

- $s \notin \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$,
- $x_{j+1} \oplus h_{k'}(\tau_{j+1}) \notin \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$,

the $(j+1)$-st query is answered as

$$y_{j+1} = \mathsf{LRW}_K\big[(E^{(1)})^{T_j,K}\big](\tau_{j+1}, x_{j+1}) \ = \ E_k^{T_j,K}(s) \oplus h_{k'}(\tau_{j+1}).$$

as shown in Lemma 3.8.6. We then introduce the event $\neg\mathsf{Bad}_j$. Recall Proposition 5, under $\neg\mathsf{Bad}_j$, we have

$$y_i \ = \ E_k^{T_j,K}(x_i \oplus h_{k'}(\tau_i)) \oplus h_{k'}(\tau_i), \quad \forall i \in [1, j],$$

and since $s \in \{0,1\}^n \setminus \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$, it follows that in $\mathbf{H}_j^2$,

$$y_{j+1} = E_k^{T_j,K}(s) \oplus h_{k'}(\tau_{j+1}) \in \{0,1\}^n \setminus \{y_1 \oplus h_{k'}(\tau_1) \oplus h_{k'}(\tau_{j+1}), \ldots, y_j \oplus h_{k'}(\tau_j) \oplus h_{k'}(\tau_{j+1})\}.$$

Moreover, because $E_k^{T_j,K}$ is a permutation, the mapping

$$x_i \oplus h_{k'}(\tau_i) \;\mapsto\; y_i = E_k^{T_j,K}(x_i \oplus h_{k'}(\tau_i)) \oplus h_{k'}(\tau_i)$$

is injective. Since s is chosen uniformly from $\{0,1\}^n \setminus \{x_1 \oplus h_{k'}(\tau_1), \ldots, x_j \oplus h_{k'}(\tau_j)\}$, $E_k^{T_j,K}(s)$ is uniformly distributed over

$$\{0,1\}^n \setminus \{E_k^{T_j,K}(x_1 \oplus h_{k'}(\tau_1)), \ldots, E_k^{T_j,K}(x_j \oplus h_{k'}(\tau_j))\} = \{0,1\}^n \setminus \{y_1 \oplus h_{k'}(\tau_1), \ldots, y_j \oplus h_{k'}(\tau_j)\}.$$

It follows that the output y_{j+1} is uniformly distributed over $\{0,1\}^n \setminus \{y_1 \oplus h_{k'}(\tau_1) \oplus h_{k'}(\tau_{j+1}), \ldots, y_j \oplus h_{k'}(\tau_j) \oplus h_{k'}(\tau_{j+1})\}$ in $\mathbf{H}_j^2$.

In contrast, in $\mathbf{H}_j^3$ the same query is answered with

$$y_{j+1} \;=\; \widetilde{\Pi}(\tau_{j+1}, x_{j+1}).$$

Next, we conduct a case-by-case analysis to demonstrate that y_{j+1} has the same distribution conditioned on all previous queries is identical in both $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$.

- **Case** $\tau_{j+1} \notin \{\tau_1, \ldots, \tau_j\}$. In $\mathbf{H}_j^2$, since h is XOR-universal, for all $i \in [1, j]$, the value $h_{k'}(\tau_i) \oplus h_{k'}(\tau_{j+1})$ is uniformly distributed over $\{0,1\}^n$. Consequently, y_{j+1} is uniformly distributed over $\{0,1\}^n$. In $\mathbf{H}_j^3$, as τ_{j+1} is new, $\widetilde{\Pi}_{\tau_{j+1}}$ represents a fresh permutation, and thus $y_{j+1} = \widetilde{\Pi}_{\tau_{j+1}}(x_{j+1})$ is uniformly distributed over $\{0,1\}^n$.

- **Case** $\tau_{j+1} \in \{\tau_1, \ldots, \tau_j\}$. Assume $\tau_{j+1} = \tau_a$ for some $a \in [1, j]$. In $\mathbf{H}_j^2$, we have $y_{j+1} \neq y_a \oplus h_{k'}(\tau_a) \oplus h_{k'}(\tau_a) = y_a$. For the remaining elements, since h is XOR-universal, they are uniformly distributed over $\{0,1\}^n$. Thus, y_{j+1} is uniformly distributed over $\{0,1\}^n \setminus \{y_a\}$. In $\mathbf{H}_j^3$, note that repeated queries are not permitted, so $x_{j+1} \neq x_a$. Therefore, $y_{j+1} = \widetilde{\Pi}_a(x_{j+1}) \neq \widetilde{\Pi}_a(x_a) = y_a$. Consequently, y_{j+1} is also uniformly distributed over $\{0,1\}^n \setminus \{y_a\}$. We note that τ_{j+1} may equal multiple elements in $\{\tau_1, \dots, \tau_j\}$, but the argument still holds.

Moreover, in both $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$, the construction of E^{j+1} follows exactly the same procedure, i.e., from the first $j+1$ classical input–output pairs and E as specified in Equation 3.35. Consequently, the two hybrids yield identical distributions under constraints described before.

Therefore, the probability with which $\mathcal{A}$ can distinguish $\mathbf{H}_j^2$ and $\mathbf{H}_j^3$ is

$$
\begin{aligned}
&|\Pr\big[\mathcal{A}(\mathbf{H}_j^2) = 1 \wedge \neg\mathsf{Bad}_j\big] - \Pr\big[\mathcal{A}(\mathbf{H}_j^3) = 1 \wedge \neg\mathsf{Bad}_j\big]| \\
&\leq |\Pr\big[\mathcal{A}(\mathbf{H}_j^2) = 1 | \neg\mathsf{Bad}_j\big] - \Pr\big[\mathcal{A}(\mathbf{H}_j^3) = 1 | \neg\mathsf{Bad}_j\big]| \\
&\leq \Pr[s \in S] + \Pr[x_{j+1} \oplus h_{k'}(\tau_{j+1}) \in S] \\
&= \frac{2j}{2^n},
\end{aligned}
$$

where $S = \{x_1 \oplus h_{k'}(\tau_1), \dots, x_j \oplus h_{k'}(\tau_j)\}$. $\square$

Lemma 3.8.8. *For $j = 0, \dots, q_C - 1$,*

$$
\Pr\big[\mathcal{A}(\mathbf{H}_j^3) = 1 \wedge \mathsf{Bad}_j\big] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \wedge \mathsf{Bad}_{j+1}]| \leq 2 \cdot q_{Q,j+1} \sqrt{\frac{2(j+1)}{2^{m+n}}},
$$

where $q_{Q,j+1}$ is the expected number of queries $\mathcal{A}$ makes to P in the $(j+1)^{st}$ stage in the ideal world (i.e., in $\mathbf{H}_{q_C}$).

Proof. Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j^3$ and $\mathbf{H}_{j+1}$. We construct from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for the experiment from Lemma 3.3.1:

Phase 1: $\mathcal{D}$ samples uniform $\widetilde{\Pi} \in \mathcal{E}(\mathcal{T}, n)$ and $E \in \mathcal{E}(m, n)$. It then runs $\mathcal{A}$, answering its quantum queries using E and its classical queries using $\widetilde{\Pi}$, until after it responds to $\mathcal{A}$'s $(j+1)^{\text{st}}$ classical query. If Bad_{j+1}, set $\mathsf{flag} = 1$; otherwise, set $\mathsf{flag} = 0$. Let $T_{j+1} = \{(\tau_1, x_1, y_1), \ldots, (\tau_{j+1}, x_{j+1}, y_{j+1})\}$ be the list of input/output pairs $\mathcal{A}$ received from its classical oracle thus far. $\mathcal{D}$ defines $F(a, K^*, x) := E^a_{k^*}(x)$ for $a \in \{1, -1\}$. It also define the following randomized algorithm $\mathcal{B}$: sample $K \leftarrow \{0,1\}^m \times \{0,1\}^\kappa$ and then compute the blinded set B to be the input/output pairs that are reprogrammed so that $F^{(B)}(a, K^*, x) = \left(E^{T_{j+1},K}_{k^*}\right)^a(x)$ for all a, K^*, x, i.e.,

$$B = \left\{((a, K^*, x), \left(E^{T_{j+1},K}_{k^*}\right)^a(x))\right\}.$$

Phase 2: $\mathcal{B}$ is run to generate B and $\mathcal{D}$ is given quantum access to an oracle F_b. $\mathcal{D}$ resumes running $\mathcal{A}$, answering its quantum queries using F_b. Phase 2 ends when $\mathcal{A}$ makes its next (i.e., $((j+2)^{\text{nd}})$ classical query.

Phase 3: $\mathcal{D}$ is given the randomness used by $\mathcal{B}$ to generate K. It resumes running $\mathcal{A}$, answering its classical queries using $\mathsf{LRW}_K\left[E^{T_{j+1},K}\right]$ and its quantum queries using $E^{T_{j+1},K}$. Finally, if $\mathsf{flag} = 1$, $\mathcal{D}$ outputs 0; otherwise, it outputs whatever $\mathcal{A}$ outputs.

It is immediate that if $b = 0$ and $\mathsf{flag} = 0$ (i.e., $\mathcal{D}$'s oracle in phase 2 is $F_0 = F$), then $\mathcal{A}$'s output is identically distributed to its output in $\mathbf{H}_{j+1}$, whereas if $b = 1$ (i.e., $\mathcal{D}$'s oracle in phase 2 is $F_1 = F^{(B)}$) and and $\mathsf{flag} = 0$, then $\mathcal{A}$'s output is identically distributed to its output in $\mathbf{H}^3_j$. It follows that $|\Pr[\mathcal{A}(\mathbf{H}^3_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]|$ is equal to the distinguishing advantage of $\mathcal{D}$ in the reprogramming experiment of Lemma 3.3.1. To bound this quantity, we bound the parameter ε and the expected number of queries made by $\mathcal{D}$ in phase 2 (when $F = F_0$.)

The value of ε can be bounded using the definition of $E^{T_{j+1},K}$ and the fact that $F^{(B)}(a, K^*, x) = \left(E_{k^*}^{T_{j+1},K}\right)^a(x)$. Fixing E and T_{j+1}, the probability that any particular input (a, K^*, x) is reprogrammed is at most the probability (over K) that it is in the set

$$\left\{ \begin{array}{c} (1, k, x_i \oplus h_{k'}(\tau_i))\,, \left(1, k, E_k^{-1}\left(y_i \oplus h_{k'}(\tau_i)\right)\right), \\ (-1, k, E_k\left(x_i \oplus h_{k'}(\tau_i)\right)), (-1, k, y_i \oplus h_{k'}(\tau_i)) \end{array} \right\}_{i=1}^{j+1}.$$

Since $h_{k'}(\tau_i)$ is uniform, taking a union bound gives $\varepsilon \leq 2(j+1)/2^{m+n}$.

The expected number of queries made by $\mathcal{D}$ in phase 2 when $F = F_0$ is equal to the expected number of queries made by $\mathcal{A}$ in its $(j+1)^{\text{st}}$ stage in $\mathbf{H}_{j+1}$. Since $\mathbf{H}_{j+1}$ and $\mathbf{H}_{q_C}$ are identical until after the $(j+1)^{\text{st}}$ is complete, this is precisely $q_{Q,j+1}$.

Applying Lemma 3.3.1 thus gives

$$\begin{aligned}
&|\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1 \land \neg\mathsf{Bad}_j\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \land \neg\mathsf{Bad}_{j+1}]| \\
&= |\Pr\left[\mathcal{A}(\mathbf{H}_j^3) = 1 \land \neg\mathsf{Bad}_{j+1}\right] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1 \land \neg\mathsf{Bad}_{j+1}]| \\
&= |\Pr[\mathcal{D} = 1 \land \mathsf{flag} = 0 | b = 0] - \Pr[\mathcal{D} = 1 \land \mathsf{flag} = 0 | b = 1]| \\
&\leq |\Pr[\mathcal{D} = 1 | b = 0] - \Pr[\mathcal{D} = 1 | b = 1]| \\
&\leq 2 \cdot q_{Q,j+1}\sqrt{\frac{2(j+1)}{2^{m+n}}},
\end{aligned}$$

where the first equality arises because the $(j+1)^{\text{st}}$ query is answered identically by $\widetilde{\Pi}$ in both worlds. The first inequality holds due to the independence of $\mathcal{D}$'s output from $\mathcal{A}$'s output when $\mathsf{flag} = 1$, and $\Pr[\mathsf{flag} = 1]$ being the same for $b = 0$ and $b = 1$. $\square$

3.8.5 Quantum oracle constructions in the search reduction proof

We describe how to implement the functions E' and E'^{-1} using a quantum oracle f in the proof of Theorem 3.7.6, given the full descriptions of $P(\cdot)$ and $E(\cdot,\cdot)$. The definitions are:

$$E'(k,\tau) = \begin{cases} P(\tau) & \text{if } f(k) = 1, \\ E(k,\tau) & \text{if } f(k) \neq 1, \end{cases} \quad E'^{-1}(k,\tau) = \begin{cases} P^{-1}(\tau) & \text{if } f(k) = 1, \\ E^{-1}(k,\tau) & \text{if } f(k) \neq 1. \end{cases}$$

Using quantum oracle access to $\mathcal{O}_f$, we can construct $\mathcal{O}_{E'}$ and $\mathcal{O}_{E'^{-1}}$. Below is the construction of $\mathcal{O}_{E'}$:

$$\begin{aligned}
& |k\rangle\,|\tau\rangle\,|y\rangle \\
\xrightarrow{\text{add aux registers}}\; & |k\rangle_1\,|\tau\rangle_2\,|y\rangle_3\,|0\rangle_4\,|0\rangle_5\,|0\rangle_6\,|0\rangle_7 \\
\xrightarrow{\mathcal{O}_{E,1,2,7}\mathcal{O}_{P,1,6}X_5\mathcal{O}_{f,1,5}\mathcal{O}_{f,1,4}}\; & |k\rangle\,|\tau\rangle\,|y\rangle\,|f(k)\rangle\,|\overline{f(k)}\rangle\,|P(x)\rangle\,|E(k,\tau)\rangle \\
\xrightarrow{\text{CCNOT}_{6,4,3}}\; & |k\rangle\,|\tau\rangle\,|y \oplus (P(\tau)\cdot f(k))\rangle\,|f(k)\rangle\,|\overline{f(k)}\rangle\,|P(x)\rangle\,|E(k,\tau)\rangle \\
\xrightarrow{\text{CCNOT}_{7,5,3}}\; & |k\rangle\,|\tau\rangle\,|y \oplus (P(\tau)\cdot f(k)) \oplus (E(k,\tau)\cdot \overline{f(k)})\rangle\,|f(k)\rangle\,|\overline{f(k)}\rangle\,|P(x)\rangle\,|E(k,\tau)\rangle \\
\xrightarrow{\mathcal{O}_{E,1,2,7}\mathcal{O}_{P,1,6}X_5\mathcal{O}_{f,1,5}\mathcal{O}_{f,1,4}}\; & |k\rangle\,|\tau\rangle\,|y \oplus (P(\tau)\cdot f(k)) \oplus (E(k,\tau)\cdot \overline{f(k)})\rangle\,|0\rangle\,|0\rangle\,|0\rangle\,|0\rangle \\
\xrightarrow{\text{drop aux}}\; & |k\rangle\,|\tau\rangle\,|y \oplus (P(\tau)\cdot f(k)) \oplus (E(k,\tau)\cdot \overline{f(k)})\rangle
\end{aligned}$$

The construction of $\mathcal{O}_{E'^{-1}}$ follows the same process, substituting P^{-1} and E^{-1} in place of P and E, respectively:

$$|k\rangle\,|\tau\rangle\,|y\rangle \longrightarrow |y \oplus (P^{-1}(\tau)\cdot f(k)) \oplus (E^{-1}(k,\tau)\cdot \overline{f(k)})\rangle\,.$$

Chapter 4: On the Two-sided Permutation Inversion Problem

The limitations identified in the previous chapter are not specific to block-cipher constructions, but instead reflect more fundamental constraints at a lower level. This chapter studies the two-sided permutation inversion problem in the quantum query model. We show that even when a quantum adversary is granted quantum oracle access or equipped with quantum advice, the inversion task does not become substantially easier. These results highlight intrinsic barriers to quantum speedups that are independent of any particular cryptographic construction.

4.1 Summary of contribution

This work studies the two-sided permutation inversion problem in the quantum query model, where an algorithm is given quantum oracle access to both a permutation and its inverse, except that the inverse oracle rejects queries on the challenge image. 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. In particular, even with inverse access and quantum advice, the problem retains essentially the same hardness as standard permutation inversion, highlighting the hardness of inversion as a primitive in quantum query complexity and cryptographic applications.

4.2 Background

4.2.1 The permutation inversion problem

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

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

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

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

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

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

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.

Remark. We note that the definitions of the permutation inversion problem with the adaptive restriction of challenge distribution are included in Section 4.4 for completeness. Results on this variant will not be covered in this section. For detailed information, please refer to [ABPS24].

4.2.2 Organization

In Section 4.3, we present technical preliminaries, including the swapping lemma and quantum random access codes (QRAC), for subsequent proof. In Section 4.4, we introduce several definitions of the permutation inversion problem, with both auxiliary information and adaptive restriction of challenge distribution. Within Section 4.5, we show methods for amplifying the success probability of inversion in the non-adaptive case. Subsequently, in Section 4.6, we illustrate a reduction from search-to-decision with auxiliary information. The amplification and reduction are then utilized to derive lower bounds, as shown in Section 4.7. Finally, in Section 4.8, we propose a novel security notion, called one-way-QCCRA2, and establish the security of two common schemes under this notion, subject to specific conditions.

4.2.3 Related work

Previous works have considered the quantum-query *function* inversion problem [HXY19, CLQ19, CGLQ20, DKRS23, Liu23]. A number of papers gave lower bounds and time-space tradeoffs for the (one-sided) quantum-query permutation inversion problem, with and without advice [Amb02, Nay10, Ros21, NABT14, HXY19, CLQ19, FK15, BY23]. The relevant highlights among these are summarized in Table 4.1.

We remark that some of these previous works [CX21, CLQ19, NABT14] do not fully address the average-case setting. Specifically, they deal with inverters that are "restricted" in the following manner. First, the inverter is said to "invert y for π" if it succeeds in the inversion experiment for the specific pair (π, y) with probability at least $2/3$. Second, the inverter is said to "invert a δ-fraction of inputs" if $\Pr_{\pi,y}[\text{the inverter inverts } y \text{ for } \pi] \geq \delta$. This type of inverter is clearly captured by our notion above: it is an $(S, T, 2\delta/3)$-SPI. However, there are successful inverters of interest that are captured by our definition but are not restricted. For example, in a cryptographic context, one would definitely be concerned about adversaries that can invert every (π, y) with a probability of exactly $1/n$. Such an adversary is clearly a $(S, T, 1/n)$-SPI, but is not a restricted inverter for any value of δ. Other works also consider the general average-case (e.g., [CGLQ20, Liu23, HXY19]) but without two-way oracle access. Note that the lower bound for restricted adversaries described in [NABT14, CLQ19] can be translated to the more general lower bound in a black box way by applying our amplification procedure described in Lemma 4.5.2.

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

Table 4.1: Summary of previous work on permutation inversion with advice. Success probability is denoted by ϵ. Note that $\epsilon = O(1)$ in [NABT14].

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

Another novelty of our work is that we give lower bounds and time-space tradeoffs for the decision problem (rather than just search). While prior work [CGLQ20] also considered the general decision game, their generic framework crucially relies on compressed oracles for random *functions* [Zha19], and hence does not directly extend to permutation inversion. More recent work has significantly broadened the scope of compressed-oracle techniques by extending them to permutations [Car25]. A natural direction for future work is to explore whether these new tools can be used to develop a general framework for proving time-space tradeoffs for decision problems in the permutation setting.

We remark that the notion of two-way quantum accessibility to a random permutation has been considered in other works; for example, [ABKM22,ABK+24] 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.

4.3 Technical preliminaries

4.3.1 Basic probabilistic lemmas

In this section we collect a series of known probabilistic results, which we used in our main proofs. We first record some basic lemmas about the behavior of certain types of random variables.

Lemma 4.3.1 (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 its expected value. Then for any $\delta > 0$,*

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

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

Lemma 4.3.2 (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

$$(1-\theta) \cdot \mathbb{I}_{[X \geq \theta]} \geq X - \theta, \tag{4.1}$$

where $\mathbb{I}_{[X \geq \theta]}$ is the indicator function for the event that $X \geq \theta$. Suppose that $X \geq \theta$. Then, Equation 4.1 amounts to $1-\theta \geq X-\theta$, which is satisfied because $X \leq 1$. Now suppose that $X < \theta$. In this case Equation 4.1 amounts to $0 \geq X - \theta$, which is satisfied whenever $X \geq 0$. Taking the expectation over Equation 4.1 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 4.3.3 (Averaging argument)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be any finite sets and let $\Omega : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ be a predicate. Suppose that $\Pr_{x,y}[\Omega(x,y) = 1] \geq \epsilon$, for some $\epsilon \in [0,1]$, where x is chosen uniformly at random in $\mathcal{X}$. Let $\theta \in (0,1)$. Then, there exists a subset $\mathcal{X}_\theta \subseteq \mathcal{X}$ of size $|\mathcal{X}_\theta| \geq (1-\theta) \cdot \epsilon |\mathcal{X}|$ such that*

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

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

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

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

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

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

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

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

4.3.2 Swapping lemma

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

Lemma 4.3.4 (Swapping Lemma, [Vaz98])**.** *Let $f, g : \mathcal{X} \to \mathcal{Y}$ be functions with $f(x) = g(x)$ for all $x \notin \mathcal{S}$, where $\mathcal{S} \subseteq \mathcal{X}$. Let $|\Psi_f\rangle$ and $|\Psi_g\rangle$ denote the final states of a quantum algorithm $\mathcal{A}$ with quantum oracle access to the functions f and g, respectively. Then,*

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

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

4.3.3 Lower bounds for quantum random access codes

Quantum random access codes [Wie83, ANTSV99, ALMO08] are a means of encoding classical bits into (potentially fewer) qubits. We use the following variant from [CLQ19].

Definition 4.3.5 (Quantum random access codes with variable length)**.** *Let* N *be an integer and let* $\mathcal{F}_N = \{f : [N] \to \mathcal{X}_N\}$ *be an ensemble of functions over some finite set* $\mathcal{X}_N$*. A quantum random access code with variable length* $(\mathsf{QRAC\text{-}VL})$ *for* $\mathcal{F}_N$ *is a pair* $(\mathsf{Enc}, \mathsf{Dec})$ *consisting of a quantum encoding algorithm* Enc *and a quantum decoding algorithm* Dec*:*

- $\mathsf{Enc}(f; R)$*: The encoding algorithm takes as input a function* $f \in \mathcal{F}_N$ *together with a set of random coins* $R \in \{0,1\}^*$*, and outputs a quantum state* ρ *on* $\ell = \ell(f)$ *many qubits (where* ℓ *may depend on* f*).*

- $\mathsf{Dec}(\rho, x; R)$*: The decoding algorithm takes as input a state* ρ*, an element* $x \in [N]$ *and random coins* $R \in \{0,1\}^*$ *(same randomness used for the encoding), and seeks to output* $f(x)$*.*

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

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

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

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

Theorem 4.3.6 ([CLQ19], Corollary 1). *For any* QRAC-VL *for the set of permutations $\mathcal{S}_N$ (of the set $[N]$) with $\delta = 1 - k/N$ for some $k = \Omega(1/N)$, we have*

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

4.4 The permutation inversion problem

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

Definition 4.4.1. *Let $m, n \in \mathbb{N}$ and $M = 2^m$, $N = 2^n$. An adaptive search-version permutation inverter (aSPI) is a pair $\mathsf{aS} = (\mathsf{aS}_0, \mathsf{aS}_1)$ of quantum algorithms, where*

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

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

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

1. *(sample coins)* a uniformly random permutation $\pi : [N] \to [N]$ and a uniformly random string $r \leftarrow \{0,1\}^*$ are sampled;

2. *(prepare advice)* aS_0 is run, producing a pair consisting of a quantum state and a string $(\rho_{\pi,r,\mu}, \mu) \leftarrow \mathsf{aS}_0(\pi, r)$;

3. *(sample instance)* a random image $y \in [N]$ is generated by first sampling a random string $x \leftarrow \{0,1\}^{n-m}$ and then letting $y = \pi(x\|\mu)$;

4. *(invert)* aS_1 is run with the oracles below, and produces a candidate preimage x^*.

$$\mathcal{O}_\pi : |w\rangle\,|z\rangle \to |w\rangle\,|z \oplus \pi(w)\rangle \qquad \mathcal{O}_{\pi^{-1}_{\perp y}} : |w\rangle\,|z\rangle \to |w\rangle\,|z \oplus \pi^{-1}_{\perp y}(w)\rangle\,, \tag{4.2}$$

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

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

To keep the notation simple, we write this entire process as $x^* \leftarrow \mathsf{aS}_1^{\pi_{\perp y}}(\rho_{\pi,r,\mu}, \mu, y, r)$. We will use $\pi_{\perp y}$ to denote simultaneous access to the two oracles in Equation 4.2 throughout the paper.

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

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

Definition 4.4.2. *An (S,T,ϵ)-aSPI is a search-version adaptive permutation inverter* $\mathsf{aS} = (\mathsf{aS}_0, \mathsf{aS}_1)$ *satisfying all of the following:*

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

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

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

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

Definition 4.4.3. *A search-version permutation inverter (SPI)* $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ *is defined as an aSPI with $m = 0$. An (S,T,ϵ)-SPI is an (S,T,ϵ)-aSPI with $m = 0$.*

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

Definition 4.4.4. *A (S,T,δ)-aDPI is a decision-version adaptive permutation inverter* $\mathsf{aD} = (\mathsf{aD}_0, \mathsf{aD}_1)$ *satisfying all of the following:*

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

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

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

Definition 4.4.5. *A decision-version permutation inverter (**DPI**)* $\mathsf{D} = (\mathsf{D}_0,\mathsf{D}_1)$ *is defined as an **aDPI** with* $m=0$*. An* (S,T,δ)*-**DPI** is an* (S,T,δ)*-**aDPI** with* $m=0$*.*

4.5 Amplification

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

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

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

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

   (b) *uses* $r_0,...,r_{\ell-1}$ *to generate* ℓ *permutation pairs* $\{\sigma_{1,i},\sigma_{2,i}\}_{i=0}^{\ell-1}$ *in* $\mathcal{P}_N$*, and then runs* $\mathsf{S}_0(\sigma_{1,i}\circ\pi\circ\sigma_{2,i}, r_{i+\ell})$ *to get a quantum state* $\rho_i := \rho_{\sigma_{1,i}\circ\pi\circ\sigma_{2,i},r_{i+\ell}}$ *for all* $i \in [0,\ell-1]$*. Finally,* $\mathsf{S}[\ell]_0$ *outputs the quantum state* $\bigotimes_{i=0}^{\ell-1}\rho_i$*.*

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

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

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

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

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

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

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

[a] How to construct this quantum oracle is described in Section 4.9.1.

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

Lemma 4.5.1 (Amplification, search)**.** *Let* S *be a* (S, T, ϵ)*-SPI for some* $\epsilon > 0$*. Then* $\mathsf{S}[\ell]$ *is a* $(\ell S, \ell(T+1), 1-(1-\epsilon)^\ell)$*-SPI.*

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

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

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

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

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

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

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

Lemma 4.5.2. *Let* S *be a* (S, T, ϵ)*-**SPI** for some* $\epsilon > 0$*. Then, we can construct a **SPI*** $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ *using* $S(\mathsf{S}[\ell])$ *qubits of advice and making* $T(\mathsf{S}[\ell])$ *queries, with*

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

such that

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

The proof is analogous to Lemma 4.5.1 and is given in Section 4.9.2.

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

Lemma 4.5.3 (Amplification, decision)**.** *Let* D *be a* (S, T, δ)*-**DPI** for some* $\delta > 0$*. Then* $\mathsf{D}[\ell]$ *is a* $(\ell S, \ell T, 1/2 - \exp(-\delta^2/(1+2\delta) \cdot \ell))$*-**DPI**.*

The proof is analogous to the search version and given in Section 4.9.3.

4.6 A search-to-decision reduction

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

Theorem 4.6.1. *Let* D *be a* (S, T, δ)-*DPI. Then for any* $\ell \in \mathbb{N}$*, we can construct a* $(n\ell S, n\ell T, \eta)$-**SPI** *with*

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

Proof. Given an δ-DPI $(\mathsf{D}_0, \mathsf{D}_1)$ with storage size S and query size T, we can construct a η'-DPI $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ with storage size ℓS and query size ℓT through ℓ-time repetition. By Lemma 4.5.3, we have that $\eta' \geq \frac{1}{2} - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right)$. Note that the algorithm $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ runs $(\mathsf{D}_0, \mathsf{D}_1)$ as a subroutine. In the following, we represent elements in $[N]$ using a binary decomposition of length $\lceil \log N \rceil$. To state our search-to-decision reduction, we introduce a generalized swap operation, denoted by $\mathsf{swap}_{a,b}$, which acts as follows for any quantum state of m qubits:

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

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

1. The algorithm S_0 proceeds as follows:

   (a) S_0 receives a random permutation $\pi : [N] \to [N]$ and a random string $r \leftarrow \{0,1\}^*$ as inputs. We parse r into $\lceil \log N \rceil$ individual substrings, i.e. $r = r_0 \| ... \| r_{\lceil \log N \rceil - 1}$; the length of each substring is clear in context.

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

2. The oracle algorithm $\mathsf{S}_1^{\mathcal{O}_\pi, \mathcal{O}_{\pi^{-1}_{\perp y}}}$ proceeds as follows:[1]

   (a) S_1 receives $\bigotimes_{j=0}^{n-1} \rho_j$, a random string $r := r_0 \| ... \| r_{n-1}$ and an image $y \in [N]$.

   (b) S_1 then runs the following routine for each $j \in [0, \lceil \log N \rceil - 1]$:

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

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

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

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

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

We now argue that the probability that $\mathcal{D}[\ell]_1$ correctly recovers the pre-image bits b_i and b_j is independent for each $i \neq j$. From Lemma 4.5.3, we know that $\mathsf{D}[\ell]_1$ runs D_1 as a subroutine, i.e. it decides the first bit of the pre-image of y by running D_1 (in Lemma 4.5.3) ℓ times with different random coins. It actually needs to recall D_1 for amplification and for each iteration in this amplification $k \in [0, \ell - 1]$, where the actual modified permutation under use is $\sigma_{i,k} \circ \pi \circ \mathsf{swap}_{0,i}$ and image is $\sigma_{i,k}(y)$. Similarly for term j, $\sigma_{j,k} \circ \pi \circ \mathsf{swap}_{0,j}$ and $\sigma_{j,k}(y)$ is used as the permutation and image. Since the random coins (r_i and r_j), which are used to modify the target permutation π, are independently random, those random permutations ($\sigma_{i,k}$ and $\sigma_{j,k}$) generated from random coins are independently random and so do those modified composition permutations, images, and advice states.

[1] Here, we borrow the notation for $\mathcal{O}_\pi$ and $\mathcal{O}_{\pi^{-1}_{\perp y}}$ from the experiment described in Section 4.4.

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

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

where the last line follows from Bernoulli's inequality. Finally, we compute the resources needed for $(\mathsf{S}_0, \mathsf{S}_1)$. By Lemma 4.5.3, $(\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$ requires space ℓS and query size ℓT. For $j \in [0, \lceil \log N \rceil - 1]$, S_0 stores $\mathsf{D}[\ell]_0$'s outputs and thus S requires storage size $\lceil \log N \rceil \ell S$. Similarly, S_1 runs $\mathsf{D}[\ell]_1$ to obtain b_j and thus it requires $\lceil \log N \rceil \ell T$ queries in total. □

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

4.7 Lower bounds

4.7.1 Search version

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

Theorem 4.7.1. *Let $N \in \mathbb{N}$. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be a $(S, T, 2\epsilon/3)$-SPI that satisfies*

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

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

Proof. To prove the claim, we construct a QRAC-VL scheme that encodes the function π^{-1} and then derive the desired space-time trade-off via Corollary 4.3.6. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an $2\epsilon/3$-SPI that succeeds on a ϵ-fraction of inputs with probability at least $2/3$. In other words, S satisfies

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

By the averaging argument in Lemma 4.3.3 with parameter $\theta = 1/2$, it follows that there exists a large subset $\mathcal{X} \subseteq \mathcal{S}_N$ of permutations with size at least $N!/2$ such that for any permutation $\pi \in \mathcal{X}$, we have that

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

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

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

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

$$\Sigma_0^{\mathcal{R}} = \mathcal{R} \setminus \{x\} \times \{0\}$$

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

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

[2]The (reversible) quantum oracle implementation is similar to the one in Definition 4.4.3. We use the function $\pi_{\perp y}$ for ease of presentation since the same proof carries over with minor modifications in the quantum oracle case.

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

$$x \in \mathcal{R}; \tag{4.3}$$

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

$$q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \;\leq\; c/T(\mathsf{S}). \tag{4.4}$$

Claim 1. Let $\mathcal{G} \subseteq [N]$ be the set of x which satisfy the conditions in Equation 4.3 and Equation 4.4. Then, there exist constants $\gamma, c \in (0, 1)$ such that

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

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

Proof. (of the claim) Let $\mathcal{H} = \mathcal{R} \cap \mathcal{I}$ denote the set of $x \in \mathcal{R}$ for which S correctly inverts $\pi(x)$ with probability at least $2/3$ over the choice of r. By the definition of the set $\mathcal{R}$, it follows that $|\mathcal{H}|$ has a binomial distribution. Therefore, in expectation, we have that $|\mathcal{H}| = \gamma|\mathcal{I}|/T(\mathsf{S})^2$. Using the multiplicative Chernoff bound in Lemma 4.3.1 and the fact that $T(\mathsf{S}) = o(\epsilon\sqrt{N})$, we get

$$\Pr_{\mathcal{R}}\left[|\mathcal{H}| \geq \frac{\gamma|\mathcal{I}|}{2\,T(\mathsf{S})^2}\right] \geq 0.9, \tag{4.5}$$

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

$$q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0,1\}) \leq T(\mathsf{S}). \tag{4.6}$$

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

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}})\big] &= \mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}}) + q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_1^{\mathcal{R}})\big] && (\Sigma_0^{\mathcal{R}}, \Sigma_1^{\mathcal{R}} \text{ are disjoint})\\
&= \mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}})\big] + \mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_1^{\mathcal{R}})\big] && (\text{linearity of expectation})\\
&= \mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \mathcal{R} \setminus \{x\} \times \{0\})\big] + \mathop{\mathbb{E}}_{\mathcal{R}}\big[q(\mathsf{S}_1^{\pi_{\perp y}}, \pi(\mathcal{R}) \setminus \{\pi(x)\} \times \{1\})\big]\\
&= \frac{\gamma}{T(\mathsf{S})^2} q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{x\} \times \{0\}) + \frac{\gamma}{T(\mathsf{S})^2} q(\mathsf{S}_1^{\pi_{\perp y}}, \pi([N]) \setminus \{\pi(x)\} \times \{1\})\\
&= \frac{\gamma}{T(\mathsf{S})^2} q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{x\} \times \{0\}) + \frac{\gamma}{T(\mathsf{S})^2} q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \setminus \{\pi(x)\} \times \{1\})\\
&&& (\pi \text{ is a permutation})\\
&\leq \frac{\gamma}{T(\mathsf{S})^2} \Big(q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0\}) + q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{1\}) \Big) && (\text{supersets})\\
&= \frac{\gamma}{T(\mathsf{S})^2} q(\mathsf{S}_1^{\pi_{\perp y}}, [N] \times \{0,1\}) \leq \frac{\gamma}{T(\mathsf{S})} && (\text{by Equation 4.6})
\end{aligned}$$

Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}}\left[q(\mathsf{S}_1^{\pi_{\perp y}}, \Sigma_0^{\mathcal{R}} \cup \Sigma_1^{\mathcal{R}}) \geq \frac{c}{T(\mathsf{S})}\right] \leq \frac{T(\mathsf{S})}{c} \cdot \frac{\gamma}{T(\mathsf{S})} = \frac{\gamma}{c}. \tag{4.7}$$

Let us now denote by $\mathcal{J}$ the subset of $x \in \mathcal{I}$ that satisfy Equation 4.3 but not Equation 4.4. Note that Equation 4.3 and Equation 4.4 are independent for each $x \in \mathcal{I}$, since Equation 4.3 is about whether $x \in \mathcal{R}$ and Equation 4.4 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 Equation 4.7, the probability that $x \in \mathcal{I}$ satisfies $x \in \mathcal{J}$ is at most

$\gamma^2/(cT(\mathsf{S})^2)$. Hence, by Markov's inequality,

$$\Pr_{\mathcal{R}}\left[|\mathcal{J}| \leq \frac{10|\mathcal{I}|\gamma^2}{cT(\mathsf{S})^2}\right] \geq 0.9. \tag{4.8}$$

Using Equation 4.5 and Equation 4.8, we get with probability at least 0.8 over the the choice of $\mathcal{R}$,

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

given that γ is a sufficiently small positive constant. □

Protocol 2 (Quantum Random Access Code For Inverting Permutations)**.**

Let $c, \gamma \in (0,1)$ be parameters. Consider the following (variable-length) quantum random-access code given by $\mathsf{QRAC\text{-}VL} = (\mathsf{Enc}, \mathsf{Dec})$ defined as follows:

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

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

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

  1. *The size of $\mathcal{G}$, encoded using $\log N$ bits;*
  2. *the set $\mathcal{G} \subseteq \mathcal{R}$, encoded using $\log\binom{|\mathcal{R}|}{|\mathcal{G}|}$ bits;*
  3. *The permutation π restricted to inputs outside of $\mathcal{G}$, encoded using $\log(N!/|\mathcal{G}|!)$ bits;*

4. *Quantum advice used by the algorithm repeated ρ times with $\alpha^{\otimes \rho}$, for $\alpha \leftarrow \mathsf{S}_0(\pi, r)$ for some ρ that we will decide later. (We can compute this as the encoder can pre-process multiple copies of the same advice. Note that this is the only part of our encoding that is not classical.)*

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

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

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

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

[a] The (reversible) quantum oracle implementation for $\bar{\pi}_{\perp y}$ is provided in Section 4.9.4.

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

Lemma 4.3.4 and the properties of the total query magnitude:

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

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

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

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

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

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

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

and we can rewrite the size of the encoding as

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

In the case when the decoder is queried on an input that is already known, that is $y \notin \pi(\mathcal{G})$ (which occurs with probability $1-|\mathcal{G}|/N$), the decoder recovers the correct pre-image with probability 1. Otherwise, the analysis is the following: with just one copy of the advice, the decoder recovers the correct pre-image with probability $2/3$, and hence with ρ many copies, the decoder can take the majority vote and recover the correct pre-image with probability $1-\exp(-\Omega(\rho))$. The latter follows from the Chernoff bound in Lemma 4.3.1. Overall, the average encoding length is

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

where the average success probability is $1-|\mathcal{G}|/N\cdot\exp(-\Omega(\rho))$. By setting $\rho = \Omega(\log(N/\epsilon)) = \Omega(\log N)$, the average success probability amounts to $1-O(1/N^2)$. Therefore, using the lower bound in Corollary 4.3.6, we have

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

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

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

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

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

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

We remark that the search inverter we consider in Theorem 4.7.1 succeeds on more than just a constant number of inputs, that is $\epsilon = \omega(1/N)$, and beats the time complexity of $T = \Omega(\sqrt{\epsilon N})$ which is required for unstructured search using Grover's algorithm. [Gro96, DH08, Zha19]. Next, we remove the restriction on the inverter by applying amplification (specifically, Corollary 4.5.2.) This yields a lower bound in the full average-case version of the search inversion problem.

Theorem 4.7.2. *Let* S *be a* (S, T, ϵ)*-*SPI *for some* $\epsilon > 0$*. Suppose that* $\epsilon = \omega(1/N)$*,* $T = o(\epsilon^2\sqrt{N})$*, and* $S \geq 1$*. Then, for sufficiently large* N *we have*

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

Proof. Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an (S, T, ϵ)-SPI, for some $\epsilon > 0$. Using Corollary 4.5.2, we can construct an SPI $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ with space and time complexities

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

such that

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

From Theorem 4.7.1 it follows that for sufficiently large $N \geq 1$,

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

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

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

the trade-off between space and time complexities is

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

□

Note that we incur a loss (ϵ^3 versus ϵ) in our search lower bound due to the fact that we need to amplify the *restricted* search inverter in Theorem 4.7.1. 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].

4.7.2 Decision version

The search lower bound of Theorem 4.7.2, when combined with the search-to-decision reduction of Theorem 4.6.1, yields a lower bound for the decision version.

Corollary 4.7.3. *Let* D *be a* (S, T, δ)*-DPI for some* $\delta > 0$*. Suppose that* $\delta = \omega(1/N)$ *and* $T = \tilde{o}\left(\delta^2 \sqrt{N}\right)$ *and* $S \geq 1$*. Then, for sufficiently large* N *we have*

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

Proof. Let $N = 2^n$. Given a $(S(\mathsf{D}), T(\mathsf{D}), \delta)$-$\mathsf{DPI} = (\mathsf{D}_0, \mathsf{D}_1)$ where D_0 outputs S-qubit state and D_1 makes T queries, one can construct an $(S(\mathsf{S}), T(\mathsf{S}), \eta)$-$\mathsf{SPI} = (\mathsf{S}_0, \mathsf{S}_1)$ by Theorem 4.6.1 with $\eta \geq 1 - \mathsf{negl}(n)$, and with space and time complexities

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

where $\ell = \Omega\left(\frac{n(1+2\delta)}{\delta^2}\right)$. It directly follows from Theorem 4.7.2 that with conditions

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

S satisfies the space-time trade-off lower bound

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

for sufficiently large N. □

Similar to the search lower bound from before, we incur a loss that amounts to a factor δ^6. This results from our specific approach which is based on the search-to-decision reduction in Theorem 4.6.1. We believe that our lower bound could potentially be improved even further.

4.8 Applications

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

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

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

Consider the following security definition.

Definition 4.8.1. *(*$\mathsf{OW\text{-}QCCRA2}$*) Let* $\mathsf{SKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a private-key encryption scheme. We say that* SKE *is* $\mathsf{OW\text{-}QCCRA2}$ *if the advantage for any quantum polynomial-time adversary* $\mathcal{A}$ *in the following* $\mathsf{OW\text{-}QCCRA2}$ *experiment is at most negligible:*

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

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

Next, we define two simple encryption schemes.

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

- Gen is given $1^{n_1+n_2}$ and outputs a description k of a uniformly random permutation π on $\{0,1\}^{n_1+n_2}$;
- Enc is given k, $m \in \{0,1\}^{n_1}$ and $r \in \{0,1\}^{n_2}$, and outputs $c := \pi(m||r)$;
- Dec is given k and $c \in \{0,1\}^{n_1+n_2}$, and outputs the first n bits of $\pi^{-1}(c)$.

Definition 4.8.2. *(ϵ-Qsecure PRP) [KL20, Zha16] Let $P_k : \{0,1\}^\lambda \times \{0,1\}^n \to \{0,1\}^n$ be a permutation family. We call P_k a ϵ-Qsecure PRP if for any efficient quantum adversary $\mathcal{A}$ who makes q quantum queries, there exist a negligible function $\epsilon(\lambda)$ such that*

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

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

PRP Scheme. Let $\{P_k : \{0,1\}^{n_1+n_2} \mapsto \{0,1\}^{n_1+n_2}\}$ be a family of ϵ-Qsecure PRPs and consider the following scheme:

- Gen takes as input a security parameter $1^{n_1+n_2}$ and returns a key $k \in \{0,1\}^{n_1+n_2}$ for P_k;
- Enc is given key $k \in \{0,1\}^{n_1+n_2}$, $m \in \{0,1\}^{n_1}$ and $r \in \{0,1\}^{n_2}$, and outputs $c := P_k(m||r)$;
- Dec is given key $k \in \{0,1\}^{n_1+n_2}$ and $c \in \{0,1\}^{n_1+n_2}$, and outputs the first n_1 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-QCCRA2. We are able to prove the following theorems.

Theorem 4.8.3. *The* PRP *scheme is* OW-QCCRA2*. In other words, for any quantum adversary $\mathcal{A}$ who makes $t(n)$ quantum queries in the pre-challenge phase and $\ell(n)$ quantum queries in the post-*

challenge phase, it holds that

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

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

Proof. Given an adversary $\mathcal{A}$ that attacks the RP scheme in the $\mathsf{OW\text{-}QCCRA2}$ experiment described in Definition 4.8.1, we can construct a (S, T, δ)-DPI $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$ in the decision inversion experiment, which takes place as follows:

1. **(sample instance and coins)** a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is sampled;

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

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

4. **(check)** If $b' = b$, output 1; otherwise output 0.

It trivially follows that

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

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

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

Therefore

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

Where $\delta\le O(\frac{\ell^2 S}{2^{2n}})^{\frac{1}{6}}$ by Corollary 4.7.3, and by Definition 4.8.2 ϵ is negligible. Remark that the above bound becomes $\frac{1}{2}+\mathsf{negl}(n)$ when $\mathcal{A}$ is a quantum polynomial time (QPT) adversary since both δ and ϵT are negligible when t and ℓ are of polynomial size. $\square$

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

4.9 Additional proofs and constructions

4.9.1 Quantum oracle construction in Protocol 1

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

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

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

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

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

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

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

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

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

4.9.2 Another amplification lemma proof

Lemma 4.5.2. *Let $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ be an ϵ-**SPI** with space and time complexity given by $S(\mathsf{S})$ and $T(\mathsf{S})$, respectively, for some $\epsilon > 0$. Then, we can construct an **SPI** $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$ with space and time* *complexities*

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

such that

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

Proof. Let $\ell = \left\lceil \frac{\ln(10)}{\epsilon} \right\rceil$. Using Lemma 4.5.1, we can construct an ℓ-time repetition of S (η)-SPI, denoted by $\mathsf{S}[\ell] = (\mathsf{S}[\ell]_0, \mathsf{S}[\ell]_1)$, with $\eta = 1-(1-\epsilon)^\ell$ and space and time complexities $S(\mathsf{S}[\ell]) = \ell \cdot S(\mathsf{S})$ and $T(\mathsf{S}[\ell]) = \ell \cdot (T(\mathsf{S}) + 1)$. In other words,

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

Let $\mathcal{S}_N$ denote the set of permutations over $[N]$. From Lemma 4.3.3 it follows that there exists $\theta = 7/9$ and a subset $\mathcal{X}_\theta \subseteq \mathcal{S}_N \times [N]$ of size at least

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

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

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

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

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

This proves the claim. $\square$

4.9.3 Decision amplification proof

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

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

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

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

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

$$\sigma_{2,i}(x_1, ..., x_n) = (x_1 \oplus r_i^*, x_2, ..., x_n), \tag{4.9}$$

   where r_i^* is some random bit generated from r_i for all $i \in [0, \ell-1]$. Then runs $\mathsf{D}_0(\sigma_{1,i} \circ \pi \circ \sigma_{2,i}, r_{i+\ell})$ to get a quantum state $\rho_i := \rho_{\sigma_{1,i}\circ\pi\circ\sigma_{2,i}, r_{i+\ell}}$ for all $i \in [0, \ell-1]$. Finally,

$\mathsf{D}[\ell]_0$ *outputs a quantum state* $\bigotimes_{i=0}^{\ell-1} \rho_i$.

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

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

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

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

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

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

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

Lemma 4.5.3. *Let* $(\mathsf{D}_0, \mathsf{D}_1)$ *be a* δ*-DPI, where* D_0 *outputs an* S*-qubit state and* D_1 *makes* T *queries. Then, we can construct an* ℓ*-time repetition of* D*, denoted by* $\mathsf{D}[\ell] = (\mathsf{D}[\ell]_0, \mathsf{D}[\ell]_1)$*, which is an* η*-DPI for* $\eta \geq \frac{1}{2} - \exp\left(-\frac{\delta^2}{(1+2\delta)} \cdot \ell\right)$*, and has space and time complexities given by*

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

Proof. Let $(\mathsf{D}_0, \mathsf{D}_1)$ be a δ-DPI for some $\delta > 0$, where D_0 outputs an S-qubit state and D_1 makes T queries. Similarly as in Lemma 4.5.1, we consider the execution of the ℓ-time repetition of δ-DPI, denoted by DPI $\mathsf{D}[\ell]$, which we define in Protocol 3. For each iteration $i \in [0, \ell-1]$, we have

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

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

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

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

Note that the last equality holds because of Equation 4.9. We then define another event

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

Given that B happens, we have $\{b_j\}_{j\in K}$ such that for all $j \in K$, $\big(\sigma_{2,j} \circ (b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$. We now consider the probability that A happens. In Equation 4.9, since all r_i^* are independently randomly generated, the value of $\big(\sigma_{2,i} \circ (b_i||0^{n-1})\big)|_0$ is independent of all other values of $\big(\sigma_{2,j} \circ (b_i||0^{n-1})\big)|_0$. Therefore, the event that $\big(\sigma_{2,i} \circ (b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$ is not correlated with all other $\big(\sigma_{2,j} \circ (b_i||0^{n-1})\big)|_0 \neq \pi^{-1}(y)|_0$, i.e., $\Pr[A|B] = \Pr[A]$. This is true for any i and K. Same as the search case, in each trial, the base inverter is solving a completely independent permutation inversion problem, thus we conclude that all ℓ trails are mutually independent.

Let $X = \sum_{i=0}^{\ell-1} X_i$, we have that $\mathbb{E}[X] \geq \ell \cdot (\frac{1}{2} + \delta)$ by the linearity of expectation. Note that $\mathsf{D}[\ell]$ succeeds in $\mathsf{DecisionInvert}$ if and only if $\mathsf{D}[\ell]_1$ can output $b^* = \pi^{-1}(y)|_0$, i.e. $X > \frac{\ell}{2}$ in which case more than half of the elements in $\{b_0, ..., b_{\ell-1}\}$ are equal to $\pi^{-1}(y)|_0$. By the multiplicative Chernoff bound in Lemma 4.3.1, the probability that $\mathsf{DecisionInvert}$ fails is at most

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

Note that the resource requirements needed for the amplification procedure amount to space and time complexities ℓS and ℓT, respectively, similar as in Lemma 4.5.1. $\square$

4.9.4 Quantum oracle construction in Protocol 2

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

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

and

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

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

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

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

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

Chapter 5: Multiangle QAOA Does Not Always Need All Its Angles

Beyond cryptography, quantum algorithms are also widely studied as potential tools for solving optimization problems. Unlike in symmetric-key cryptography and permutation inversion, quantum algorithms in this setting may offer more tangible advantages, although these advantages remain limited in practice.

One prominent example is QAOA, a widely studied hybrid classical-quantum algorithm for combinatorial optimization. However, its provable approximation guarantees are still much lower than those of classical polynomial-time approximation algorithms.

For the benchmark NP-hard problem Max-Cut, the classical Goemans-Williamson SDP algorithm [GW95] achieves an approximation ratio of about 0.87856 on general nonnegative-weight graphs, while for graphs of maximum degree 3, an improved SDP-based algorithm achieves a ratio of 0.9326 [HLZ04]. By contrast, the known provable guarantees for QAOA are mainly restricted to special graph families. For example, the original QAOA paper [FGG14] showed that, for $p = 1$ on 3-regular graphs, QAOA achieves an approximation ratio of at least 0.6924. Later work [WL21] established a guarantee of 0.7559 for $p = 2$ on 3-regular graphs. No worst-case polynomial-time approximation guarantee is currently known for QAOA on general graphs.

These limitations have motivated the study of QAOA variants with expanded parameter spaces, in the hope of improving empirical performance and, potentially, approximation quality. Multi-angle

QAOA (ma-QAOA) is one such variant, and numerical studies suggest that it can outperform standard QAOA [HTO+21].

This chapter explores a structural extension of ma-QAOA and demonstrates, through numerical evidence, that optimization complexity can often be reduced with little or no loss in solution quality, although the extent and generality of such reductions are inherently limited. At the same time, even for these variants, approximation-ratio guarantees remain largely unexplored, and no larger provable guarantees are currently known.

5.1 Summary of contribution

Introducing additional tunable parameters to quantum circuits is a powerful way of improving performance without increasing hardware requirements. A previously introduced multiangle extension of the quantum approximate optimization algorithm (ma-QAOA) significantly improves the solution quality compared with QAOA by allowing the parameters for each Hamiltonian term to vary independently. However, this flexibility comes at the cost of a substantially enlarged parameter space, and prior work suggests that many of these parameters are redundant.

In this work, we show that problem symmetries can be systematically exploited to reduce the number of variational parameters without sacrificing performance. Focusing on Max-Cut, we perform a comprehensive numerical study over all 7,565 connected, non-isomorphic 8-node graphs with non-trivial symmetry groups. We find that symmetry-based parameter tying often achieves near-optimal performance while substantially reducing the number of parameters, and that in some cases enforcing only the largest symmetry of the instance already yields most of the achievable reduction.

Even when some performance degradation occurs, symmetry guided reduction remains effective:

large reductions in parameter count can be obtained with only modest loss in objective value. In contrast, we show that random parameter reduction strategies lead to significantly worse performance, indicating the nontrivial role of symmetry.

Overall, our results provide numerical evidence that exploiting problem symmetries can substantially reduce the optimization complexity of ma-QAOA while preserving most of its performance benefits.

5.2 Background

Quantum hardware has improved rapidly in recent years [AAB+19,WBC+21,MLA+22,RMP+22], opening up the possibility of demonstrating quantum advantage on a relevant practical problem. Combinatorial optimization problems are commonly considered targets for near-term quantum devices [MW13, MMB22], with the quantum approximate optimization algorithm (QAOA) [HP00,FGG14] as a promising candidate algorithm because of its low hardware resource requirements [HSN+21,KLGS22,NSY+22, SG21].

QAOA solves optimization problems using a parameterized circuit composed of layers of alternating operators, with two operators being evolutions with a Hamiltonian encoding the objective function and a problem-instance-independent mixer Hamiltonian. The evolution times are free parameters (often called angles), which are optimized with the goal of maximizing the expected quality of the measurement outcomes.

The success of variational quantum algorithms with a large number of trainable parameters such as quantum neural networks and the variational quantum eigensolver [CAB+21] motivated the introduction of additional parameters in QAOA. Intuitively, adding additional parameters to the algorithm based on the structure of the problem can only increase the circuit expressiveness and thereby can only improve the algorithm's performance.

Multiangle QAOA (ma-QAOA) is a modification of QAOA that incorporates additional parameters [HLO+22] by allowing the parameter associated with each term in the problem and mixer Hamiltonian to vary independently. ma-QAOA has been shown to solve Max-Cut on star graphs exactly using only one layer, whereas QAOA achieves an approximation ratio of only 0.75. The improvement in the quality of the solution achieved by the introduction of the parameters is modest, however, suggesting that the large number of parameters does not translate to a highly expressive circuit. Moreover, preliminary ma-QAOA research has shown that parameters tend to cluster around multiples of 0.25π [Her22]. Together, these observations suggest that the number of parameters in ma-QAOA can be reduced without affecting the solution quality.

In this work we demonstrate the connection between the redundancy in ma-QAOA parameters and the problem symmetries. Specifically, we reduce the number of parameters by setting the parameters connected by a chosen symmetry to be equal. Using Max-Cut as an example, we demonstrate numerically that exploiting problem symmetries frequently allows substantial parameter reduction with little or no loss in performance.

Inspired by this observation, we propose a modification of ma-QAOA that uses the full symmetry group (max-sym-QAOA). The full symmetry group can be obtained efficiently for many classes of graphs, and fast heuristic solvers can be used in practice [SHHS21].

5.2.1 Organization

This work is organized as follows. First, we introduce binary optimization, QAOA, and graph symmetry background material in Section 5.3. In Section 5.4 we then discuss the methods used in this work. In Section 5.5 we discuss our results, and we conclude with a discussion in Section 5.6.

5.3 Technical preliminaries

5.3.1 Binary optimization problem

We consider binary optimization(BO) problems of the form $\max_{x\in\{0,1\}^n} f(x)$, where $f(x)$ is a non-negative objective function over the Boolean cube $\{0,1\}^n$. It is often a sum of other functions that describe the system, called clauses.

When solving BO problems on quantum hardware, we construct a cost Hamiltonian H_c that encodes $f(x)$, so that $H_c\left|x\right\rangle = f(x)\left|x\right\rangle$. Then the optimization problem becomes

$$\max_{x\in\{0,1\}^n} \left\langle x\right| H_c \left|x\right\rangle .$$

The outcome of the algorithm is marked as x^*, and algorithm performance is typically quantified by the approximation ratio $r \in [0,1]$ given by

$$r := \frac{f(x^*)}{\max f(x)} = \frac{\left\langle x^*\right| H_c \left|x^*\right\rangle}{\max \left\langle x\right| H_c \left|x\right\rangle}. \tag{5.1}$$

5.3.2 QAOA

QAOA is a hybrid quantum-classical algorithm that finds approximate solutions to combinatorial optimization problems [FGG14]. To solve a given optimization problem with QAOA, one must construct a cost Hamiltonian H_c that encodes the objective function and a mixer Hamiltonian H_m. Let $U(\gamma, C) = e^{-iH_c\gamma}$ and $U(\beta, B) = e^{-iH_m\beta}$, where γ and β are free parameters. These two unitaries are applied to an initial state $|s\rangle$, which is an eigenvector of H_m. The outcome of p iterations of the algorithm is denoted $|\gamma, \beta\rangle_p$ and is

$$\begin{aligned}|\gamma, \beta\rangle_p =& U(\beta_p, H_m)U(\gamma_p, H_c)\ldots \\ & U(\beta_1, H_m)U(\gamma_1, H_c)\,|s\rangle\,.\end{aligned}$$

The parameters γ and β are chosen to maximize

$$E(\gamma, \beta) = \langle\gamma, \beta|\, H_c \,|\gamma, \beta\rangle\,.$$

Measuring the state $|\gamma, \beta\rangle$ gives an approximate solution to the BO problem encoded by H_c.

ma-QAOA is similar to QAOA; however, the definitions of $U(\gamma, C)$ and $U(\beta, B)$ are changed to

$$U(\gamma, C) = e^{-i\sum_a C_a\gamma_a} \qquad \text{and} \qquad U(\beta, B) = e^{-i\sum_b X_b\beta_b},$$

where C_a is a clause in the objective function and X_b is the Pauli-x operator acting on qubit b. Throughout this work, r_{y} refers to the approximation of y-QAOA, where y is a variation of QAOA.

5.3.3 QAOA on Max-Cut problem

The Max-Cut problem is well studied in QAOA literature (e.g. [HSN+21, SA19]) and is thus a natural problem to consider when studying QAOA variants. Given a simple graph $G = (V, E)$, the Max-Cut problem aims to partition V into two disjoint sets so that the number of edges with endpoints in both sets is maximized. This problem is NP-hard to solve exactly.

When solving the Max-Cut problem using QAOA, the cost Hamiltonian is

$$H_c = \sum_{uv \in E} \frac{1}{2}(I - Z_u Z_v),$$

and the mixer Hamiltonian is typically

$$H_m = \sum_{v \in V} X_v,$$

where Z_v and X_v are single-qubit Pauli operators acting on qubit v. Each layer of the QAOA circuit has one γ for all edges and one β for all vertices. The number of tunable parameters is just $2p$, independent of the graph size.

In ma-QAOA, each vertex and each edge has its own angle. Thus, there are $(|E| + |V|) \cdot p$ parameters to optimize in this modified algorithm. One drawback to this approach is that finding $(|E| + |V|) \cdot p$ parameters can be difficult, especially as the size of the problems grows. However, the algorithm has better performance than QAOA has on average [HLO+22].

5.3.4 Graph symmetries

A graph automorphism is a permutation of the vertex set of a graph, $\sigma : V \to V$, that satisfies the condition that a pair of vertices, (u, v), forms an edge if and only if the pair $(\sigma(u), \sigma(v))$ also forms an edge. Automorphisms can be represented by products of disjoint cycles, $\sigma = \pi_1 \dots \pi_k$, where $\pi_l = (i_1, i_2, \dots, i_j)$ and each entry in the cycle is a unique integer. In this notation, $\sigma(i_a) = i_{a+1}$ modulo j. Any set of automorphisms can generate a corresponding vertex and edge orbit, which are the equivalence classes of the vertices (edges) of a graph G under the action of the automorphism.

A generator of a group is a set of automorphisms $(\sigma_1, \dots, \sigma_n)$ containing group elements so that (possibly, repeated) application of the generators on themselves and each other can produce all elements in the group. In this work, we call the group generator of the automorphism group of the graph G the symmetry generator, and it can generate corresponding vertex and edge orbits. We denote the vertex (edge) orbit of the symmetry generator as the *maximum vertex (edge) orbit*, written as $\mathcal{O}_v(\mathcal{O}_e)$.

5.4 Methods

We use the symmetry structure of the problem to reduce the number of parameters in ma-QAOA. Symmetry is known to impact QAOA performance [SHHS21]. To analyze the role of symmetries, we introduce three modifications called best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA.

sym-QAOA selects a single automorphism of the target graph and assigns the same angle to all vertices in the same vertex orbit and the same angle to all edges in the same edge orbit. It then optimizes the QAOA parameters and executes the resulting QAOA circuit. best-1sym-QAOA runs sym-QAOA over all automorphisms of G and selects the one that gives the largest Max-Cut value.

Algorithm 1: max-sym-QAOA

Input : Graph G, number of layers p.
Output: Optimized $\{\beta, \gamma\}$ approx. max-cut $\tilde{A}$

Construct cost Hamiltonian H_c from G
Find the symmetry generator of G and corresponding maximum vertex/edge orbit sets $\mathcal{O}_v$, $\mathcal{O}_e$
Sample $|\mathcal{O}_v| + |\mathcal{O}_e|$ initial parameters $\{\beta, \gamma\} = \{\beta_0, \ldots, \beta_{|\mathcal{O}_v-1|}, \gamma_0, \ldots, \gamma_{|\mathcal{O}_e-1|}\}$. Fix vertices/edges in the same orbit to have the same value $|\psi(\beta, \gamma)\rangle \leftarrow$ QAOAcirc$(\{\beta, \gamma\}, p)$
$E(\beta, \gamma) \leftarrow \langle \psi(\beta, \gamma)| H_c |\psi(\beta, \gamma)\rangle$
$\{\beta, \gamma\}, x^* \leftarrow$ classical optimization algorithms to optimize $E(\beta, \gamma)$

When using max-sym-QAOA to solve Max-Cut on a graph G, the first step is to compute the symmetry generator of G and determine the corresponding maximum vertex orbit, $\mathcal{O}_v$, and edge orbit, $\mathcal{O}_e$. The algorithm requires $|\mathcal{O}_v|+|\mathcal{O}_e|$ parameters, where each element in the same vertex orbit or edge orbit receives the same parameter. The algorithm samples $|\mathcal{O}_v| + |\mathcal{O}_e|$ parameters randomly as initial parameters and runs the QAOA variational quantum circuit as a subroutine using those parameters. QAOA optimization steps are applied until the solution converges to the optimal solution or until the desired number of iterations has been performed. The formal algorithm is described in Algorithm 1.

Finding the generating set of the automorphism group of a graph is an extra step in max-sym-QAOA compared with ma-QAOA. The time complexity of this step is at most quasi-polynomial since its polynomial time equivalent graph isomorphism problem can be solved by a quasi-polynomial algorithm [Bab16]. Also, many polynomial time heuristics exist for specific classes of graphs such as **nauty** [McK07, MP13] used in this work.

rand-group-QAOA groups vertices in the problem graph randomly into $|\mathcal{O}_v|$ sets and edges randomly into $|\mathcal{O}_e|$ sets, so that the number of parameters is the same as that of max-sym-QAOA.

The generator used in max-sym-QAOA is usually a set of automorphisms, so max-sym-QAOA is not always contained in best-1sym-QAOA, which ranges over all single automorphisms. Thus, max-sym-QAOA may perform better than best-1sym-QAOA.

5.5 Results

In this work we implement one iteration of max-sym-QAOA, rand-group-QAOA, and best-1sym-QAOA, using the graph descriptions from [McK]. We then compare the algorithms with one another and ma-QAOA using the data found in [LH]. We use COBYLA to optimize the QAOA parameters, although we expect similar results to be obtained with other gradient-free and gradient-based local methods.

5.5.1 best-1sym-QAOA vs. ma-QAOA

Among all $7,565$ graphs with nontrivial symmetries, best-1sym-QAOA has fewer parameters than ma-QAOA has on $5,918$ graphs. Thus, we analyze only best-1sym-QAOA on those graphs. Figure 5.1 shows the difference in approximation ratios between best-1sym-QAOA and ma-QAOA for these $5,918$ graphs. best-1sym-QAOA has the same approximation ratio as ma-QAOA has on $5,097$ graphs, which is approximately 86.1% of the studied graphs, while using on average 28.1% fewer parameters than ma-QAOA uses.

We also quantify the ratio of the difference in the Max-Cut (approximation ratio) values of ma-QAOA and best-1sym-QAOA to the difference in the Max-Cut (approximation ratio) values of ma-QAOA and QAOA as

$$k_{\text{best-1sym-QAOA}} := \frac{f(x^*_{\text{ma}}) - f(x^*_{\text{best-1sym-QAOA}})}{f(x^*_{\text{ma}}) - f(x^*_{\text{QAOA}})}, \tag{5.2}$$

where $f(x^*_{\mathsf{y}})$ denotes the approximate Max-Cut found by $\mathsf{y} - QAOA$. When $k = 0$, best-1sym-QAOA recovers ma-QAOA; and when $k = 1$, best-1sym-QAOA performs the same as QAOA. Thus,

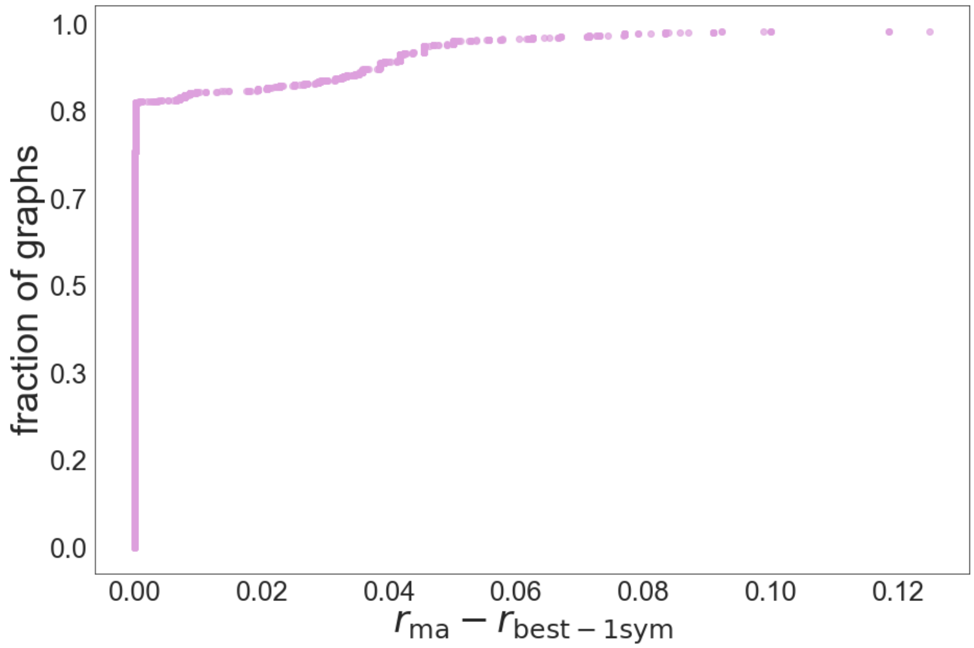

Figure 5.1: Difference of approximation ratio r between ma-QAOA and best-1sym-QAOA, as defined in (5.1). For most graphs with symmetry, one symmetry can be used to reduce the number of parameters without affecting the solution quality.

this ratio indicates whether best-1sym-QAOA performance is closer to ma-QAOA performance or QAOA performance. In this study, the denominator of the ratio is always nonzero.

It is encouraging that the approximation ratios for best-1sym-QAOA and ma-QAOA are similar. But this finding is of limited value if both methods use the same number of parameters. We therefore consider the quantity $l_{\text{best-1sym-QAOA}}$, which is the relative difference in parameters between ma-QAOA and best-1sym-QAOA (as compared with the difference in the number of parameters between ma-QAOA and QAOA). That is,

$$l_{\text{best-1sym-QAOA}} := \frac{|E| + |V| - (|\mathcal{O}_e(\sigma)| + |\mathcal{O}_v(\sigma)|)}{|E| + |V| - 2}, \tag{5.3}$$

where $|\mathcal{O}_v(\sigma)|$ and $|\mathcal{O}_e(\sigma)|$ are the number of vertex orbits and edge orbits, respectively, induced by the automorphism σ. This ratio determines how close the number of parameters in best-1sym-QAOA is to either ma-QAOA or QAOA, depending on whether the ratio is closer to 0 or not.

5.5.2 max-sym-QAOA vs. ma-QAOA

Of the 11,117 connected, non-isomorphic 8-vertex graphs, $3,552$ have only trivial symmetries. In these cases max-sym-QAOA is ma-QAOA. Thus, our analysis focuses on the $7,565$ graphs that contain nontrivial symmetry, where max-sym-QAOA has fewer parameters to optimize over. As shown in Figure 5.2, max-sym-QAOA performs as well as ma-QAOA on 2,713 of these graphs. Furthermore, it performs the same as QAOA on 30 graphs, which is only about 0.4% of the graphs with nontrivial symmetry. These results indicate that max-sym-QAOA performance is comparable, even though it requires fewer parameters. We define the ratio of the difference in the Max-Cut values of ma-QAOA and max-sym-QAOA to the difference in Max-Cut values of ma-QAOA and QAOA as

$$k_{\textsf{max-sym}} := \frac{f(x^*_{\textsf{ma}}) - f(x^*_{\textsf{max-sym}})}{f(x^*_{\textsf{ma}}) - f(x^*_{\textsf{QAOA}})}. \tag{5.4}$$

When $k = 0$, max-sym-QAOA recovers ma-QAOA; and when $k = 1$, max-sym-QAOA performs the same as QAOA. Thus, this ratio indicates whether max-sym-QAOA's performance is closer to ma-QAOA's performance or QAOA's performance. In this study, the denominator of the ratio is always nonzero.

We quantify the ratio of the reduction of parameters from ma-QAOA to max-sym-QAOA over the difference in parameters between ma-QAOA and QAOA as

$$l_{\textsf{max-sym}} := \frac{|E| + |V| - (|\mathcal{O}_e| + |\mathcal{O}_v|)}{|E| + |V| - 2}, \tag{5.5}$$

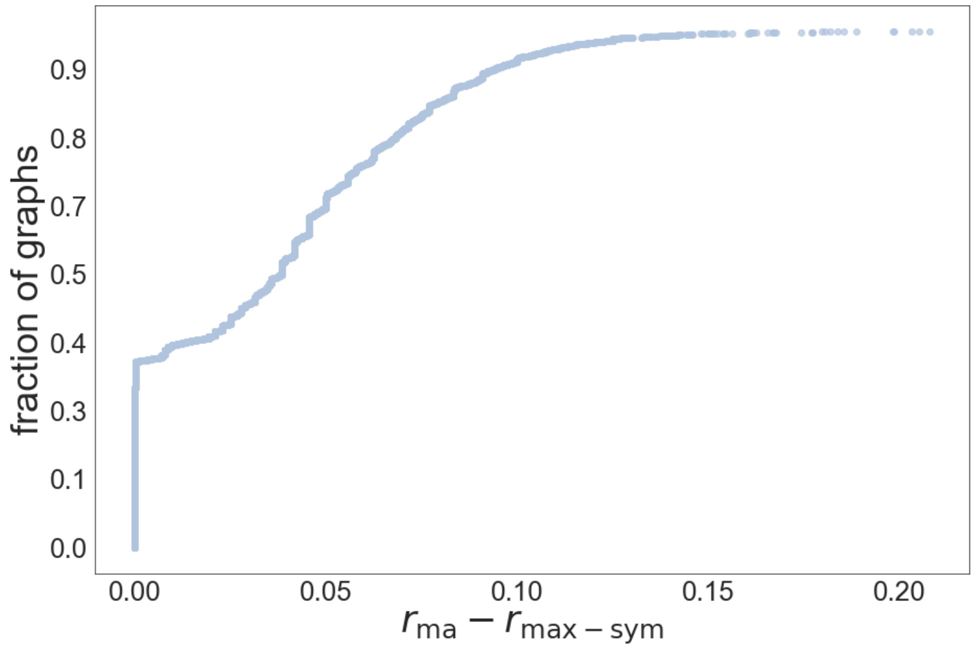

Figure 5.2: Difference of approximation ratio r between ma-QAOA and max-sym-QAOA, as defined in (5.1). In a plurality of graphs with symmetry, simply using the largest symmetry leads to a reduction in the number of parameters with no impact on solution quality.

since ma-QAOA uses $|E| + |V|$ parameters, max-sym-QAOA requires $|\mathcal{O}_e| + |\mathcal{O}_v|$ parameters, and QAOA uses two parameters.

Figure 5.3 shows that a positive correlation exists between the quantities k and l. Among results that achieve the result equivalent to ma-QAOA, the number of parameters reduced is spread nearly evenly over $[0.1, 0.8]$. Since there are almost no points between $[0, 0.1]$, max-sym-QAOA almost always requires at least 10% fewer parameters than ma-QAOA requires. $l_{\text{max-sym}}$ averaged over all those graphs with nontrivial symmetries is 0.37.

Note that specifically for those graphs where reducing the number of parameters leads to a decrease in the objective, max-sym-QAOA can reduce the parameter count by 37.1% at the cost of only a 6.1% decrease in the objective.

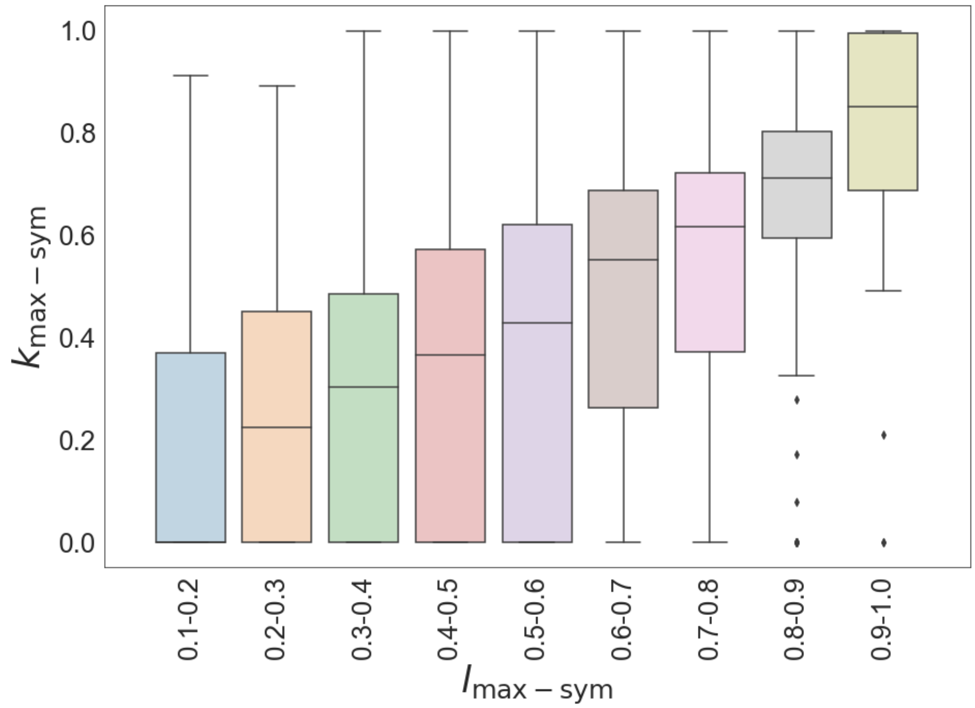

Figure 5.3: Comparison of $k_{\text{max-sym}}$ and $l_{\text{max-sym}}$ for max-sym-QAOA, as defined in (5.4) and (5.5). There is a positive correlation between these two variables.

5.5.3 Evidence for central role of symmetries

max-sym-QAOA is also compared with rand-group-QAOA, which groups vertices and edges randomly so that the corresponding number of parameters is the same as that of max-sym-QAOA. Figure 5.4 demonstrates the centrality of symmetries to ma-QAOA parameter redundancy by showing that the parameter reduction strategy of max-sym-QAOA has a clear advantage over rand-group-QAOA in terms of solution quality.

5.5.4 max-sym-QAOA vs. best-1sym-QAOA

In this section we compare max-sym-QAOA and best-1sym-QAOA on the 5,918 graphs for which best-sym-QAOA requires fewer parameters than does ma-QAOA.

Figure 5.5 indicates that best-1sym-QAOA has $k = 0$ on nearly twice as many graphs as max-sym-QAOA. Additionally, $k < 0.6$ for the majority of graphs solved with best-1sym-QAOA while k is spread over $[0, 1]$ with max-sym-QAOA.

Although max-sym-QAOA does not perform as well as best-1sym-QAOA on the majority of graphs, the average $l_{\text{max-sym}}$ is around 0.39 while the average $l_{\text{best-sym}}$ is only 0.31, so best-1sym-QAOA has more parameters than max-sym-QAOA, on average.

5.6 Discussion

In this work we demonstrate the connection between the parameter redundancy in ma-QAOA and the symmetries of the problem to be optimized. Specifically, we show that the number of parameters in ma-QAOA can often be dramatically reduced without affecting the solution quality. To that

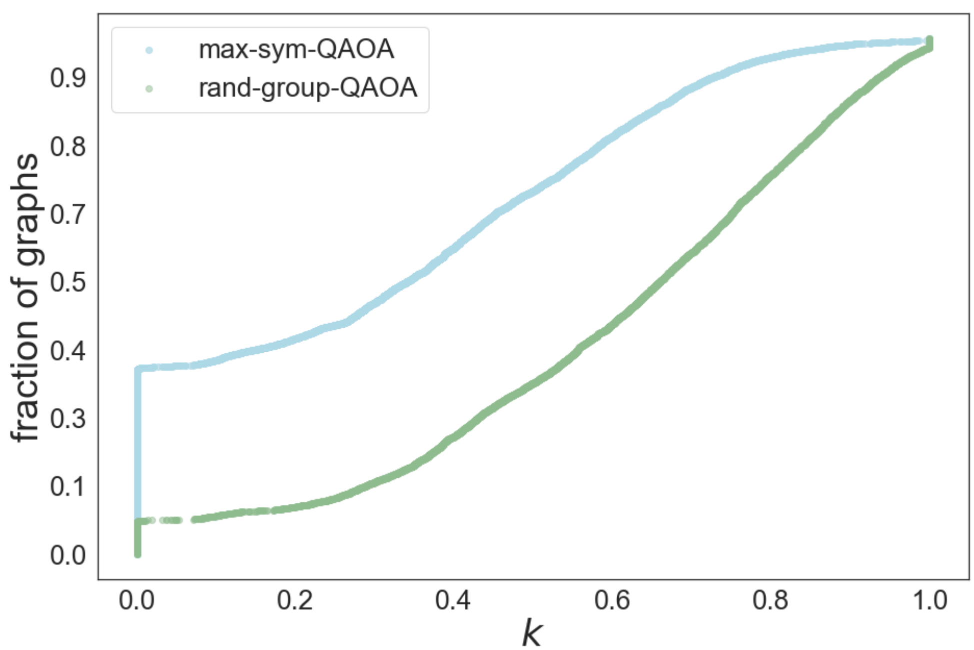

Figure 5.4: Fraction of graphs achieving ratio k for max-sym-QAOA and rand-group-QAOA. If a random parameter reduction is used, the performance deteriorates significantly, suggesting the central role of symmetries.

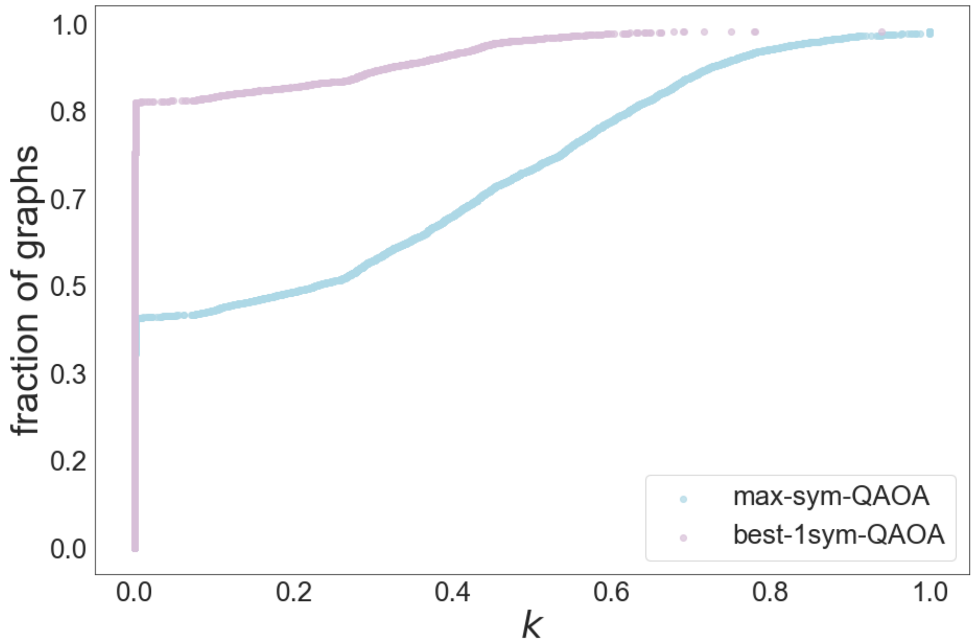

Figure 5.5: Fraction of graphs achieving ratio k for max-sym-QAOA and best-1sym-QAOA, as defined in (5.2) and (5.4).

end, we introduce three QAOA variations that require fewer parameters than ma-QAOA: best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA. The three algorithms assign classical parameters based on the symmetries (automorphisms) of the underlying problem graph. We evaluate these algorithms on all connected, non-isomorphic 8-vertex graphs and compare the results with those of ma-QAOA.

In most cases, max-sym-QAOA requires at least 10% fewer parameters than ma-QAOA does, while maintaining a comparable approximation ratio, which is the primary metric of QAOA success. In fact, in over one-third of the connected 8-vertex graphs with nontrivial symmetry, max-sym-QAOA finds the same approximate Max-Cut as does ma-QAOA. Furthermore, a positive correlation exists between the number of parameters reduced and the reduction in the approximation ratio, as expected. Additionally, significantly more graphs have $k = 0$ when solved with max-sym-QAOA than rand-group-QAOA, implying that max-sym-QAOA outperforms rand-group-QAOA in general. On the other hand, significantly more graphs have $k = 0$ when solved with best-1sym-QAOA than max-sym-QAOA, yet max-sym-QAOA needs fewer parameters (on average) than does best-1sym-QAOA.

Thus, out of best-1sym-QAOA, max-sym-QAOA, and rand-group-QAOA, rand-group-QAOA appears to have the worst performance while best-1sym-QAOA appears to have the closest performance to ma-QAOA on these small graphs. The failure of rand-group-QAOA demonstrates the importance of symmetry to parameter setting in QAOA.

Sauvage et al. proposed using symmetries to improve the performance of variational quantum algorithms [SLCC22]. They observed that in ma-QAOA applied to the Max-Cut problem, the number of parameters can be reduced and the trainability improved by using an approach equivalent to max-sym-QAOA described in this work. Unlike [SLCC22], we highlight the role of symmetries in quantum optimization by showing that symmetry-based parameter reduction leads to much better performance than does a random approach. Moreover, we consider utilizing a part of the symmetry group (best-1sym-QAOA).

Approximately one-third of the graphs considered in this study had only trivial symmetry, and max-sym-QAOA is equivalent to ma-QAOA in these cases. More broadly, only a limited fraction of problem instances exhibit nontrivial symmetry in the first place, which suggests that symmetry-based parameter reduction is inherently restricted in its general applicability as a tool for improving ma-QAOA and QAOA. Nonetheless, numerical evidence in [HLO+22] suggests that parameter redundancy may still be present even for graphs with trivial symmetry. Therefore, an important future direction is to understand whether and how such redundancy can be systematically exploited for problem instances with no explicit symmetries.

Furthermore, even among instances that do possess symmetry, our results are primarily based on numerical evidence, and there are currently no theoretical guarantees on when symmetry-based reductions are effective, or on how much the parameter space can be reduced without degrading performance. Developing a theoretical understanding of the scope and limitations of symmetry-based structural methods for reducing optimization complexity thus remains an important open problem.

Bibliography

[AAB$^+$19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[ABK$^+$24] Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz, and Patrick Struck. Post-quantum security of tweakable even-mansour, and applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 310–338. Springer, 2024.

[ABKM22] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the even-mansour cipher. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 458–487. Springer, 2022.

[ABMS25] Gorjan Alagic, Chen Bai, Christian Majenz, and Kaiyan Shi. Post-quantum security of block cipher constructions. *arXiv preprint arXiv:2510.08725*, 2025.

[ABPS24] Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. *IACR Communications in Cryptology*, 1(1), 2024.

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 269–295. Springer, 2019.

[ALMO08] Andris Ambainis, Debbie Leung, Laura Mancinska, and Maris Ozols. Quantum random access codes with shared randomness. *arXiv preprint arXiv:0810.2937*, 2008.

[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002.

[ANTSV99] Andris Ambainis, Ashwin Nayak, Ammon Ta-Shma, and Umesh Vazirani. Dense quantum coding and a lower bound for 1-way quantum automata. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 376–383, 1999.

[ATTU16] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Post-quantum security of the cbc, cfb, ofb, ctr, and xts modes of operation. In *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7*, pages 44–63. Springer, 2016.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 684–697, 2016.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

[BBC+25] Jyotirmoy Basak, Ritam Bhaumik, Amit Kumar Chauhan, Ravindra Jejurikar, Ashwin Jha, Anandarup Roy, André Schrottenloher, and Suprita Talnikar. Post-quantum security of key-alternating feistel ciphers. *Cryptology ePrint Archive*, 2025.

[BCF+22] Ritam Bhaumik, André Chailloux, Paul Frixons, Bart Mennink, and María Naya-Plasencia. Block cipher doubling for a post-quantum world. *Cryptology ePrint Archive*, 2022.

[BCG+12] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, et al. Prince–a low-latency block cipher for pervasive computing applications. In *International conference on the theory and application of cryptology and information security*, pages 208–225. Springer, 2012.

[BDPA11] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Cryptographic sponge functions. Submission to NIST (Round 3), 2011.

[BEM24] Chen Bai, Mehdi Esmaili, and Atul Mantri. Quantum security analysis of the key-alternating ciphers. *arXiv preprint arXiv:2412.05026*, 2024.

[BHNP+19] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: the offline simon's algorithm. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 552–583. Springer, 2019.

[BHT97] Gilles Brassard, Peter Hoyer, and Alain Tapp. Quantum algorithm for the collision problem. *arXiv preprint quant-ph/9705002*, 1997.

[BLNPS21] Xavier Bonnetain, Gaëtan Leurent, María Naya-Plasencia, and André Schrottenloher. Quantum linearization attacks. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*, pages 422–452. Springer, 2021.

[BSS22] Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. Beyond quadratic speedups in quantum attacks on symmetric schemes. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 315–344. Springer, 2022.

[BY23] Aleksandrs Belovs and Duyal Yolcu. One-way ticket to las vegas and the quantum adversary. *arXiv preprint arXiv:2301.02003*, 2023.

[CAB$^+$21] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[Car25] Joseph Carolan. Compressed permutation oracles. *arXiv preprint arXiv:2509.18586*, 2025.

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

[Chi17] Andrew M Childs. Lecture notes on quantum algorithms. *Lecture notes at University of Maryland*, 5:3, 2017.

[CHL+25] Alexandru Cojocaru, Minki Hhan, Qipeng Liu, Takashi Yamakawa, and Aaram Yun. Quantum lifting for invertible permutations and ideal ciphers. In *Annual International Cryptology Conference*, pages 481–512. Springer, 2025.

[CLQ19] Kai-Min Chung, Tai-Ning Liao, and Luowen Qian. Lower bounds for function inversion with quantum advice. *arXiv preprint arXiv:1911.09176*, 2019.

[CMM23] Matthew Campagna, Alexander Maximov, and John Preuß Mattsson. Galois counter mode with secure short tags (gcm-sst). 2023.

[CNPS17] André Chailloux, María Naya-Plasencia, and André Schrottenloher. An efficient quantum collision search algorithm and implications on symmetric cryptography. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 211–240. Springer, 2017.

[CSR+08] David Clunie, Rich Shroeppel, Phillip Rogaway, Vijay Bharadwaj, and Neils Ferguson. Public comments on the xts-aes mode. *Collected email comments released by NIST, available from their web page*, 2008.

[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. *Theoretical Computer Science*, 855:16–42, 2021.

[DBN+01] Morris J Dworkin, Elaine Barker, James R Nechvatal, James Foti, Lawrence E Bassham, E Roback, James F Dray Jr, et al. Advanced encryption standard (aes). *NIST publication*, 2001.

[DH08] Catalin Dohotaru and Peter Hoyer. Exact quantum lower bound for grover's problem. *arXiv preprint arXiv:0810.3647*, 2008.

[Din15] Itai Dinur. Cryptanalytic time-memory-data tradeoffs for fx-constructions with applications to prince and pride. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 231–253. Springer, 2015.

[DKRS23] Orr Dunkelman, Nathan Keller, Eyal Ronen, and Adi Shamir. Quantum time/memory/data tradeoff attacks. *Designs, Codes and Cryptography*, pages 1–19, 2023.

[Dwo01] Morris Dworkin. Recommendation for block cipher modes of operation. *NIST special publication*, 800:38B, 2001.

[Dwo07] Morris J Dworkin. *Sp 800-38d. recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac*. National Institute of Standards & Technology, 2007.

[Dwo10a] Morris J Dworkin. Recommendation for block cipher modes of operation: The xts-aes mode for confidentiality on storage devices. 2010.

[Dwo10b] Dworkin, Morris. Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices. Technical Report SP 800-38E, National Institute of Standards and Technology (NIST), Gaithersburg, MD, February 2010.

[Dwo15] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 2015.

[EMST78] William F Ehrsam, Carl HW Meyer, John L Smith, and Walter L Tuchman. Message verification and transmission error detection by block chaining, February 14 1978. US Patent 4,074,066.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv:1411.4028*, 2014.

[FK15] Bill Fefferman and Shelby Kimmel. Quantum vs classical proofs and subset verification. *arXiv preprint arXiv:1510.06750*, 2015.

[GHHM21] Alex B Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the qrom. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 637–667. Springer, 2021.

[GJMG11] Bertoni Guido, Daemen Joan, P Michaël, and VA Gilles. Cryptographic sponge functions, 2011.

[GLL17] Shay Gueron, Adam Langley, and Yehuda Lindell. Aes-gcm-siv: specification and analysis. *Cryptology ePrint Archive*, 2017.

[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*, pages 212–219, 1996.

[GT12] Peter Gaži and Stefano Tessaro. Efficient and optimally secure key-length extension for block ciphers via randomized cascading. In *Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31*, pages 63–80. Springer, 2012.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[Her22] Rebekah Herrman. Investigating parameters in the multi-angle approximate optimization algorithm. In *APS March Meeting Abstracts*, volume 2022, pages G00–075, 2022.

[HLO$^+$22] Rebekah Herrman, Phillip C Lotshaw, James Ostrowski, Travis S Humble, and George Siopsis. Multi-angle quantum approximate optimization algorithm. *Scientific Reports*, 12(1):1–10, 2022.

[HLZ04] Eran Halperin, Dror Livnat, and Uri Zwick. Max cut in cubic graphs. *Journal of Algorithms*, 53(2):169–185, 2004.

[Hos25] Akinori Hosoyamada. Post-quantum security of keyed sponge-based constructions through a modular approach. *Cryptology ePrint Archive*, 2025.

[HP00] Tad Hogg and Dmitriy Portnov. Quantum optimization. *Information Sciences*, 128(3-4):181–197, 2000.

[HS18] Akinori Hosoyamada and Yu Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *Cryptographers' Track at the RSA Conference*, pages 198–218. Springer, 2018.

[HSN+21] Matthew P Harrigan, Kevin J Sung, Matthew Neeley, Kevin J Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C Bardin, Rami Barends, Sergio Boixo, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics*, 17(3):332–336, 2021.

[HTO+21] Rebekah Herrman, Lorna Treffert, James Ostrowski, Phillip C Lotshaw, Travis S Humble, and George Siopsis. Impact of graph structures for QAOA on MaxCut. *Quantum Information Processing*, 20(9):1–21, 2021.

[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*, pages 584–614. Springer, 2019.

[HY24] Minki Hhan and Shogo Yamada. Pseudorandom function-like states from common haar unitary. *arXiv preprint arXiv:2411.03201*, 2024.

[IJMM24] Akiko Inoue, Ashwin Jha, Bart Mennink, and Kazuhiko Minematsu. Generic security of gcm-sst. *Cryptology ePrint Archive*, 2024.

[IK03] Tetsu Iwata and Kaoru Kurosawa. Omac: One-key cbc mac. In *Fast Software Encryption: 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers 10*, pages 129–153. Springer, 2003.

[IM19] Takanori Isobe and Kazuhiko Minematsu. Plaintext recovery attacks against xts beyond collisions. In *International Conference on Selected Areas in Cryptography*, pages 103–123. Springer, 2019.

[IOM12] Tetsu Iwata, Keisuke Ohashi, and Kazuhiko Minematsu. Breaking and repairing gcm security proofs. In *Advances in Cryptology–CRYPTO 2012: 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 31–49. Springer, 2012.

[JN20] Ashwin Jha and Mridul Nandi. Tight security of cascaded lrw2. *Journal of Cryptology*, 33(3):1272–1317, 2020.

[JST21] Joseph Jaeger, Fang Song, and Stefano Tessaro. Quantum key-length extension. In *Theory of Cryptography Conference*, pages 209–239. Springer, 2021.

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[KLGS22] Ashish Kakkar, Jeffrey Larson, Alexey Galda, and Ruslan Shaydulin. Characterizing error mitigation by symmetry verification in QAOA. *arXiv:2204.05852*, 2022.

[KLLNP16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II 36*, pages 207–237. Springer, 2016.

[KM12] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type even-mansour cipher. In *2012 international symposium on information theory and its applications*, pages 312–316. IEEE, 2012.

[KR96] Joe Kilian and Phillip Rogaway. How to protect des against exhaustive key search. In *Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16*, pages 252–267. Springer, 1996.

[KR01] Joe Kilian and Phillip Rogaway. How to protect des against exhaustive key search (an analysis of desx). *Journal of Cryptology*, 14(1):17–35, 2001.

[KSGB23] Zeki Kazan, Kaiyan Shi, Adam Groce, and Andrew P Bray. The test of tests: A framework for differentially private hypothesis testing. In *International Conference on Machine Learning*, pages 16131–16151. PMLR, 2023.

[LH] Phillip C. Lotshaw and Travis S. Humble. QAOA dataset. Found at https://code.ornl.gov/qci/qaoa-dataset-version1.

[Liu23] Qipeng Liu. Non-uniformity and quantum advice in the quantum random oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 117–143. Springer, 2023.

[LL23] Nathalie Lang and Stefan Lucks. On the post-quantum security of classical authenticated encryption schemes. In *International Conference on Cryptology in Africa*, pages 79–104. Springer, 2023.

[LM08] Moses Liskov and Kazuhiko Minematsu. Comments on xts-aes. *Comments to NIST, available from their web page*, 2008.

[LRW02] Moses Liskov, Ronald L Rivest, and David Wagner. Tweakable block ciphers. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pages 31–46. Springer, 2002.

[McK] Brendan McKay. Graphs [data set]. Found at http://users.cecs.anu.edu.au/ bdm/data/graphs.html.

[McK07] Brendan D McKay. Nauty user's guide (version 2.4). *Computer Science Dept., Australian National University*, pages 225–239, 2007.

[Min06] Kazuhiko Minematsu. Improved security analysis of xex and lrw modes. In *International Workshop on Selected Areas in Cryptography*, pages 96–113. Springer, 2006.

[MLA+22] Lars S Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob FF Bulmer, Filippo M Miatto, Leonhard Neuhaus, Lukas G Helt, Matthew J Collins, et al. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, 2022.

[MMB22] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379, 2022.

[MP13] Brendan D McKay and Adolfo Piperno. Nauty and Traces user's guide (Version 2.5). *Computer Science Department, Australian National University, Canberra, Australia*, 2013.

[MV04] David A McGrew and John Viega. The security and performance of the galois/counter mode of operation (full version). *Cryptology ePrint Archive*, 2004.

[MW13] Catherine C McGeoch and Cong Wang. Experimental evaluation of an adiabiatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers*, pages 1–11, 2013.

[NABT14] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. Quantum lower bound for inverting a permutation with advice. *arXiv preprint arXiv:1408.3193*, 2014.

[Nay10] Ashwin Nayak. Inverting a permutation is as hard as unordered search. *arXiv preprint arXiv:1007.2899*, 2010.

[NOMI15] Yuichi Niwa, Keisuke Ohashi, Kazuhiko Minematsu, and Tetsu Iwata. Gcm security bounds reconsidered. In *Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers 22*, pages 385–407. Springer, 2015.

[NSY$^+$22] Pradeep Niroula, Ruslan Shaydulin, Romina Yalovetzky, Pierre Minssen, Dylan Herman, Shaohan Hu, and Marco Pistoia. Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer, 2022.

[P1606] Ieee draft standard for authenticated encryption with length expansion for storage device. *IEEE Unapproved Draft Std P1619.1/D4, Oct 2007*, 2006.

[P1619] Ieee standard for cryptographic protection of data on block-oriented storage devices. *IEEE Std 1619-2018 (Revision of IEEE Std 1619-2007)*, pages 1–41, 2019.

[PR00] Erez Petrank and Charles Rackoff. Cbc mac for real-time data sources. *Journal of Cryptology*, 13:315–338, 2000.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[RMP+22] Martin Ringbauer, Michael Meth, Lukas Postler, Roman Stricker, Rainer Blatt, Philipp Schindler, and Thomas Monz. A universal qudit quantum processor with trapped ions. *Nature Physics*, pages 1–5, 2022.

[Rog02] Phillip Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 98–107, 2002.

[Rog04] Phillip Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 16–31. Springer, 2004.

[Rog13] Phillip Rogaway. Evaluation of some blockcipher modes of operation. 2011. *Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan*, 2013.

[Ros21] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. *arXiv preprint arXiv:2103.08975*, 2021.

[SA19] Ruslan Shaydulin and Yuri Alexeev. Evaluating quantum approximate optimization algorithm: A case study. In *International Green and Sustainable Computing Conference*, pages 1–6. IEEE, 2019.

[SCK+24] Kaiyan Shi, Kaushik Chakraborty, Wen Yu Kon, Omar Amer, Marco Pistoia, and Charles Lim. On the relativistic zero knowledge quantum proofs of knowledge. *arXiv preprint arXiv:2409.03635*, 2024.

[SG21] Ruslan Shaydulin and Alexey Galda. Error mitigation for deep quantum optimization circuits by leveraging problem symmetries. In *International Conference on Quantum Computing and Engineering*. IEEE, 2021.

[SHHS21] Ruslan Shaydulin, Stuart Hadfield, Tad Hogg, and Ilya Safro. Classical symmetries and the quantum approximate optimization algorithm. *Quantum Information Processing*, 20(11):1–28, 2021.

[Sho94] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[SHS+22] Kaiyan Shi, Rebekah Herrman, Ruslan Shaydulin, Shouvanik Chakrabarti, Marco Pistoia, and Jeffrey Larson. Multiangle qaoa does not always need all its angles. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 414–419. IEEE, 2022.

[SLCC22] Frederic Sauvage, Martin Larocca, Patrick J Coles, and M Cerezo. Building spatial symmetries into parameterized quantum circuits for faster training. *arXiv:2207.14413*, 2022.

[SS19] Shingo Sato and Junji Shikata. So-cca secure pke in the quantum random oracle model or the quantum ideal cipher model. In *IMA International Conference on Cryptography and Coding*, pages 317–341. Springer, 2019.

[Vau00] Serge Vaudenay. Decorrelation over infinite domains: the encrypted cbc-mac case. In *International Workshop on Selected Areas in Cryptography*, pages 189–201. Springer, 2000.

[Vaz98] Umesh Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London A 365: 1759-1768*, 1998.

[WBC+21] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al. Strong quantum computational advantage using a superconducting quantum processor. *Physical Review Letters*, 127(18):180501, 2021.

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

[WL21] Jonathan Wurtz and Peter Love. Maxcut quantum approximate optimization algorithm performance guarantees for p¿ 1. *Physical Review A*, 103(4):042612, 2021.

[Zal99] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746, 1999.

[Zha16] Mark Zhandry. A note on quantum-secure prps. *arXiv preprint arXiv:1611.05564*, 2016.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryp-*

tology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39, pages 239–268. Springer, 2019.