ABSTRACT

Title of Dissertation: SYMMETRIC-KEY CRYPTOGRAPHY AND QUERY COMPLEXITY IN THE QUANTUM WORLD

Chen Bai
Doctor of Philosophy, 2024

Dissertation Directed by: Professor Gorjan Alagic and Jonathan Katz
Department of Electrical and Computer Engineering

Quantum computers are likely to have a significant impact on cryptography. Many commonly used cryptosystems will be completely broken once large quantum computers are available. Since quantum computers can solve the factoring problem in polynomial time, the security of RSA would not hold against quantum computers. For symmetric-key cryptosystems, the primary quantum attack is key recovery via Grover search, which provides a quadratic speedup. One way to address this is to double the key length. However, recent results have shown that doubling the key length may not be sufficient in all cases. Therefore, it is crucial to understand the security of various symmetric-key constructions against quantum attackers.

In this thesis, we give the first proof of post-quantum security for certain symmetric primitives. We begin with a fundamental block cipher, the Even-Mansour cipher, and the tweakable Even-Mansour construction. Our research shows that both are secure in a realistic quantum attack model. For example, we prove that $2^{n/3}$ quantum queries are necessary to break the Even-Mansour cipher. We also consider the practical applications that our work implies. Using our

framework, we derive post-quantum security proofs for three concrete symmetric-key schemes: Elephant (an Authenticated Encryption (AE) finalist of NIST's lightweight cryptography standardization effort), Chaskey (an ISO-standardized Message Authentication Code), and Minalpher (an AE second-round candidate of the CAESAR competition).

In addition, we consider the two-sided permutation inversion problem in the quantum query model. In this problem, given an image y and quantum oracle access to a permutation P (and its inverse oracle), the goal is to find its pre-image x such that P(x)=y. We prove an optimal lower bound $\Omega(\sqrt{2^n})$ for this problem, against an adaptive quantum adversary. Moreover, we apply our lower bound above to show that a natural encryption scheme constructed from random permutations is secure against quantum attacks.

SYMMETRIC-KEY CRYPTOGRAPHY AND QUERY COMPLEXITY IN THE QUANTUM WORLD

by

Chen Bai

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
2024

Advisory Committee:
Professor Jonathan Katz, Chair/Advisor
Professor Gorjan Alagic, Co-Chair/Advisor
Professor Dana Dachman-Soled
Professor Christian Majenz
Professor Prakash Narayan
Professor Andrew Childs, Dean's Representative

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I would like to express my sincere gratitude to my advisors, Professor Jonathan Katz and Professor Gorjan Alagic. When I first approached Jon, I had no prior experience in cryptography. I am deeply appreciative of his acceptance as his student, which granted me entry into his reading group and sparked my interest in cryptography. I would also like to express my heartfelt thanks to Gorjan for introducing me to the intersection of quantum computing and cryptography and providing unwavering support from the beginning of my journey in this field. Over the past years, I would like to thank both of them for giving me an invaluable opportunity to work on challenging and extremely interesting projects. Their expertise, patience, and unwavering commitment have been instrumental in shaping the direction of my research and refining my academic skills. It has been a pleasure to work with and learn from them. Thank you, Jon and Gorjan, for everything!

I would like to thank Professor Andrew Childs, Professor Prakash Narayan, Professor Dana Dachman-Soled, and Professor Christian Majenz for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing the manuscript.

A huge shoutout to my co-authors for the incredible and productive collaborations. Alongside my advisors, special thanks to Christian Majenz, Alexander Poremba, Kaiyan Shi, and

Patrick Struck for their invaluable contributions through insightful discussions and joint paper writing sessions. I would also like to thank my group mates Joseph Carolan, Manasi Shingane and Kaiyan Shi for countless discussions on various topics.

During my PhD program, I had the pleasure of meeting some amazing friends who made my journey a lot more fun and memorable. Bibhusa Rawal and Brian Kim were among the first friends I made during my PhD orientation, and we've been close ever since. I want to express my gratitude to them for the countless late-night study sessions, uplifting conversations, and just being there for me during the early stages of my PhD. I'd also like to thank Lei Chen, Jinjing Han, Yuqian Hu, and Jun Wang for their unwavering support and camaraderie throughout my doctoral journey. I'm especially thankful to Zeyu Zhang, who is an excellent gym trainer and a great friend. Moreover, I want to thank Tian Gan for being such a wonderful friend, even though we've known each other for a relatively short time. Lastly, I want to express my appreciation to all the people I didn't mention but who certainly would have deserved it.

I would like to express my heartfelt appreciation to my parents, who have been my unwavering source of support and faith throughout my life. Being a foreign student, I don't get to visit my home country frequently, but my parents have always been there for me. I would also like to extend my gratitude to my relatives living in the United States, including my aunt, uncle, older cousin, and little cousin for their constant support. Thank you!

Table of Contents

List of Figures

List of Abbreviations

AE	Authenticated encryption
AEAD	Authenticated encryption with associated data
AES	Advanced Encryption Standard
aTPI	Adaptive two-sided permutation inversion problem
BBBV	Bennett, Bernstein, Brassard, Vazirani
CCA	Chosen-ciphertext attack
CPA	Chosen-plaintext attack
CRA	Chosen-randomness attack
DSA	Digital signature algorithm
ECC	Elliptic curve cryptography
EM	Even-Mansour
HSP	Hidden subgroup problems
IND	Indistinguishability
ISO	International Organization for Standardization
MA	Minalpher
MAC	Message authentication code
NIST	National Institute of Standards and Technology
OWF	One-way function
PKE	Public-key encryption
POVM	Positive-operator valued measure
PPT	Probabilistic polynomial-time
PQC	Post-quantum cryptography
PRF	Pseudorandom function
PRG	Pseudorandom generator
PRP	Pseudorandom permutation
QCCA	Quantum chosen-ciphertext attacks
QCCRA	Quantum chosen-claintext randomness-access attacks
QCPA	Quantum chosen-plaintext attacks
QCPRA	Quantum chosen-plaintext randomness-access attacks
QCRA	Quantum chosen-randomness attacks
QPT	Quantum probabilistic polynomial-time
QROM	Quantum random oracle model
SHA	Secure hash algorithm
SKE	Symmetric-key encryption

RSA	Rivest–Shamir–Adleman
TEM	Tweakable Even-Mansour
TEM-KX	Tweakable Even-Mansour with key expansion
TEM-KX1	Tweakable Even-Mansour with a specifically-designed key expansion
TPI	Two-sided permutation inversion problem

Chapter 1: Introduction

One of the goals of cryptography is to use encryption schemes to hide or code messages so that only the intended person can read them. Under this protection, two parties, which are referred to as the sender (Alice) and the receiver (Bob), can communicate secretly over an insecure channel in a way that no eavesdropper (Eve) can understand the message being sent from Alice to Bob. In the modern world, cryptography not only ensures the confidentiality of data but can also be used for other security goals such as authentication, integrity, and non-repudiation. For example, a Message Authentication Code (MAC) [4] ensures the authenticity and integrity of messages sent from Alice to Bob. Authenticated encryption (AE) [4] achieves authenticity, confidentiality, and integrity at the same time. Digital signatures assure the authenticity, integrity, and non-repudiation of a sent message so that everyone can verify the signature with the public-key. In the modern digital world, cryptography plays a significant role, and its applications are ubiquitous.

The security of symmetric-key encryption schemes relies on a secret key shared by the communicating parties in advance but unknown to the eavesdropper. In this setting, Alice and Bob share a key and use this key to communicate secretly: Alice sends a message, or plaintext, to Bob by using the shared key to encrypt the message first and then get a ciphertext that is sent to Bob. Bob could then use the same key to decrypt the ciphertext and recover the original plaintext.

This is also called symmetric encryption since the message is encrypted and decrypted using the same key. On the other hand, if the message is encrypted and decrypted using different keys, this is called asymmetric or public-key encryption [4].

For encryption, ”perfect secrecy” states that the adversary will not learn anything about the message that was sent through observing the corresponding ciphertext, even with unlimited computational power [5]. However, perfect secrecy is not practical. Any perfectly secret encryption scheme requires a key that is at least as long as the message and the key can be used only once [4]. Therefore, for practical purposes, an encryption scheme is still considered secure if it leaks only a ”negligible” amount of information to an adversary with bounded computational power. This is called computational security.

With security definitions, an immediate concern is how to prove the security of a cryptosystem. The security proof is usually a reduction that transforms any attacker of the cryptosystem into a machine that breaks the underlying assumption. If the underlying assumption is hard to break, then so is the cryptosystem. A computational hardness assumption is a hypothesis that a problem cannot be solved efficiently (in polynomial time) by any known algorithm. One example is the factorization problem: given an integer, determine its prime factors. This problem is assumed to be a hard problem as the computational effort for all known classical algorithms grows sub-exponentially with the size of the integer to be factored. The security of RSA [6], a commonly used public-key cryptosystem, is based on the hardness of the factoring problem. The hardness assumption can also be more general. For example, the existence of a one-way function [7], i.e., a function that is easy to evaluate but hard to invert [8, 9], implies the existence of many useful cryptographic tools such as pseudorandom generators, message authentication codes, and digital signature schemes [10, 11].

Provable security relies strongly on the underlying hardness assumption. All hardness assumptions are based on problems that are hard to solve with current knowledge and computational power. However, in recent years, there has been a substantial amount of research on quantum computers. These devices leverage the principles of quantum mechanics to perform certain types of calculations at speeds that are impossible for all known algorithms for classical (i.e., non-quantum) computers [12, 13, 14].

The ability of quantum computers to store, transmit, and process quantum data also opens many new possibilities for information processing. Although large-scale quantum computers are still in the experimental and research phase, many scientists believe quantum computing can be achieved soon [15].

1.1 Post-quantum Cryptography

In 1994, Peter Shor introduced an efficient quantum algorithm [12] to solve the factorization problem, which runs in polynomial time. Specifically, to factor an integer N using a quantum computer, the time Shor's algorithm takes is polynomial in $\log N$. The most efficient known classical algorithm works in sub-exponential time. Therefore, Shor's algorithm shows that a quantum computer can achieve a potentially super polynomial speedup. Subsequent works have shown that such a speedup can also be applied to other related problems, such as discrete logarithm problems [13] and hidden subgroup problems (HSP) [16], which are considered hard in the classical world. Since the security of many currently widely used public-key cryptosystems, such as RSA, ECC, and DSA [6, 17, 18], is based on the hardness of these problems, quantum computers will have a devastating impact on modern public-key cryptography. Therefore, one

important and urgent goal for cryptographers is to develop cryptographic systems that are secure against both quantum and classical computers and can interoperate with existing communication protocols and networks.

For symmetric-key cryptography, the threat from quantum computers does not seem as devastating as for public-key cryptography. The reason is that public-key cryptosystems are usually constructed from some well-understood mathematical problems, such as the factorization problem. The security of these systems relies heavily on the hardness of those problems. If the underlying problem is broken, all the related public-key cryptosystems become insecure. On the other hand, symmetric primitives are more related to "structureless" problems. Ideally, designers claim that a symmetric-key scheme, such as a block cipher, is secure by proving that it is indistinguishable from a truly random permutation. This is usually accomplished by proving its resistance against attacks such as key search (brute-force) and differential attacks [19]. The security of symmetric primitives doesn't rely on hard mathematical problems [20]. Therefore, they are not impacted by the classical hard problems that quantum computers can solve much faster. Symmetric primitives suffer from reduced security against quantum attacks, but this security reduction is much less drastic than for many asymmetric primitives. So far, the most common quantum attack on symmetric algorithms follows from Grover's algorithm [21] for searching an unstructured database of size N in $O(N^{1/2})$ time. The best-known classical algorithm for this problem needs no fewer than $O(N)$ operations, which implies that the application of Grover's algorithm could offer a quadratic speedup on key search. One immediate solution is to double the key length [22] to offer the same level of security against quantum algorithms. This idea leads to a recommendation of applying the current symmetric encryption standard AES [23] with 256-bit keys (AES-256) instead of AES with 128-bit keys (AES-128), which was initially proposed by

the PQCRYPTO project ”Post-Quantum Cryptography for Long-Term Security” in 2015 [24].

1.2 Quantum Query Complexity

As we've seen, quantum computers will have a huge impact on the modern cryptography world as soon as they are applicable. Therefore, it's very important to study post-quantum secure cryptosystems. A natural question is: How do we quantify the advantages and limitations of quantum computers to get some level of post-quantum security? In theoretical computer science, a useful model for studying such problems is called the ”query complexity model.” In the quantum world, it studies how many quantum queries a computational algorithm needs to solve a particular oracle problem. Such quantum queries are usually made by ”quantum oracles.” In the quantum oracle model, the function for a specific situation is abstracted as a ”black box, ” allowing a quantum algorithm to interact with this oracle through quantum queries to obtain information about the problem's input. These quantum queries are typically implemented as quantum gates or operations in applications.

The quantum query complexity model has played an important role in quantum computing theory for two main reasons. First, it captures most of the known quantum algorithms, such as Shor, Deutsch-Jozsa, Simon, and Grover [12, 14, 21, 25]. Second, query complexity could prove lower bounds, which are essential for learning the security of cryptographic primitives. Grover's algorithm is a good example: it's a quantum attack showing that no more than $O(\sqrt{N})$ queries are required to solve the unstructured search problem. On the other hand, the BBBV lower bound proved that $\Omega(\sqrt{N})$ queries are needed to solve such a problem [26]. Therefore, BBBV proves that Grover's attack is asymptotically optimal.

In this thesis, we focus on studying post-quantum security in the quantum query complexity model. We prove that some classically secure symmetric cryptographic primitives are post-quantum secure. We also provide time-space tradeoffs for certain complexity problems against quantum attacks. Moreover, we provide applications of our theoretical results.

1.3 Outline of Thesis

Chapter 2: Preliminaries. We start with a preliminary section that includes basic notations, concepts, and background knowledge that we will need for the entire thesis. In Section 2.1, we provide some basic concepts, mathematical notations, and terminologies. In Section 2.2, we recall some basic concepts from quantum computation, including quantum circuits, quantum oracles, and quantum attack models. In Section 2.3, we cover some essential concepts in classical cryptography, such as symmetric-key encryption schemes, security notions, and block ciphers.

Chapter 3: Technical results. In this chapter, we provide new technical results that are required to prove our main results. We also believe that these technical results are of independent interest. In Section 3.1, we start with an ”arbitrary reprogramming lemma,” which plays an important role in proving the post-quantum security of the Even-Mansour (EM) cipher [27]. In Section 3.2, we introduce another important technical lemma that is used in [27], ”the resampling lemma.” Additionally, we show how to generalize the resampling lemma so that it is sufficient to handle tweakable block ciphers. This new sampling lemma is the key ingredient to prove the post-quantum security of the tweakable Even-Mansour (TEM) cipher [28].

Chapter 4: Post-quantum security of Even Mansour Constructions. In this chapter, we prove the post-quantum security of the Even-Mansour Cipher and its tweakable version in the appropriate quantum attack model. The details of these works can be found in [27] and [28]. This chapter includes three sections:

- **Section 4.1: Even-Mansour Cipher.** In this section, we introduce our framework for proving the post-quantum security of the Even-Mansour cipher [27]. In Section 4.1.1, we start with an overview of the Even-Mansour cipher and its security in the quantum world. In general, quantum attacks are in two models: the Q1 model, where the adversary has quantum access to the public primitives but only classical access to the keyed primitives, and the Q2 model, where the adversary has quantum access to all oracles. We review existing quantum attacks in both models and discuss why the Q1 model is far more realistic. We then present our main result, Theorem 4.1, where we give a lower bound showing that $\approx 2^{n/3}$ queries are necessary for attacking the Even-Mansour cipher in the Q1 model. We also prove that our bound is optimal since it matches the best existing quantum attacks. In Section 4.1.2, we provide a formal proof. We employ the hybrid method and the game-playing technique [29], complemented by the resampling and arbitrary reprogramming lemma introduced in Section 3. It is worth noting that the adversary is *adaptive* in our setting, meaning that the adversary can adaptively select the order of quantum and classical queries. Moreover, in our main theorem, the adversary has both forward and inverse query access to the oracles. We argue that the inverse case is entirely symmetric to the forward case in section 4.1.3. We also take the forward-only case into account and give a lower bound in section 4.1.4.

- **Section 4.2: Tweakable Even-Mansour Cipher.** In this section, we develop a framework for proving the post-quantum security of the tweakable Even-Mansour cipher [28] in the Q1 model. In Section 4.2.1, we begin with an overview of tweakable block ciphers, delve into the increased complexity of proofs when incorporating tweaks, and elaborate on our approach to addressing this challenge by developing a new resampling lemma. (Lemma 3.5). In Section 4.2.2, we give the proofs of the following results: we prove the post-quantum security of three different variants of tweakable Even-Mansour constructions: the original tweakable Even-Mansour Cipher (TEM), the tweakable Even-Mansour with key expansion (TEM-KX), and the tweakable Even-Mansour with a specifically designed key expansion scheme (TEM-KX1). While the proof strategies for these theorems share similarities with Theorem 4.1, the details exhibit significant differences since they rely heavily on our new resampling lemma.

- **Section 4.3: Applications.** Our results in Sections 4.1 and 4.2 lead to many applications. In particular, our results, along with existing theorems, suggest post-quantum security for the following cryptographic constructions in an ideal model:

  1. **Elephant** [2]: An Authenticated Encryption scheme with Associated Data (AEAD) scheme, which is a finalist of NIST's lightweight cryptography standardization effort.
  2. **Minalpher** [3]: An AEAD scheme that is a second-round candidate of the CAESAR competition.
  3. **Chaskey** [1]: An ISO-standardized Message Authentication Code (MAC) scheme.

Chapter 5: Two-sided permutation inversion problems. In this chapter, we consider the two-sided permutation inversion problem (TPI) in the quantum world. Given an image y and quantum query access to a permutation π, the goal is to find its pre-image. In our setting, the adversary gets quantum query access to both the forward and inverse oracles of the permutation. To make the problem nontrivial, the inverse oracle will output a reject symbol when queried on the challenge image y. Moreover, we also consider the adaptive case, where the adversary gets to choose a part of the pre-image. As our main result, we prove a lower bound for solving this problem. The details of this work can be found in [28]. After a short review of the permutation inversion in Section 5.1, we start with the reduction in Section 5.2; given an algorithm that solves an adaptive version of the two-sided permutation inversion problem (aTPI), we construct another algorithm that solves the unstructured search problem (UNIQUESEARCH). In the aTPI problem, the adversary gets to choose a part of the pre-image μ first and then receives the challenge image y such that $y = \pi(x||\mu)$. We show that solving aTPI is as hard as solving UNIQUESEARCH, and our reduction can be directly converted to the normal TPI problem. Since the UNIQUESEARCH problem is known to have a tight lower bound [30], we are able to give an optimal lower bound for aTPI (TPI) through the reduction. We provide the bound in Section 5.3. As an application, in Section 5.4 we show that our results imply the one-wayness of a quantum variant of a chosen ciphertext attack model, namely QCCRA2.

Chapter 6: Conclusion and Outlook. In this section, we will briefly summarize our works and discuss future works.

Chapter 2: Preliminaries

2.1 Basic Notations

We start with some basic mathematical notations and terminologies. The set $[n]$ denotes $\{1, 2, ..., n\}$ and $\{0, 1\}^*$ denotes the set of all finite bit strings. The function $\mathsf{negl}(n)$ is a negligible function with parameter n, and poly is polynomial. PPT stands for classical probabilistic polynomial-time and QPT stands for quantum polynomial-time. For any set S, $x \leftarrow S$ means that an element x is sampled uniformly at random from the set S. We let $\mathcal{P}(n)$ denote the set of all permutations on $\{0, 1\}^n$. In the *public-permutation model* (or random permutation model), $P \leftarrow \mathcal{P}(n)$ means that a permutation is sampled uniformly and then provided as an oracle (in both the forward and inverse directions) to all parties.

The symbol "$\mathbb{1}$" denotes the identity matrix. The special symbol "$\perp$" denotes reject; if $f(x) = \perp$, it means the output of x is unavailable. The symbol "$||$" denotes the concatenation of bit strings. $|x|$ denotes the length or size of a bit string x;

Lowercase Greek letters are usually used to denote quantum states. Pure states are usually written in Dirac notation, for example, $|\psi\rangle$ and $|\phi\rangle$. Mixed states are usually written as density operators, for example, θ and ρ. Exceptions are δ, ε, and λ, which are usually used to denote some constants or as error terms (usually small).

2.2 Quantum Computing

In this section, we will introduce the fundamental foundations and concepts of quantum computing that are utilized in this thesis.

2.2.1 Quantum Mechanics

In contrast to classical computing, which relies on classical bits with values of 0 or 1, quantum computing employs quantum bits, or *qubits*. These qubits can exist in multiple states simultaneously, a property known as superposition. Consider the 1-qubit states $|0\rangle$ and $|1\rangle$, both can be represented as two unit vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad , |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad .$$

In general, a single qubit can be in any superposition $\alpha_0|0\rangle + \alpha_1|1\rangle$ with $|\alpha_0|^2 + |\alpha_1|^2 = 1$.

For systems with more than one qubit, consider a physical system that can be in N mutually exclusive classical states; the quantum state $|\phi\rangle$ can be written as a superposition of all those classical states:

$$|\phi\rangle = \sum_{x \in N} \alpha_x |x\rangle, \tag{2.1}$$

with $\sum_x |\alpha_x|^2 = 1$. α_x is the *amplitude* of $|x\rangle$. Moreover, the states $|0\rangle, \cdots, |N-1\rangle$ form an orthogonal basis of an N-dimensional Hilbert space, also known as the *computational (or standard) basis*.

If $N = 2^n$, then state $|\phi\rangle$ in Equation 2.1 can be considered as a $n-$qubit quantum state

with 2^n basic states, each of the form $|a_0\rangle \otimes \cdots \otimes |a_{n-1}\rangle$. "$\otimes$" denotes the tensor product, the above can also be simplified to $|a_0, \cdots, a_{n-1}\rangle$.

Pure states and Mixed states. A *pure state* in quantum mechanics refers to the state of a quantum system that can be described by a single, definite quantum state vector in a complex vector space (Hilbert space). Mathematically, a pure state is represented by a ket vector $|\phi\rangle$ in a Hilbert space, as above. A pure state can also be written as $\rho = |\phi\rangle\langle\phi|$, where ρ is the *density matrix* of the state. On the other hand, a *mixed state* can be defined as a probability distribution of pure states. It reflects a situation where the observer does not have complete knowledge of the system and must describe it statistically. Given pure states $|\phi_1\rangle, \cdots, |\phi_m\rangle$ and a probability distribution $\{p_i\}_{i=1}^m$, the density matrix of a mixed state is $\rho = \sum_{i=1}^m p_i|\phi_i\rangle\langle\phi_i|$.

Measurements. When a quantum system is measured, the quantum state "collapses" to one of the possible classical outcomes. For example, if we measure the quantum state in Equation 2.1, the quantum superposition $|\phi\rangle$ will collapse to the classical state $|x\rangle$ with probability $|\alpha_x|^2$. This is known as the Born rule. Such measurement is called *measurement in the computational basis.*

There also exists a more general kind of measurement, called *projective measurement*, which is described by a set of projectors $P_1, ..., P_m$ with $\sum_{i=1}^m P_i = \mathbb{1}$. These projectors are then pairwise orthogonal, meaning that $P_iP_j = 0$ if $i \neq j$. The projector P_i projects on some subspace H_i of the total Hilbert space H , and every state $|\phi\rangle \in H$ can be decomposed in a unique way as $|\phi\rangle = \sum_{i=1}^m |\phi_i\rangle$, with $|\phi_i\rangle = P_i|\phi\rangle \in H_i$. If we apply the projective measurement to the pure state $|\phi\rangle$, we will get outcome i with probability $||\,|\phi_i\rangle||^2 = \text{Tr}(P_i|\phi\rangle\langle\phi|) = \langle\phi|P_i|\phi\rangle$. If the projectors are instead m positive semi-definite matrices that sum to identity, this more general measurement is called a positive-operator valued measure (POVM).

We can also use projective measurement on part of the qubits. For example, given a n-qubit quantum $|\phi\rangle$, we can just measure the first qubit by setting $P_1 = |0\rangle\langle 0| \otimes \mathbb{1}_{2^{n-1}}$ and $P_2 = |1\rangle\langle 1| \otimes \mathbb{1}_{2^{n-1}}$. In this case, the state will collapse to either $|0\rangle|\psi\rangle$ or $|1\rangle|\psi\rangle$, where $|\psi\rangle$ is a $(n-1)$-qubit state , after measurement.

Entanglement. A very important feature of quantum mechanics is the *entanglement*, which refers to quantum correlations between different qubits. Formally, given a state $|\phi\rangle$ in product space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, it is an entangled state if it cannot be written as a tensor product $|\phi_A\rangle \otimes |\phi_B\rangle$, where $|\phi_A\rangle$ is a quantum state in the Hilbert space $\mathcal{H}_A$ and $|\phi_B\rangle$ is a quantum state in $\mathcal{H}_B$. A famous example is the following 2-qubit state

$$|\phi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle,$$

which is also referred to as a *Bell state*. Suppose we only measure the first qubit. The state will collapse to $|00\rangle$ if we get outcome 0, and $|11\rangle$ if we get outcome 1. Therefore, the final state of the second qubit is dependent on what the final state of the first qubit turns out to be. This is an entangled state.

2.2.2 Quantum Circuits and Queries

Quantum Circuits. A quantum circuit is analogous to a classical boolean circuit in the sense that it operates on quantum registers, which represent a collection of qubits, instead of operating on classical registers. Moreover, the boolean gates (AND, OR, NOT, etc.) are replaced by elementary quantum gates. Quantum gates are represented by unitary operators. A unitary operator is a linear transformation that acts on quantum states, usually denoted by matrix U. It has several

properties. First, a unitary operator must preserve the norm of vectors, and it has to be a unitary transform. For example, given a state $|\phi\rangle = \sum_i \alpha_i |i\rangle$, after applying a unitary operator U we can get

$$|\psi\rangle = U|\phi\rangle = U \sum_i \alpha_i |i\rangle = \sum_i \beta_i |i\rangle,$$

where we must have $\sum_i |\beta^i|^2 = 1$. Moreover, U always has an inverse, and its inverse equals its conjugate transpose, i.e., $U^{-1} = U^*$. This also implies that any non-measuring operation on quantum states must be reversible by applying its inverse gate.

A quantum gate is a unitary transformation on a small number of qubits. Here are some fundamental quantum gates, the identity ($\mathbb{1}$) and the pauli gates (X, Y, Z):

$$\mathbb{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}.$$

There are also some quantum gates that are especially important and have been used in many quantum algorithms.

Hadamard gate (H) and phase gate (S):

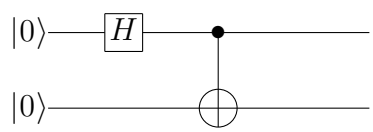
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}.$$

CNOT gate and SWAP gate:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A set of quantum gates is called *universal* if it can be used to approximate any unitary transformation. One example of a set of the universal gates is $\{$H, S and CNOT$\}$.

Here is an example of a quantum circuit. Given a 2-qubit state $|00\rangle$, to get a Bell state [31] $|\phi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, We can construct the following quantum circuit:



Quantum Oracles and Quantum Query Model.

Similar to the classical setting, a quantum oracle is a black-box quantum operation that performs a specific task or computes a specific function. Quantum oracles are often associated with quantum queries, which are used in many quantum algorithms. To explain the quantum query model, we consider a function $f : \{0,1\}^n \to \{0,1\}^m$. A quantum oracle O_f is the following unitary transformation:

$$O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle,$$

where $|x\rangle$ and $|y\rangle$ represent the states of the input and output registers, respectively. It's worth noting that while the input x is classical, a quantum computer can apply O_f to a superposition

of various x, a capability beyond classical computing. As a result, a quantum algorithm can apply O_f on a superposition of basis states, granting simultaneous access to all input bits. Each invocation of such a quantum oracle is termed a quantum query.

Quantum algorithms typically interact with oracles and their own internal unitaries in an interleaved manner to harness the strengths of each in solving specific computational problems. Consider a T-query quantum algorithm starting with an initial state, ϕ_0 (often an all-zero state), and then alternately applying unitary operators $U_0, \cdots, U_T$, and quantum oracle O_f. In addition to input and output registers, a quantum algorithm includes a workspace register for interacting with unitary operators. We can extend O_f by tensoring it with the identity operation on the workspace register.

$$O_f : |x\rangle|y\rangle|w\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle|w\rangle.$$

The final state of the algorithm after T quantum queries is as follows:

$$U_T O_f U_{T-1} O_f \cdots U_1 O_f U_0 |\phi_0\rangle.$$

The algorithm produces its output by measuring the final state. The quantum query model is employed in various quantum algorithms, including Grover's algorithm. Moreover, a growing body of research is currently dedicated to investigating a 'hybrid' model, wherein algorithms have access to both classical and quantum oracles.

Quantum Query Complexity. As previously demonstrated, the quantum query model finds application in numerous quantum algorithms. To assess the performance of these algorithms, it

is crucial to quantify the number of queries needed to complete the task. Consider the following problem: given a set of N elements forming a set $X = \{x_1, x_2, \cdots, x_N\}$ and a boolean function $f : X \to \{0, 1\}$, find an element $x^* \in X$ such that $f(x^*) = 1$. This problem is known as the unstructured search problem. The quantum *query complexity* of this problem is the *minimum* number of queries needed to find x^* for any quantum algorithm that aims to solve this problem.

It is noteworthy that several quantum algorithms, including Shor [12] and Grover [21], provide upper bounds on problem-solving efficiency. The objective is to present effective attacks showcasing the advantages of quantum algorithms. Conversely, lower bound results are typically employed to demonstrate limitations in solving problems. For instance, Boyer, Brassard, Høyer, and Tapp [32] established the optimality of Grover's algorithm by providing a lower bound for solving the unstructured problem. Lower bound results are commonly utilized to prove security. In this thesis, many of our results are accompanied by lower bounds.

2.3 Classical Cryptography

In this section, we recall some basic concepts and terminologies of classical cryptography necessary for this thesis. Firstly, it's important to note that all the results in this thesis are post-quantum results. Therefore, for all classical security notions involving probabilistic polynomial time (PPT) algorithms, we adapt them using quantum polynomial time (QPT) algorithms.

2.3.1 Security Notions and Proof Methods

Negligibility

Definition 2.1. A negligible function is one that is asymptotically smaller than any inverse poly-

nomial function. Formally, a function f from the natural numbers to the non-negative real numbers is considered negligible if, for every positive polynomial p, there exists an integer N such that for all integers $n \geq N$, it holds that $f(n) \leq \frac{1}{p(n)}$.

We use $\mathsf{negl}(n)$ to denote an arbitrary negligible function.

Computational Indistinguishability

Two probability distributions are computationally indistinguishable if no efficient algorithm can distinguish them. Formally,

Definition 2.2. Two sets of distributions $\mathcal{X} = \{X_N\}_{n\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_N\}_{n\in\mathbb{N}}$ are considered computationally indistinguishable if for every QPT distinguisher D, there exists a negligible function $\mathsf{negl}(n)$ such that:

$$\left| \Pr_{x \leftarrow X_n}[D(1^n, x) = 1] - \Pr_{y \leftarrow Y_n}[D(1^n, y) = 1] \right| \leq \mathsf{negl}(n)$$

Computational Security

In Chapter 1, we discussed perfect secrecy and why it is not practical. While perfect secrecy requires zero leakage even against an adversary with unlimited computational power, it is the model we desire. However, for all practical applications, it is unnecessarily strong. In such cases, an encryption scheme would still be considered secure if it leaked only a *small* amount of information to an attacker with only *bounded* computational power. Security definitions under these conditions are called *computational*, to distinguish them from *information-theoretic* notions like perfect secrecy. Computational security is now the prevailing method for defining security for many cryptographic purposes.

Formally, computational security includes two relaxations compared to perfect secrecy.

1. Computational security only holds against bounded (or efficient) adversaries that operate for a limited time; this implies that adversaries could potentially break the security given enough time or unbounded resources.

2. Attackers can succeed, but as long as the successful probability is small enough (usually $\mathsf{negl}(n)$), we wouldn't need to worry about it.

Provable Security

Provable Security is a mathematical approach used to evaluate the security of a cryptosystem. It relies on the adversarial model being considered. Different adversarial models represent various attack scenarios; for example, chosen-plaintext attack (CPA) and chosen-ciphertext attack (CCA) are common ones. It is essential for the chosen adversarial model to accurately reflect the adversary's true activities; otherwise, the proof becomes meaningless. Additionally, the validity of the approach depends on the correctness of the underlying mathematical assumptions. In other words, if someone disproves the correctness of the underlying assumptions, the proof of security becomes compromised.

Since provable security offers strong theoretical guarantees, it can sometimes be impractical for real-world applications, especially when the underlying assumptions are not entirely reliable or applicable. Therefore, the goal of cryptographers is to design cryptosystems that are both robust, meaning provably secure under well-defined mathematical assumptions, and efficient, ensuring computational security in practice.

Reduction Method

We previously discussed various security notions. One immediate question arises: How can we establish the security of a cryptographic construction? A common approach is to assume

that a certain mathematical problem is hard, or that a lower-level cryptographic primitive is secure, and then to prove that the construction based on this problem or primitive is secure under this assumption. Now, let "breaking the construction" be problem A, and let "breaking the problem/primitive" be problem B. The proof is executed by presenting a reduction that transforms any efficient adversary $\mathcal{A}$ that solves problem A into an efficient algorithm $\mathcal{A}'$ that solves problem B. In this way, if we solve A we could solve B, so we say problem A is at least as hard as problem B, denoted as $B \leq A$.

Reduction stands as a fundamental method for establishing security in cryptography, and it plays a central role in the proofs presented in this thesis. Let's consider the cryptographic construction denoted as Π and the corresponding problem/primitive designated as X. In the subsequent paragraph, we will briefly outline the steps involved in the proof process":

1. Fix an efficient adversary $\mathcal{A}$ attacking Π. Let the success probability of the adversary be denoted by $\varepsilon(n)$.

2. Construct an efficient algorithm $\mathcal{A}'$, referred to as the "reduction", that attempts to solve problem X using adversary $\mathcal{A}$ as a subroutine. It is important to note that $\mathcal{A}'$ knows nothing about how $\mathcal{A}$ works; we can consider $\mathcal{A}$ as a black box to $\mathcal{A}'$, which is able to attack Π. Thus, given an input instance x of problem X, algorithm $\mathcal{A}'$ simulates an instance of Π for $\mathcal{A}$ such that:

   (a) As far as $\mathcal{A}$ can tell, it is interacting with Π. That is, the view of $\mathcal{A}$ when run as a subroutine by $\mathcal{A}'$ should be distributed identically to (or at least close to) the view of $\mathcal{A}$ when it interacts with Π itself.

   (b) If $\mathcal{A}$ succeeds in "breaking" the instance of Π that is being simulated by $\mathcal{A}'$, this

should allow $\mathcal{A}'$ to solve the instance x it was given, at least with an inverse polynomial probability of $1/p(n)$.

3. Considering both 2(a) and 2(b) together implies that the algorithm solves X with a probability of $\varepsilon(n)/p(n)$. If $\varepsilon(n)$ is not negligible, then neither is $\varepsilon(n)/p(n)$. Additionally, if $\mathcal{A}$ is efficient, it leads to the existence of an efficient algorithm $\mathcal{A}'$ that solves X with a non-negligible probability. This contradicts the initial assumption.

4. Based on our hardness assumption regarding X, we deduce that no efficient adversary $\mathcal{A}$ can successfully break Π with a non-negligible probability. In other words, Π is computationally secure.

Hybrid Argument

Another method for proving the security of cryptographic schemes is the hybrid argument. This technique is frequently used to demonstrate that certain security properties, typically achieved in an idealized model, are maintained as an adversary interacts with a real-world, practical instantiation of the cryptographic system [33]. The hybrid argument plays a crucial role in bridging the gap between the idealized model and real-world security, instilling confidence in the system's actual security. We will now briefly outline the steps for proof.

1. We start with an idealized model, or "ideal world", where the security is straightforward or easy to prove.

2. The hybrid argument then proceeds by defining a sequence of intermediate hybrids that facilitate the transition from the ideal world to the real world. The goal is to prove that security properties are preserved throughout all transitions. More specifically, we demon-

strate that an adversary's probability of success in breaking the scheme remains bounded in each hybrid. Therefore, the key point of the hybrid argument is to carefully design the intermediate hybrids so that the aforementioned security arguments hold.

3. As the hybrid transitions into the real world, we establish the security by demonstrating that the cryptographic scheme maintains its desired security in the real-world setting.

Formally, to show two distributions D_1 and D_2 are computationally distinguishable, we define a series of hybrid distributions $H_0, H_1, \cdots, H_m$, where $D_1 = H_1$ and $D_2 = H_m$. For any $i \in \{1, \cdots, m-1\}$, define the distinguishing advantage of any QPT algorithm $\mathcal{A}$ as

$$\mathsf{Dist}_{H_i,H_{i+1}}(\mathcal{A}) = \left|\Pr[x \xleftarrow{\$} H_i : \mathcal{A}(x) = 1] - \Pr[x \xleftarrow{\$} H_{i+1} : \mathcal{A}(x) = 1]\right|$$

By triangle inequality, for QPT algorithm A, we have

$$\mathsf{Dist}_{D_1,D_2}(\mathcal{A}) \leq \sum_{i=0}^{m-1} \mathsf{Dist}_{H_i,H_{i+1}}(\mathcal{A}).$$

To prove that D_1 and D_2 cannot be distinguished, the hybrid argument suggests proving that the distinguishing advantage between H_i and H_{i+1} is negligible for all i. In Sections 4.1.2 and 4.2.2, we utilize the hybrid argument to establish our primary theorems.

Game-playing Proof

Another highly effective method for proving the security of a cryptographic protocol is the game-playing proof technique, which is defined in terms of a game played between an adversary and a challenger (representing the protocol). The adversary's objective is to break the security property of that protocol. This game is typically structured into phases or rounds, each represent-

ing a step in the execution of the protocol. During each phase, the challenger interacts with the adversary by performing protocol-related actions such as key generation, message encryption, and decryption. On the other hand, the adversary can take action to try to break the security property, for example, by learning the secret key of a cipher. The actions that the adversary could take depend on the adversarial model we choose.

Bellare and Rogaway [29] first presented a general framework for game-playing proofs. Given two games, G_1 and G_2, which are described above, they define an *identical-till-bad-is-set* scenario. This means that G_1 and G_2 behave identically to each other unless the *bad* event occurs. Therefore, the only difference between G_1 and G_2 is the probability that the *bad* event will occur. This probability determines the distinguishing advantage between the two games.

Game-playing proofs serve as a common approach for formalizing security in cryptographic protocols, offering a structured method for reasoning about the attainable security of the protocol. This technique is extensively employed in our proofs. For instance, Lemma 3.1 and Lemma 3.3 are proven using a game-playing strategy.

2.3.2 Primitives

In this section, we recall some cryptographic primitives that are covered in this thesis.

Pseudorandom Generator

A pseudorandom generator (PRG) G is an efficient, deterministic algorithm for transforming a short, uniform string into a longer, 'pseudorandom' output string. G is a PRG if no efficient distinguisher can detect whether the given string is output by G or a string chosen uniformly at random. The formal definition of a PRG is provided below.

Definition 2.3. Let l be a polynomial number, and let G be a deterministic QPT algorithm such that for any input $s \in \{0,1\}^n$, the algorithm outputs a bit string $G(s) \in \{0,1\}^{l(n)}$. G is a PRG if $l(n) > n$ for every n, and for any QPT distinguishers $\mathcal{D}$, the following holds:

$$\left| \Pr_{s \xleftarrow{\$} \{0,1\}^n} [\mathcal{D}(G(s)) = 1] - \Pr_{r \xleftarrow{\$} \{0,1\}^{l(n)}} [\mathcal{D}(r) = 1] \right| \leq \mathsf{negl}(n).$$

Pseudorandom Function

Pseudorandom functions (PRFs) are a generalization of PRGs. Instead of considering pseudorandom strings, PRF considers pseudorandom (keyed) functions. A keyed function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ is a two-input function, where the first input is the key k. For any fixed key, $F_k(\cdot) = F(k, \cdot)$ represents a function from $\{0,1\}^*$ to $\{0,1\}^*$. We call F a PRF if the function F_k is indistinguishable from a function chosen uniformly random from the set of all functions with the same domain and range. The set of all functions $F : \{0,1\}^n \to \{0,1\}^m$ is denoted as $\mathcal{F}(n,m)$, and we have $|\mathcal{F}(n,m)| = (2^n)^{2^m}$. In the quantum world, F is a Qsecure PRF [34] if the indistinguishability holds against QPT adversaries.

Definition 2.4. (ε-Qsecure PRF) [4, 35] Let $F : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^m$ be a deterministic and efficient keyed function for any finite κ, n, and m. F a ε-Qsecure PRF if for any QPT adversary $\mathcal{A}$ who makes q quantum queries, there exist a negligible function $\varepsilon(\lambda)$ such that

$$\left| \Pr_{k \xleftarrow{\$} \{0,1\}^\kappa} [\mathcal{A}^{F_k(\cdot)}(1^n) = 1] - \Pr_{f \xleftarrow{\$} \mathcal{F}(n,m)} [\mathcal{A}^{f(\cdot)}(1^n) = 1] \right| \leq \varepsilon \cdot \mathsf{poly}(q).$$

Note that $\mathcal{A}^{F_k(\cdot)}$ signifies that the adversary $\mathcal{A}$ has quantum oracle access to F_k, which enables $\mathcal{A}$ to submit superposition queries (inputs) to the quantum oracle O_{F_k}.

Pseudorandom Permutation

A pseudorandom permutation (PRP) is a PRF that is also an invertible permutation on some space for any key. A PRP P is a permutation that is computationally indistinguishable from any permutation R, which is uniformly sampled from the set of all permutations within the same space. We use $\mathcal{P}(n)$ to denote the set of all permutations in space $\{0,1\}^n$. In the quantum setting, the following definition formalizes this concept.

Definition 2.5. (ε-Qsecure PRP) [4, 35] 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 QPT adversary $\mathcal{A}$ who makes q quantum queries, there exist a negligible function $\varepsilon(\lambda)$ such that

$$\left| \Pr_{k \xleftarrow{\$} \{0,1\}^\kappa} \left[\mathcal{A}^{P_k(\cdot), P_k^{-1}(\cdot)}(1^n) = 1 \right] - \Pr_{R \xleftarrow{\$} \mathcal{P}(n)} \left[\mathcal{A}^{R, R^{-1}(\cdot)}(1^n) = 1 \right] \right| \leq \varepsilon \cdot \mathsf{poly}(q) \,.$$

2.3.3 Symmetric Key Encryption

In this section, we will review fundamental concepts and definitions for symmetric (secret) key encryption schemes.

Definition 2.6 (Symmetric-key Encryption Scheme)**.** A symmetric-key encryption scheme is a tuple of QPT algorithms $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where

1. Gen: The key generation algorithm takes as input the security parameter 1^n and generates a random key k.

2. Enc: The encryption algorithm takes the key k and the plaintext m as inputs, outputs a cipherext $c = \mathsf{Enc}_k(m)$.

3. Dec: The decryption algorithm takes the ciphertext c and the key k as inputs, outputs $m = \mathsf{Dec}_k(c)$.

Security Definitions

The definition of security consists of two ingredients: an attack model, which specifics the adversary's power, and a security goal. We start with the security goal. To analyze the security of a symmetric-key encryption scheme, we first have to define security. This can be achieved through *Indistinguishability*.

Indistinguishablity

Indistinguishability is a game-based security definition involving an adversary receiving a ciphertext, which encrypts one of two chosen messages denoted as m_0 and m_1. The adversary's goal is to determine which message is encrypted. The security requirement is that no QPT adversary can successfully determine the message with a probability significantly better than 1/2, equivalent to random guessing. Below, we formally define the indistinguishability experiment involving a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Definition 2.7 (Indistinguishability experiment $\mathsf{INDEXP}_{\mathcal{A},\Pi}$)**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme and $\mathcal{A}$ be a QPT adversary, the experiment proceeds as follows:

1. $\mathcal{A}$ is given input 1^n, and outputs a pair of messages m_0 and m_1 with $|m_0| = |m_1|$.

2. A key k is generated by running $\mathsf{Gen}(1^n)$ and a uniform bit $b \in \{0,1\}^n$ is chosen. $\mathcal{C}$ computes the challenge cipher $c = \mathsf{Enc}_k(m_b)$ and gives to $\mathcal{A}$.

3. $\mathcal{A}$ outputs a bit b'.

4. The experiment outputs 1 if $b' = b$, outputs 0 otherwise. If it outputs 1, we say the experiment succeeds.

In the above experiment, the adversary can only learn the ciphertext, and there is no further interaction between the adversary and the sender or receiver. Next, we define indistinguishability under this model.

Definition 2.8. A symmetric-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable encryptions* in the presence of an adversary if for all QPT adversaries $\mathcal{A}$ there is a negligible function $\mathsf{negl}(n)$ such that for all n,

$$\Pr[\mathsf{INDEXP}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

2.3.4 Cryptographic Techniques

In this section, we will discuss certain cryptographic systems that are relevant to this thesis. It is important to note that in this thesis, all of these cryptosystems are being evaluated under an attack model where the adversary is a QPT algorithm but only has classical oracle access to the keyed primitives.

Message Authentication Codes. A *Message Authentication Code* (MAC) is a cryptographic technique used to verify the integrity and authenticity of a message. It relies on a symmetric key shared between the sender and the receiver. The sender employs this key to generate a MAC, which is transmitted alongside the message. Upon receiving the message, the receiver, using the same key, calculates the MAC on the received message and compares it with the transmitted MAC to verify the message's integrity. A MAC Π consists of three QPT algorithms $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$

such that:

1. Gen takes as input a security parameter 1^n and outputs a key k, written as $k \leftarrow \mathsf{Gen}(1^n)$.

2. Mac takes as input a key k and a message $m \in \{0,1\}^*$, outputs a tag t. We write this as $t \leftarrow \mathsf{Mac}_k(m)$

3. Ver takes as input a key k, a message m and a tag t. It outputs a bit b, where $b = 1$ means valid and $b = 0$ means invalid. This is written as $b = \mathsf{Ver}_k(m, t)$.

For the security of MAC, consider the following experiment:

Definition 2.9 ((The MAC experiment $\mathsf{MAC}_{\mathcal{A},\Pi}$))**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$ be a MAC, $\mathcal{A}$ be a QPT adversary and n be the security parameter, the experiment proceeds as follows:

1. $k \leftarrow \mathsf{Gen}(1^n)$.

2. $\mathcal{A}$ is given input 1^n and classical oracle access to $\mathsf{Mac}_k(\cdot)$. $\mathcal{A}$ outputs (m, t). Let Q denote the set of all queries made by $\mathcal{A}$.

3. The experiment outputs 1 if and only if $\mathsf{Ver}_k(m, t) = 1$ and $m \notin Q$.

Definition 2.10. A MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$ is secure or unforgeable if for all QPT adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(n)$ such that for all n,

$$\Pr[\mathsf{MAC}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Authenticated Encryption with Associated Data. Authenticated encryption (AE) is a cryptographic technique that combines encryption and message authentication to protect data in transit.

AE ensures both confidentiality (through message encryption) and integrity (through message authentication) of the data, preventing unauthorized access and tampering. Authenticated encryption with associated data (AEAD), first formalized by Phillip Rogaway [36], is an AE scheme when additional data (associated data) accompanies the message. Such associated data needs to be authenticated but doesn't necessarily need to be encrypted. Formally, consider the following experiment:

Definition 2.11 ((The AEAD forgery experiment $\mathsf{AEAD}_{\mathcal{A},\Pi}$)). Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key scheme, $\mathcal{A}$ be a QPT adversary, α be the associated data, and n be the security parameter, the experiment proceeds as follows:

1. $k \leftarrow \mathsf{Gen}(1^n)$.

2. $\mathcal{A}$ is given input 1^n, α and classical oracle access to $\mathsf{Enc}_k(\cdot, \alpha)$. $\mathcal{A}$ outputs a ciphertext c. Let Q denote the set of all queries made by $\mathcal{A}$.

3. let $m = \mathsf{Dec}_k(c, \alpha)$. The experiment outputs 1 if and only if $m \neq \perp$ and $m \notin Q$.

Definition 2.12. A symmetric-key scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is unforgeable if for all QPT adversaries $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(n)$ such that for all n,

$$\Pr[\mathsf{AEAD}_{\mathcal{A},\Pi}(n) = 1] \leq \mathsf{negl}(n).$$

Chapter 3: Technical Results

In this chapter, we will cover the technical lemmas needed for our work. We begin with the arbitrary reprogramming lemma, crucial for proving the post-quantum security of the Even-Mansour (EM) cipher [27]. Additionally, we introduce another essential technical lemma used in [27], the resampling lemma. This lemma is then upgraded to a more general but complex version, serving as a key ingredient in proving the post-quantum security of the tweakable Even-Mansour (TEM) cipher [28].

3.1 Arbitrary Reprogramming Lemma

We start with the arbitrary reprogramming lemma, which is a particular extension of the "blinding lemma" of Alagic et al. [37, Theorem 10]. It is a game-based lemma; the goal is to prove that given a distinguisher $\mathcal{D}$, the trace distance between the output state of $\mathcal{D}$ with quantum oracle access to a function F and $\mathcal{D}$ with quantum oracle access to $\bar{F}$, where $\bar{F}$ is a "blinded version" of F, is small unless $\mathcal{D}$ makes a large number of queries. The detailed reprogramming experiment proceeds as follows.

First, a distinguisher $\mathcal{D}$ specifies an arbitrary function F along with a probabilistic algorithm $\mathcal{B}$ which describes how to reprogram F. Specifically, the output of $\mathcal{B}$ is a set of points B_1 at which F may be reprogrammed, along with the values the function should take at those poten-

tially reprogrammed points. Then $\mathcal{D}$ is given quantum access to either F or the reprogrammed version of F, and its goal is to determine which is the case. When $\mathcal{D}$ is done making its oracle queries, it is also given the randomness that was used to run $\mathcal{B}$. Intuitively, the only way $\mathcal{D}$ can tell if its oracle has been reprogrammed is by querying with significant amplitude on some point in B_1. We bound $\mathcal{D}$'s advantage in terms of the probability that any particular value lies in the set B_1 defined by $\mathcal{B}$'s output.

By suitably modifying the proof of Alagic et al. [37, Theorem 10], one can show that the distinguishing probability of $\mathcal{D}$ in the scenario described above is at most $2q \cdot \sqrt{\varepsilon}$, where q is an upper bound on the number of oracle queries and ε is an upper bound on the probability that any given input x is reprogrammed (i.e., that $x \in B_1$). However, that result is only proved for distinguishers with a fixed upper bound on the number of queries they make. To obtain a tighter bound for our application in Section 4.1, we need a version of the result for distinguishers that may *adaptively* choose how many queries they make based on outcomes of intermediate measurements. We recover the aforementioned bound in the case where we now let q denote the number of queries made by $\mathcal{D}$ *in expectation*.

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

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

We are now ready to state and prove our generalized reprogramming lemma.

Lemma 3.1 (Arbitrary Reprogramming Lemma)**.** Let $\mathcal{D}$ be a distinguisher in the following ex-

periment:

Phase 1: $\mathcal{D}$ outputs descriptions of a function $F_0 = F : \{0,1\}^m \to \{0,1\}^n$ and a randomized algorithm $\mathcal{B}$ whose output is a set $B \subset \{0,1\}^m \times \{0,1\}^n$ where each $x \in \{0,1\}^m$ is the first element of at most one tuple in B. Let $B_1 = \{x \mid \exists y : (x,y) \in B\}$ and $\varepsilon = \max_{x \in \{0,1\}^m} \{\Pr_{B \leftarrow \mathcal{B}}[x \in B_1]\}$.

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

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

The name "arbitrary reprogramming" is motivated by the fact that F is arbitrary (and known), and the adversary can reprogram F arbitrarily- so long as some bound on the probability of reprogramming each individual input exists.

Lemma 3.1 allows for distinguishers that choose the number of queries they make adaptively, e.g., depending on the oracle provided and the outcomes of any measurements, and the bound is in terms of the number of queries $\mathcal{D}$ makes *in expectation*. As discussed later in Section 4.1, the ability to directly handle such adaptive distinguishers is necessary for our proof and, to our knowledge, has not been addressed before. To formally reason about adaptive distinguishers, we model the intermediate operations of the distinguisher and the measurements it makes as

quantum channels. With this as our goal, we first recall some necessary background and establish some notation.

Recall that a density matrix ρ is a positive semidefinite matrix with unit trace. A quantum channel—the most general transformation between density matrices allowed by quantum theory—is a completely positive, trace-preserving, linear map. The quantum channel corresponding to the unitary operation U is the map $\rho \mapsto U\rho U^\dagger$. Another type of quantum channel is a *pinching*, which corresponds to the operation of making a measurement. Specializing to the only kind of pinching needed in our proof, consider the measurement of a single-qubit register C given by the projectors $\{\Pi_0, \Pi_1\}$ with $\Pi_b = |b\rangle\langle b|_C$. This corresponds to the pinching $\mathcal{M}_C$ where

$$\mathcal{M}_C(\rho) = \Pi_0\rho\Pi_0 + \Pi_1\rho\Pi_1.$$

Observe that a pinching only produces the post-measurement state, and does not separately give the outcome (i.e., the result 0 or 1).

Consider a quantum algorithm $\mathcal{D}$ with access to an oracle $\mathcal{O}$ operating on registers X, Y (so $\mathcal{O}|x\rangle|y\rangle = |x\rangle|y \oplus \mathcal{O}(x)\rangle$). We define the unitary $c\mathcal{O}$ for the *controlled* version of $\mathcal{O}$, operating on registers C, X, and Y (with C a single-qubit register), as

$$c\mathcal{O}|c\rangle|x\rangle|y\rangle = |c\rangle|x\rangle|y \oplus c \cdot \mathcal{O}(x)\rangle.$$

With this in place, we may now view an execution of $\mathcal{D}^O$ as follows. The algorithm uses registers C, X, Y, and E. Let $q_{\max}$ be an upper bound on the number of queries $\mathcal{D}$ ever makes. Then $\mathcal{D}$

applies the quantum channel

$$(\Phi \circ c\mathcal{O} \circ \mathcal{M}_C)^{q_{\max}} \tag{3.1}$$

to some initial state $\rho = \rho_0^{(0)}$. That is, for each of $q_{\max}$ iterations, $\mathcal{D}$ applies to its current state the pinching $\mathcal{M}_C$ followed by the controlled oracle $c\mathcal{O}$ and then an arbitrary quantum channel Φ (that we take to be the same in all iterations without loss of generality[1]) operating on all its registers. Finally, $\mathcal{D}$ applies a measurement to produce its final output. If we let $\rho_{i-1}^{(0)}$ denote the intermediate state immediately before the pinching is applied in the ith iteration, then $p_{i-1} = \mathrm{Tr}\left[|1\rangle\langle 1|_C\, \rho_{i-1}^{(0)}\right]$ represents the probability that the oracle is applied (or, equivalently, that a query is made) in the ith iteration, and so $q = \sum_{i=1}^{q_{\max}} p_{i-1}$ is the expected number of queries made by $\mathcal{D}$ when interacting with oracle $\mathcal{O}$.

Proof of Lemma 3.1. An execution of $\mathcal{D}$ takes the form of Equation (3.1) up to a final measurement. For some fixed value of the randomness r used to run $\mathcal{B}$, set $\Upsilon_b = \Phi \circ c\mathcal{O}_{F_b} \circ \mathcal{M}_C$, and define

$$\rho_k \stackrel{\text{def}}{=} \left(\Upsilon_1^{q_{\max}-k} \circ \Upsilon_0^k\right)(\rho),$$

so that ρ_k is the final state if the first k queries are answered using a (controlled) F_0 oracle and then the remaining $q_{\max} - k$ queries are answered using a (controlled) F_1 oracle. Furthermore, we define $\rho_i^{(0)} = \Upsilon_0^i(\rho)$. Note also that $\rho_{q_{\max}}$ (resp., ρ_0) is the final state of the algorithm when the F_0 oracle (resp., F_1 oracle) is used the entire time. We bound $\mathbb{E}_r\left[\delta\left(|r\rangle\langle r| \otimes \rho_{q_{\max}},\ |r\rangle\langle r| \otimes \rho_0\right)\right]$, where $\delta(\cdot,\cdot)$ denotes the trace distance.

Define $\tilde{F}^{(B)}(x) = F(x) \oplus F^{(B)}(x)$, and note that $\tilde{F}^{(B)}(x) = 0^n$ for $x \notin B_1$. Since trace

[1] This can be done by having a register serve as a counter that is incremented with each application of Φ.

distance is non-increasing under quantum channels, for any r we have

$$\begin{aligned}\delta\left(|r\rangle\langle r|\otimes\rho_k,\ |r\rangle\langle r|\otimes\rho_{k-1}\right) &\leq \delta\left(c\mathcal{O}_{F_0}\circ\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right),\ c\mathcal{O}_{F_1}\circ\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right)\right)\\ &= \delta\left(\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right),\ c\mathcal{O}_{\tilde{F}^{(B)}}\circ\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right)\right).\end{aligned}$$

By definition of a controlled oracle,

$$\begin{aligned}c\mathcal{O}_{\tilde{F}^{(B)}}\circ\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right) &= c\mathcal{O}_{\tilde{F}^{(B)}}\left(|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\,|1\rangle\langle 1|_C\right)+|0\rangle\langle 0|_C\,\rho_{k-1}^{(0)}\,|0\rangle\langle 0|_C\\ &= \mathcal{O}_{\tilde{F}^{(B)}}\left(|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\,|1\rangle\langle 1|_C\right)+|0\rangle\langle 0|_C\,\rho_{k-1}^{(0)}\,|0\rangle\langle 0|_C,\end{aligned}$$

and thus

$$\begin{aligned}&\delta\left(\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right),\ c\mathcal{O}_{\tilde{F}^{(B)}}\circ\mathcal{M}_C\left(\rho_{k-1}^{(0)}\right)\right)\\ &= \delta\left(|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\,|1\rangle\langle 1|_C,\ \mathcal{O}_{\tilde{F}^{(B)}}\left(|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\,|1\rangle\langle 1|_C\right)\right)\\ &= p_{k-1}\cdot\delta\left(\sigma_{k-1},\ \mathcal{O}_{\tilde{F}^{(B)}}\left(\sigma_{k-1}\right)\right)\end{aligned}$$

where, recall, $p_{k-1}=\mathrm{Tr}\left[|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\right]$ is the probability that a query is made in the kth iteration, and we define the normalized state $\sigma_{k-1}\stackrel{\text{def}}{=}\frac{|1\rangle\langle 1|_C\,\rho_{k-1}^{(0)}\,|1\rangle\langle 1|_C}{p_{k-1}}$. Therefore,

$$\begin{aligned}&\mathbb{E}_r\left[\delta\left(|r\rangle\langle r|\otimes\rho_{q_{\max}},\ |r\rangle\langle r|\otimes\rho_0\right)\right]\\ &\leq \sum_{k=1}^{q_{\max}}\mathbb{E}_B\left[\delta((|r\rangle\langle r|\otimes\rho_k,\ |r\rangle\langle r|\otimes\rho_{k-1})\right]\\ &\leq \sum_{k=1}^{q_{\max}}p_{k-1}\cdot\mathbb{E}_B\left[\delta\left(\sigma_{k-1},\ \mathcal{O}_{\tilde{F}^{(B)}}\left(\sigma_{k-1}\right)\right)\right]\\ &\leq q\cdot\max_{\sigma}\mathbb{E}_B\left[\delta\left(\sigma,\ \mathcal{O}_{\tilde{F}^{(B)}}\left(\sigma\right)\right)\right],\end{aligned}\tag{3.2}$$

where we write $\mathbb{E}_B$ for the expectation over the set B output by $\mathcal{B}$ in place of $\mathbb{E}_r$.

Since σ can be purified to some state $|\psi\rangle$, and $\delta(|\psi\rangle, |\psi'\rangle) \leq \||\psi\rangle - |\psi'\rangle\|_2$ for pure states $|\psi\rangle, |\psi'\rangle$, we have

$$\begin{aligned}
\max_{\sigma} \mathbb{E}_B \left[\delta\left(\sigma,\ \mathcal{O}_{\tilde{F}^{(B)}}(\sigma)\right)\right] &\leq \max_{|\psi\rangle} \mathbb{E}_B \left[\delta\left(|\psi\rangle,\ \mathcal{O}_{\tilde{F}^{(B)}}|\psi\rangle\right)\right] \\
&\leq \max_{|\psi\rangle} \mathbb{E}_B \left[\||\psi\rangle - \mathcal{O}_{\tilde{F}^{(B)}}|\psi\rangle\|_2\right].
\end{aligned}$$

Because $\mathcal{O}_{\tilde{F}^{(B)}}$ acts as the identity on $(\mathbb{I} - \Pi_{B_1})|\psi\rangle$ for any $|\psi\rangle$, we have

$$\begin{aligned}
&\mathbb{E}_B \left[\||\psi\rangle - \mathcal{O}_{\tilde{F}^{(B)}}|\psi\rangle\|_2\right] \\
&= \mathbb{E}_B \left[\|\Pi_{B_1}|\psi\rangle - \mathcal{O}_{\tilde{F}^{(B)}}\Pi_{B_1}|\psi\rangle + (\mathbb{I} - \mathcal{O}_{\tilde{F}^{(B)}})(\mathbb{I} - \Pi_{B_1})|\psi\rangle\|_2\right] \\
&\leq \mathbb{E}_B \left[\|\Pi_{B_1}|\psi\rangle\|_2\right] + \mathbb{E}_B \left[\|\mathcal{O}_{\tilde{F}^{(B)}}\Pi_{B_1}|\psi\rangle\|_2\right] \\
&= 2 \cdot \mathbb{E}_B \left[\|\Pi_{B_1}|\psi\rangle\|_2\right] \\
&\leq 2\sqrt{\mathbb{E}_B \left[\|\Pi_{B_1}|\psi\rangle\|_2^2\right]}, \qquad (3.3)
\end{aligned}$$

using Jensen's inequality in the last step. Let $|\psi\rangle = \sum_{x\in\{0,1\}^m, y\in\{0,1\}^n} \alpha_{x,y}|x\rangle|y\rangle$ where $\||\psi\rangle\|_2^2 = \sum_{x,y} \alpha_{x,y}^2 = 1$. Then

$$\begin{aligned}
\mathbb{E}_B \left[\|\Pi_{B_1}|\psi\rangle\|_2^2\right] &= \mathbb{E}_B \left[\sum_{x,y:\, x\in B_1} \alpha_{x,y}^2\right] \\
&= \sum_{x,y} \alpha_{x,y}^2 \cdot \Pr[x \in B_1] \leq \varepsilon.
\end{aligned}$$

Together with Equations (3.2) and (3.3), this gives the desired result. □

3.2 Resampling Lemmas

In this section, we'll walk through the resampling lemma which is an extension of the "adaptive reprogramming lemma" of Grilo et al. [38]. This is also a game-based lemma, and the goal is also to bound the trace distance between the output state of $\mathcal{D}$ with quantum access to an oracle and $\mathcal{D}$ with its "reprogramming version".

We consider the following experiment: first, a distinguisher $\mathcal{D}$ is given quantum access to an oracle for a random function F; then, in the second stage, F may be "reprogrammed" so its value on a single, uniform point s is changed to an independent, uniform value. Because the distribution of $F(s)$ is the same both before and after any reprogramming, we refer to this as "resampling." The goal for $\mathcal{D}$ is to determine whether or not its oracle was resampled. Intuitively, the only way $\mathcal{D}$ can tell if this is the case—even if it is given s and unbounded access to the oracle in the second stage—is if $\mathcal{D}$ happened to put a large amplitude on s in some query to the oracle in the first stage. We now formalize this intuition.

We begin by establishing notation and recalling a result of Grilo et al. [38]. Given a function $F : \{0,1\}^m \to \{0,1\}^n$ and $s \in \{0,1\}^m$, $y \in \{0,1\}^n$, define the "reprogrammed" function $F_{s\mapsto y} : \{0,1\}^m \to \{0,1\}^n$ as

$$F_{s\mapsto y}(w) = \begin{cases} y & \text{if } w = s \\ F(w) & \text{otherwise.} \end{cases}$$

The following is a special case of [38, Prop. 1]:

Lemma 3.2 (Resampling for random functions)**.** Let $\mathcal{D}$ be a distinguisher in the following exper-

iment:

Phase 1: A uniform $F : \{0,1\}^m \to \{0,1\}^n$ is chosen, and $\mathcal{D}$ is given quantum access to $F_0 = F$.

Phase 2: Uniform $s \in \{0,1\}^m$, $y \in \{0,1\}^n$ are chosen, and we let $F_1 = F_{s \mapsto y}$. A uniform bit b is chosen, and $\mathcal{D}$ is given s and quantum access to F_b. Then $\mathcal{D}$ outputs a guess b'.

For any $\mathcal{D}$ making at most q queries to F_0 in phase 1, it holds that

$$|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0]| \leq 1.5\sqrt{q/2^m}\,.$$

In contrast to the arbitrary reprogramming lemma, the resampling lemma involves a scenario where the distinguisher has quantum oracle access to a random function (isn't arbitrary and known to $\mathcal{D}$) at the first phase and then receives access to either this function or it's "reprogrammed" version at the second phase. Since the reprogramming is restricted to resampling output values from the same distribution used to initially sample outputs of F, only the queries that $\mathcal{D}$ makes in the first phase matter. That's also the reason we call this lemma the "resampling" lemma.

3.2.1 Resampling Lemma for Random Permutations

We extend the above to the case of two-way accessible, random *permutations*. Now, a random permutation $P : \{0,1\}^n \to \{0,1\}^n$ is chosen in the first phase; in the second phase, P may be reprogrammed by swapping the outputs corresponding to two uniform inputs. For

$s_0, s_1 \in \{0,1\}^n$, we define

$$\mathsf{swap}_{s_0,\,s_1}(x) = \begin{cases} s_1 & \text{if } x = s_0 \\ s_0 & \text{if } x = s_1 \\ x & \text{otherwise.} \end{cases}$$

We prove the following lemma:

Lemma 3.3 (Resampling Lemma for random permutations)**.** Let $\mathcal{D}$ be a distinguisher in the following experiment:

Phase 1: A uniform permutation $P : \{0,1\}^n \to \{0,1\}^n$ is chosen, and $\mathcal{D}$ is given quantum access to $P_0 = P$ and $P_0^{-1} = P^{-1}$.

Phase 2: Uniform $s_0, s_1 \in \{0,1\}^n$ are chosen, and we let $P_1 = P \circ \mathsf{swap}_{s_0,\,s_1}$. Uniform $b \in \{0,1\}$ is chosen, and $\mathcal{D}$ is given s_0, s_1, and quantum access to P_b, P_b^{-1}. Then $\mathcal{D}$ outputs a guess b'.

For any $\mathcal{D}$ making at most q queries (combined) to P_0, P_0^{-1} in the first phase,

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

Before we start the proof, we begin by introducing a superposition-oracle technique based on the one by Zhandry [39], but different in that our oracle represents a two-way accessible, uniform permutation (rather than a uniform function). We also do not need to "compress" the oracle, as an inefficient representation suffices for our purposes.

For an arbitrary function $f : \{0,1\}^n \to \{0,1\}^n$, define the state

$$|f\rangle_F = \bigotimes_{x\in\{0,1\}^n} |f(x)\rangle_{F_x},$$

where F is the collection of registers $\{F_x\}_{x\in\{0,1\}^n}$. We represent an evaluation of f via an operator O whose action on the computational basis is given by

$$O_{XYF}\ |x\rangle_X|y\rangle_Y|f\rangle_F = \mathrm{CNOT}^{\otimes n}_{F_x:Y}|x\rangle_X|y\rangle_Y|f\rangle_F = |x\rangle_X|y \oplus f(x)\rangle_Y|f\rangle_F,$$

where X, Y are n-qubit registers. Handling inverse queries to f is more difficult. We want to define an inverse operator O^{inv} such that, for any permutation π,

$$O^{\mathrm{inv}}_{XYF}|\pi\rangle_F = \left(\sum_{x,y\in\{0,1\}^n} |y\rangle\langle y|_Y \otimes \mathsf{X}^x_X \otimes |y\rangle\langle y|_{F_x}\right)|\pi\rangle_F \tag{3.4}$$

(where X is the Pauli-X operator, and for $x \in \{0,1\}^n$ we let $\mathsf{X}^x := \mathsf{X}^{x_1} \otimes \mathsf{X}^{x_2} \otimes \ldots \otimes \mathsf{X}^{x_n}$ so that $\mathsf{X}^x|\hat{x}\rangle = |\hat{x} \oplus x\rangle$); then,

$$O^{\mathrm{inv}}_{XYF}|x\rangle_X|y\rangle_Y|\pi\rangle_F = |x \oplus \pi^{-1}(y)\rangle_X|y\rangle_Y|\pi\rangle_F.$$

In order for O^{inv} to be a well-defined unitary operator, however, we must extend its definition to the entire space of functions. A convenient extension is given by the following action on arbitrary computational basis states:

$$O^{\mathrm{inv}}_{XYF} = \prod_{x'\in\{0,1\}^n} \left(\mathsf{X}^{x'}_X \otimes |y\rangle\langle y|_{F_{x'}} + (\mathbb{1} - |y\rangle\langle y|)_{F_{x'}}\right),$$

so that

$$O^{\text{inv}}_{XYF}|x\rangle_X|y\rangle_Y|f\rangle_F = |x \oplus \left(\oplus_{x':f(x')=y} x'\right)\rangle_X|y\rangle_Y|f\rangle_F.$$

In other words, the inverse operator XORs all preimages (under f) of the value in register Y into the contents of register X.

We may view a uniform permutation as a uniform superposition over all permutations in $\mathcal{P}_n$; i.e., we model a uniform permutation as the state

$$|\phi_0\rangle_F = (2^n!)^{-\frac{1}{2}} \sum_{\pi\in\mathcal{P}_n} |\pi\rangle_F.$$

The final state of any oracle algorithm $\mathcal{D}$ is identically distributed whether we (1) sample uniform $\pi \in \mathcal{P}_n$ and then run $\mathcal{D}$ with access to π and π^{-1}, or (2) run $\mathcal{D}$ with access to O and O^{inv} after initializing the F-registers to $|\phi_0\rangle_F$ (and, if desired, at the end of its execution, measure the F-registers to obtain π and the residual state of $\mathcal{D}$).

Our proof relies on the following lemma, which is a special case of the conclusion of implication $(\diamond')$ in [40]. (Here and in the following, we denote the complementary projector of a projector P by $\bar{P} \stackrel{\text{def}}{=} \mathbb{1} - P$.)

Lemma 3.4 (Gentle measurement lemma)**.** Let $|\psi\rangle$ be a quantum state and let $\{P_i\}_{i=1}^q$ be a collection of projectors with $\left\|\bar{P}_i|\psi\rangle\right\|_2^2 \leq \varepsilon_i$ for all i. Then

$$1 - |\langle\psi|\left(P_q \cdots P_1\right)|\psi\rangle|^2 \leq \sum_{i=1}^q \varepsilon_i.$$

Proof of Lemma 3.3. We split the distinguisher $\mathcal{D}$ into two stages $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ corresponding to the first and second phases of the experiment in Lemma 3.3. As discussed above, we run the

experiment using the superposition oracle $|\phi_0\rangle_F$ and then measure the F-registers at the end. Informally, our goal is to show that on average over the choice of reprogrammed positions s_0, s_1, the adversary-oracle state after $\mathcal{D}_0$ finishes is almost invariant under the reprogramming operation (i.e., the swap of registers F_{s_0} and F_{s_1}) unless $\mathcal{D}_0$ makes a large number of oracle queries. This will follow from Lemma 3.4 because, on average over the choice of s_0, s_1, any particular query of $\mathcal{D}_0$ (whether using O or O^{inv}) only involves F_{s_0} or F_{s_1} with negligible amplitude.

We begin by defining the projectors

$$(P_{s_0 s_1})_X = \begin{cases} \mathbb{1} & s_0 = s_1 \\ \mathbb{1} - |s_0\rangle\langle s_0| - |s_1\rangle\langle s_1| & s_0 \neq s_1 \end{cases}$$

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

It is straightforward to verify that for any s_0, s_1:

$$\left[\mathsf{Swap}_{F_{s_0}F_{s_1}},\ O_{XYF}\left(P_{s_0 s_1}\right)_X\right] = 0 \tag{3.5}$$

$$\left[\mathsf{Swap}_{F_{s_0}F_{s_1}},\ O^{\text{inv}}_{XYF}\left(P^{\text{inv}}_{s_0 s_1}\right)_{FY}\right] = 0, \tag{3.6}$$

where $[\cdot,\cdot]$ denotes the commutator operation, and Swap_{AB} is the swap operator (i.e., $\mathsf{Swap}_{A,B}|x\rangle_A|x'\rangle_B = |x'\rangle_A|x\rangle_B$ if the target registers A, B are distinct, and the identity if A and B refer to the same register). In words, this means that if we project a forward query to inputs other than s_0, s_1, then swapping the outputs of a function at s_0 and s_1 before evaluating that function has no effect; the sane holds if we project an inverse query (for some associated function f) to the set of output

values that are not equal to $f(s_0)$ or $f(s_1)$.

Since $\bar{P}_{s_0 s_1} \stackrel{\text{def}}{=} \mathbb{1} - P_{s_0 s_1} \leq |s_0\rangle\langle s_0| + |s_1\rangle\langle s_1|$ it follows that for any normalized state $|\psi\rangle_{XE}$ (where E is an arbitrary other register),

$$\begin{aligned} \mathop{\mathbb{E}}_{s_0,s_1} \left[\left\| \left(\bar{P}_{s_0 s_1}\right)_X |\psi\rangle_{XE} \right\|_2^2 \right] &\leq \mathop{\mathbb{E}}_{s_0,s_1} \left[\langle\psi| \left(|s_0\rangle\langle s_0| + |s_1\rangle\langle s_1| \right) |\psi\rangle \right] \\ &= 2 \cdot 2^{-n}. \end{aligned} \tag{3.7}$$

We show a similar statement about $P^{\text{inv}}_{s_0 s_1}$. We can express a valid adversary/oracle state $|\psi\rangle_{YXEF}$ (that is thus only supported on the span of $\mathcal{P}_n$) as

$$|\psi\rangle_{YXEF} = \sum_{x,y \in \{0,1\}^n} c_{xy} |y\rangle_Y |y\rangle_{F_x} |\psi_{xy}\rangle_{XEF_{x^c}}, \tag{3.8}$$

for some normalized quantum states $\{|\psi_{xy}\rangle\}_{x,y\in\{0,1\}^n}$, with $\sum_{x,y\in\{0,1\}^n} |c_{xy}|^2 = 1$ and $\langle y|_{F_{x'}} |\psi_{xy}\rangle_{XEF_{x^c}} = 0$ for all $x' \neq x$. If $s_0 = s_1$, then $\left\| \left(\bar{P}^{\text{inv}}_{s_0 s_1}\right)_{YF} |\psi\rangle_{YXEF} \right\|_2^2 = 0 \leq 2 \cdot 2^{-n}$. It is thus immediate from eq. (3.8) that

$$\mathop{\mathbb{E}}_{s_0,s_1} \left[\left\| \left(\bar{P}^{\text{inv}}_{s_0 s_1}\right)_{YF} |\psi\rangle_{YXEF} \right\|_2^2 \right] \leq 2 \cdot 2^{-n} \tag{3.9}$$

Without loss of generality, we assume $\mathcal{D}_0$ starts with initial state $|\psi_0\rangle = |\psi_0'\rangle|\phi_0\rangle$ (which we take to include the superposition oracle's initial state $|\phi_0\rangle$), computes the state

$$|\psi\rangle = U_{\mathcal{D}_0} |\psi_0\rangle = U_q O_q U_{q-1} O_{q-1} \cdots U_1 O_1 |\psi_0\rangle,$$

and outputs all its registers as a state register E. Here, each $O_i \in \{O, O^{\text{inv}}\}$ acts on registers

XYF, and each U_j acts on registers XYE. To each choice of s_0, s_1 we assign a decomposition $|\psi\rangle = |\psi_{\text{good}}(s_0, s_1)\rangle + |\psi_{\text{bad}}(s_0, s_1)\rangle$ by defining

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

where $P^i_{s_0 s_1} = P_{s_0 s_1}$ if $O_i = O$, $P^i_{s_0 s_1} = P^{\text{inv}}_{s_0 s_1}$ if $O_i = O^{\text{inv}}$, and $z \in \mathbb{C}$ is such that $|z| = 1$ and $\langle \psi \mid \psi_{\text{good}}(s_0, s_1)\rangle \in \mathbb{R}_{\geq 0}$.

$$|\psi_{\text{good}}(s_0, s_1)\rangle = z \cdot U_{\mathcal{D}_0} Q^q_{s_0 s_1} \cdots Q^1_{s_0 s_1} |\psi_0\rangle,$$

with $Q^i_{s_0 s_1} = \tilde{U}_i^\dagger P^i_{s_0 s_1} \tilde{U}_i$ for $\tilde{U}_i = U_{i-1} O_{i-1} \ldots U_1 O_1$. Let

$$\varepsilon_i(s_0, s_1) = \left\| \bar{Q}^i_{s_0 s_1} |\psi_0\rangle \right\|_2^2 = \left\| \bar{P}^i_{s_0 s_1} \tilde{U}_i |\psi_0\rangle \right\|_2^2.$$

Applying Lemma 3.4 yields

$$1 - |\langle \psi \mid \psi_{\text{good}}(s_0, s_1)\rangle|^2 \leq \sum_{i=1}^{q} \varepsilon_i(s_0, s_1). \tag{3.10}$$

We will now analyze the impact of reprogramming the superposition oracle after $\mathcal{D}_0$ has finished. Recall that reprogramming swaps the values of the permutation at points s_0 and s_1, which is implemented in the superposition-oracle framework by applying $\mathsf{Swap}_{F_{s_0} F_{s_1}}$. Note that $\mathsf{Swap}_{F_{s_0} F_{s_1}} |\phi_0\rangle = |\phi_0\rangle$. As the adversary's internal unitaries U_i do not act on F, Equa-

tions Eq. (3.5) and Eq. (3.6) then imply that

$$\mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi_{\text{good}}(s_0,s_1)\rangle = |\psi_{\text{good}}(s_0,s_1)\rangle\,.$$

The standard formula for the trace distance of pure states thus yields

$$\frac{1}{2}\left\||\psi\rangle\langle\psi| - \mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}\right\|_1 = \sqrt{1-\left|\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\right|^2}. \tag{3.11}$$

We further have

$$\begin{aligned}\left|\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\right| &= \left|\langle\psi\mid\psi\rangle + \langle\psi_{\text{bad}}(s_0,s_1)|\left(\mathsf{Swap}_{F_{s_0}F_{s_1}} - \mathbb{1}\right)|\psi_{\text{bad}}(s_0,s_1)\rangle\right| \\ &\geq 1 - 2\||\psi_{\text{bad}}(s_0,s_1)\rangle\|_2^2 \end{aligned} \tag{3.12}$$

using the triangle and Cauchy-Schwarz inequalities. Combining Equations Eq. (3.11) and Eq. (3.12) we obtain

$$\frac{1}{2}\left\||\psi\rangle\langle\psi| - \mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}\right\|_1 \leq 2\cdot\||\psi_{\text{bad}}(s_0,s_1)\rangle\|_2.$$

But as $|\psi_{\text{bad}}(s_0,s_1)\rangle = |\psi\rangle - |\psi_{\text{good}}(s_0,s_1)\rangle$, we have

$$\begin{aligned}\||\psi_{\text{bad}}(s_0,s_1)\rangle\|_2^2 &= 2 - 2\cdot\operatorname{Re}\langle\psi\mid\psi_{\text{good}}(s_0,s_1)\rangle \\ &= 2 - 2\cdot|\langle\psi\mid\psi_{\text{good}}(s_0,s_1)\rangle| \\ &\leq 2\sum_{i=1}^{q}\varepsilon_i(s_0,s_1).\end{aligned}$$

Combining the last two equations we obtain

$$\frac{1}{2}\left\| |\psi\rangle\langle\psi| - \mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}\right\|_1 \leq 2\sqrt{2}\sqrt{\sum_{i=1}^{q}\varepsilon_i(s_0,s_1)}\,. \tag{3.13}$$

The remainder of the proof is the same as the analogous part of the proof of [38, Theorem 6]. $\mathcal{D}_1$'s task boils down to distinguishing the states $|\psi\rangle$ and $\mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle$, for uniform s_0, s_1 that $\mathcal{D}_1$ receives as input, using the limited set of instructions allowed by the superposition oracle. We can therefore bound $\mathcal{D}$'s advantage by the maximum distinguishing advantage for these two states when using arbitrary quantum computation, averaged over the choice of s_0, s_1. Using the standard formula for this maximum distinguishing advantage we obtain

$$\begin{aligned}
\Pr\left[\mathcal{D} \text{ outputs } b\right] - \frac{1}{2} &\leq \frac{1}{4}\mathop{\mathbb{E}}_{s_0,s_1}\left[\left\| |\psi\rangle\langle\psi| - \mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle\langle\psi|\mathsf{Swap}_{F_{s_0}F_{s_1}}\right\|_1\right] \\
&\leq \sqrt{2}\mathop{\mathbb{E}}_{s_0,s_1}\left[\sqrt{\sum_{i=1}^{q}\varepsilon_i(s_0,s_1)}\right] \\
&\leq \sqrt{2}\sqrt{\mathop{\mathbb{E}}_{s_0,s_1}\left[\sum_{i=1}^{q}\varepsilon_i(s_0,s_1)\right]} \leq 2\sqrt{\frac{q}{2^n}},
\end{aligned}$$

where the second inequality is Equation Eq. (3.13), the third is Jensen's inequality, and the last is from Equations Eq. (3.7)–Eq. (3.10). This implies the lemma. □

3.2.2 Resampling Lemma with Adaptivity

Lemma 3.3 works for random permutations with both forward and inverse access. However, the resampling points s_0 and s_1 are restricted to be uniformly random. This is enough for being used to prove the post-quantum security of the Even-Mansour cipher, but when it comes

to the tweakable Even-Mansour cipher, the scenario required for the resampling lemma is more complex and the current lemma won't be enough. To handle those tweakable block ciphers, we'll need a new resampling lemma. The new resampling experiment proceeds as follows:

At phase one, a distinguisher $\mathcal{D}$ is first given quantum oracle access to a uniform permutation $P : \{0,1\}^n \to \{0,1\}^n$. Then, two points $s_0, s_1 \in \{0,1\}^n$ are chosen according to some distribution, and in a second phase $\mathcal{D}$ is given access either to the original permutation $P^{(0)} = P$ or a modified permutation $P^{(1)}$ that is the same as P except that the values of $P(s_0)$ and $P(s_1)$ are swapped. (See below for details.) We show, roughly speaking, that so long as the distribution of s_0, s_1 has high min-entropy and $\mathcal{D}$ makes only a bounded number of queries in the first phase of the experiment, $\mathcal{D}$ cannot distinguish those possibilities.

To prove this lemma, we develop a novel permutation variant of the stateful simulation technique for quantum-accessible random oracles, usually referred to as the *superposition oracle* [39]. In this technique, *some* information about the input-output pairs learned by the adversary via quantum queries can be read off directly from the oracle's internal quantum register. In the original superposition oracle technique [39], this useful feature is a consequence of the statistical independence of the function values of a random oracle. Existing generalizations to invertible random permutations lack this feature [27]. We now state the new resampling lemma.

Lemma 3.5. Let $F \subset \mathcal{P}(n)$. Consider the following experiment involving a quantum distinguisher $\mathcal{D}$:

Phase 1: Choose uniform $P \in \mathcal{P}(n)$, and give $\mathcal{D}$ quantum access to P. $\mathcal{D}$ outputs (D, τ), where D is a distribution on $\{0,1\}^n$ and $\tau \in F$.

Phase 2: Sample $\hat{s} \leftarrow D$, set $s_0 = \tau \circ P(\hat{s})$, and choose $s_1 \leftarrow \{0,1\}^n$. Let $P^{(0)} = P$ and define

$$P^{(1)} = P \circ \mathsf{swap}_{s_0,\, s_1}.$$

Let $\varepsilon = 2 \cdot \mathbb{E}_{(D,\tau)\leftarrow\mathcal{D}^P}\left[\max_{x\in\{0,1\}^n} \Pr_{x'\leftarrow D}[x' = x]\right]$. For any $\mathcal{D}$ making at most q queries to P in phase 1,

$$\begin{aligned}&|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0]| \\ &\leq \sqrt{\varepsilon} \cdot \left(1 + \sqrt{q + \log\left(\frac{11|F|}{\sqrt{\varepsilon}}\right)}\right).\end{aligned}$$

Proof. Note that $s_1 = s_0$ then $P^{(0)} = P^{(1)}$. Thus, the distinguishing advantage of $\mathcal{D}$ is upper bounded by its distinguishing advantage conditioned on $s_1 \neq s_0$, and this is what we analyze in the rest of the proof.

Given $s_1 \neq s_0$, let $H \subset \{0,1\}^n$ be a set of size 2^{n-1} containing s_0 but not s_1, and let M be a bijection between H and $\{0,1\}^n \setminus H$ that maps s_0 to s_1. Define

$$\langle x \rangle = \begin{cases} \{x, M(x)\} & \text{if } x \in H \\ \{x, M^{-1}(x)\} & \text{if } x \notin H \end{cases}.$$

We use the plain superposition oracle for permutations as defined, e.g., by Alagic et al. [27] to simulate the permutation P. The resampling experiment with a superposition in place of P acts on quantum registers X (query input), Y (query output), E (adversary memory), and F (the oracle simulation's internal register). The oracle register F is partitioned into 2^n registers F_x, indexed by permutation inputs x. The initial state is

$$|\eta\rangle_F = (2^n!)^{-1/2} \sum_{\pi\in\mathcal{P}(n)} |\pi\rangle_F\,,$$

where $|\pi\rangle_F = \bigotimes_x |\pi(x)\rangle_{F_x}$.

We begin by defining a basis B_M of $\mathbb{C}\mathcal{P}(n) = \mathrm{span}\{|\pi\rangle : \pi \in \mathcal{P}(n)\}$. Define the relation $R_M \subset \mathcal{P}(n) \times \mathcal{P}(n)$ such that

$$(\pi, \sigma) \in R_M \Leftrightarrow \{\pi(x), \pi(M(x))\} = \{\sigma(x), \sigma(M(x))\} \text{ for all } x \in H\,,$$

with the corresponding equivalence classes

$$[\pi]_M = \{\sigma \in \mathcal{P}(n) : (\pi, \sigma) \in R_M\}\,.$$

We denote the set of all equivalence classes by $\mathcal{P}(n)/R_M$. For any $x, x' \in \{0,1\}^n$ and $c \in \{0,1\}$, define the quantum state

$$|\Psi^c_{x,x'}\rangle = \frac{1}{\sqrt{2}}\left(|x\rangle|x'\rangle + (-1)^c|x'\rangle|x\rangle\right)\,.$$

Define $\Gamma_M = \mathcal{P}(n)/R_M \times \{0,1\}^H$. Although Γ_M and the equivalence classes $[\pi]_M$ depend on M, we will sometimes suppress this in the notation.

For each pair $([\pi], y) \in \Gamma$ we define a vector $|([\pi], y)\rangle_F$ as follows. Let π be such that $\pi(x) > \pi(M(x))$ for all $x \in H$, where "$<$" denotes lexicographic order; we call this π the canonical representative of $[\pi]$. Define

$$|([\pi], y)\rangle_F := \bigotimes_{x \in H} \left|\Psi^{y_x}_{\pi(x),\pi(M(x))}\right\rangle_{F_x F_{M(x)}}\,.$$

Observe that if $[\pi] = [\sigma]$ and $y = y'$ then $\langle([\pi], y) \mid ([\sigma], y')\rangle = 1$, and otherwise $\langle([\pi], y) \mid ([\sigma], y')\rangle =$

0. The set

$$B_M = \{|([\pi], y)\rangle : ([\pi], y) \in \Gamma\}$$

is thus an orthonormal set. To see that it forms a basis of $\mathbb{C}\mathcal{P}(n)$, observe that $|B_M| = |\mathcal{P}(n)|$. It follows that any state $|\varphi\rangle_{XYEF}$ can be decomposed as

$$|\varphi\rangle_{XYEF} = \sum_{([\pi],y)\in\Gamma} |\varphi([\pi], y)\rangle_{XYE} \otimes |([\pi], y)\rangle_F \,,$$

where $|\varphi([\pi], y)\rangle$ are subnormalized such that

$$\sum_{([\pi],y)\in\Gamma} \||\varphi([\pi], y)\rangle\|^2 = 1 \,.$$

Define $\Gamma_j = \{([\pi], y) \in \Gamma : |y| \leq j\}$, where $|y|$ denotes Hamming weight.

Claim 3.6. Let $|\phi_q\rangle_{XYEF}$ be the global state after the (unitary part of the) distinguisher has made q queries in phase 1 to a superposition oracle initialized in any state $|\tilde{\tau}\rangle$ such that $\langle([\pi], y) \mid \tilde{\tau}\rangle = 0$ for all $y \neq 0$. Then for all y with $|y| > q$, we have $|\, \phi_q([\pi]_M, y)\,\rangle = 0$.

Proof. We prove the claim by induction on q. The base case $q = 0$ holds by assumption. For the inductive step, say the claim holds for $q - 1$, and recall that

$$|\phi_q\rangle_{XYEF} = U_{XYE} O_{XYF} |\phi_{q-1}\rangle_{XYEF} \,.$$

By the induction hypothesis, we can decompose

$$|\phi_{q-1}\rangle_{XYEF} = \sum_{([\pi],y)\in\Gamma_{q-1}} |\psi_{q-1}([\pi],y)\rangle_{XYE} \otimes |([\pi],y)\rangle_F \,.$$

Using this decomposition and a linearity argument, it suffices to show that for $|y| \leq q-1$, the state $O_{XYF}|x\rangle_X|y\rangle_Y|([\pi],y)\rangle_F$ is supported on basis vectors $|([\pi'],y')\rangle_F$ with $|y'| \leq q$. This follows from the fact that

$$O_{XYF}|x\rangle_X = |x\rangle_X \otimes O^{(x)}_{YF_x} \,.$$

for some operator $O^{(x)}$. This establishes the claim. □

Next, define the projector

$$\Pi_F^{\leq q} := \sum_{([\pi],y)\in\Gamma_q} |([\pi],y)\rangle\langle([\pi],y)|_F$$

and let $\Pi^{\pm} = \frac{1}{2}(\mathbb{1} \pm \mathsf{Swap})$ be the projectors onto the symmetric and antisymmetric subspaces of $\mathbb{C}^{2^n} \otimes \mathbb{C}^{2^n}$.

We will rely on the following claim:

Claim 3.7. For any $m \in \mathbb{N}$ we have

$$\Pr_{\sigma\leftarrow\mathcal{P}(n)}[\exists\tau \in F, S \subset \{0,1\}^n \; \forall x \in S : |S| = m \wedge \tau\circ\sigma(x) \in \langle x\rangle] \leq 11 \cdot 2^{-m} \cdot |F| \,,$$

Proof. For fixed $\tau \in F$ and $S \subset \{0,1\}^n$ of size m, the number of permutations P for which

$P(x) \in \langle x \rangle$ for all $x \in S$ is at most $2^m \cdot (2^n - m)!$. Thus,

$$\Pr_{\sigma \leftarrow \mathcal{P}(n)} [\forall x \in S : \tau \circ \sigma(x) \in \langle x \rangle] \leq 2^m \frac{(2^n - m)!}{2^n!} .$$

A union bound over all τ and S yields

$$\Pr_{\sigma \leftarrow \mathcal{P}(n)} [\exists \tau \in F, S \subset \{0,1\}^n \text{ with } |S| = m \; \forall x \in S : \tau \circ \sigma(x) \in \langle x \rangle] \leq \frac{|F| 2^m}{m!} .$$

Using $11m! \geq 4^m$ proves the claim.

□

We now return to the proof of Lemma 3.5. Let $\Sigma_F^{\leq m}$ be the projector onto the subspace of $\mathbb{C}\mathcal{P}(n)$ spanned by the permutations π such that

$$|\{x \in \{0,1\}^n \mid \forall \tau \in F : \tau \circ \pi(x) \in \langle x \rangle\}| \leq m .$$

The claim implies

$$\left\| |\eta\rangle - \frac{1}{\sqrt{\|\Sigma_F^{\leq m} |\eta\rangle\|}} \Sigma_F^{\leq m} |\eta\rangle \right\| \leq 2 \cdot \sqrt{11 \cdot 2^{-m} |F|} .$$

Note that $\Pi^{\leq 0} \Sigma^{\leq m} |\eta\rangle = \Sigma^{\leq m} |\eta\rangle$. We analyze the resampling experiment where the random permutation is replaced by a superposition oracle initialized with $\frac{1}{\sqrt{\|\Sigma_F^{\leq m} |\eta\rangle\|}} \Sigma_F^{\leq m} |\eta\rangle_F$.

Let $|\psi\rangle_{XYEF}$ denote the global state after phase 1, conditioned on a particular pair (D, τ) output by the distinguisher. As in [38], we can relax the task of the distinguisher as follows: instead of merely providing access to an oracle interface acting on $|\psi\rangle_{XYEF}$ for $b = 0$ and

$\mathsf{Swap}_{F_{s_0}F_{s_1}}|\psi\rangle_{XYEF}$ for $b = 1$, we give the distinguisher arbitrary access to all registers; the distinguisher's task is then to distinguish those quantum states.

For $x \in \{0,1\}^n$, define the projector $Q^{\langle x\rangle} = \sum_{y\in\langle x\rangle} |y\rangle\langle y|$. In the following, z is a variable that corresponds to the result of measuring $F_{\hat{s}}$, i.e., $\tau(z) = s_0$. Setting

$$\Pi_{\psi,\hat{s},z} = \frac{1}{\||z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\|^2}|z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle\langle\psi|_{XYEF}|z\rangle\langle z|_{F_{\hat{s}}}\,,$$

it follows that

$$\begin{aligned}
&2\Pr[b = b' \mid (D,H,M), s_0] - 1\\
&\leq \frac{1}{2}\left\|\Pi_{\psi,\hat{s},z} - \mathsf{Swap}_{F_{\langle\tau(z)\rangle}}\Pi_{\psi,\hat{s},z}\mathsf{Swap}_{F_{\langle\tau(z)\rangle}}\right\|_1\\
&= \frac{1}{2}\left\|\Pi_{\psi,\hat{s},z}\left(\mathbb{1} - \mathsf{Swap}\right)_{F_{\langle\tau(z)\rangle}} + \left(\mathbb{1} - \mathsf{Swap}\right)_{F_{\langle\tau(z)\rangle}}\Pi_{\psi,\hat{s},z}\mathsf{Swap}_{F_{\langle\tau(z)\rangle}}\right\|_1\\
&\leq \left\|\Pi_{\psi,\hat{s},z}\Pi^-_{F_{\langle\tau(z)\rangle}}\right\|_1 + \left\|\Pi^-_{F_{\langle\tau(z)\rangle}}\Pi_{\psi,\hat{s},z}\mathsf{Swap}_{F_{\langle\tau(z)\rangle}}\right\|_1\\
&= \frac{2}{\||z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\|}\left\|\Pi^-_{F_{\langle\tau(z)\rangle}}|z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\right\|_2.
\end{aligned}$$

(The second inequality is the triangle inequality.) Taking the expectation over $\hat{s} \leftarrow D$ and z, we get

$$\begin{aligned}
&2\Pr[b = b' \mid (D,H,M)] - 1\\
&\leq 2\mathbb{E}_{\hat{s},z}\frac{1}{\||z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\|}\left\|\Pi^-_{F_{\langle\tau(z)\rangle}}|z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\right\|_2\\
&\leq 2\sqrt{\mathbb{E}_{\hat{s},z}\frac{1}{\||z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\|}\left\|\Pi^-_{F_{\langle\tau(z)\rangle}}|z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\right\|^2}\\
&= 2\sqrt{\sum_{\hat{s},z} D(\hat{s})\left\|\Pi^-_{F_{\langle\tau(z)\rangle}}|z\rangle\langle z|_{F_{\hat{s}}}|\psi\rangle_{XYEF}\right\|^2}\,,
\end{aligned} \tag{3.14}$$

where the first inequality is Jensen's inequality.

It remains to prove the following claim:

Claim 3.8. For any pair (D, τ) and any normalized state $|\varphi\rangle_{XYEF}$ such that

$$\Pi_{\overline{F}}^{\leq q}|\varphi\rangle_{XYEF} = |\varphi\rangle_{XYEF} \text{ and } \Sigma_{\overline{F}}^{\leq m}|\varphi\rangle_{XYEF} = |\varphi\rangle_{XYEF}\,,$$

we have

$$\sum_{\hat{s},z} D(\hat{s}) \left\| \Pi^{-}_{F_{\langle\tau(z)\rangle}} |z\rangle\langle z|_{F_{\hat{s}}} |\psi\rangle_{XYEF} \right\|^2 \leq (m+q)\varepsilon_D\,.$$

Proof. Observe that

$$\Pi^{-}\left|\Psi^{0}_{\pi(x),\pi(M(x))}\right\rangle = 0 \text{ and } \Pi^{-}\left|\Psi^{1}_{\pi(x),\pi(M(x))}\right\rangle = \left|\Psi^{1}_{\pi(x),\pi(M(x))}\right\rangle$$

for all x and all canonical representatives π. It follows that

$$\Pi^{-}_{F_{s_0}F_{s_1}}|\varphi\rangle_{XYEF} = \sum_{\substack{([\pi],y)\in\Gamma_q:\\ y_{s_0}=1}} |\varphi([\pi],y)\rangle_{XYE} \otimes |([\pi],y)\rangle_F\,.$$

We can now bound

$$\begin{aligned}
&\sum_{\hat{s},z} D(\hat{s}) \left\| \Pi^{-}_{F_{\langle\tau(z)\rangle}} |z\rangle\langle z|_{F_{\hat{s}}} |\psi\rangle_{XYEF} \right\|^2 \\
&\leq \sum_{\hat{s}} \sum_{z:\hat{s}\in\langle\hat{\tau}(z)\rangle} D(\hat{s}) \left\| |z\rangle\langle z|_{F_{\hat{s}}} |\psi\rangle_{XYEF} \right\|^2 \\
&+ \sum_{\hat{s}} \sum_{z:\hat{s}\notin\langle\hat{\tau}(z)\rangle} D(\hat{s}) \left\| \left(\Pi^{-}_{F_{\langle\tau(z)\rangle}} \otimes |z\rangle\langle z|_{F_{\hat{s}}}\right) |\psi\rangle_{XYEF} \right\|^2\,.
\end{aligned}$$

We bound the two terms separately, beginning with the second. We decompose

$$|\psi\rangle_{XYEF} = \sum_{([\pi],y)\in\Gamma_q} |\psi([\pi],y)\rangle_{XYE} \otimes |([\pi],y)\rangle_F$$

and denote the only element of $\langle x\rangle \cap H$ by $\tilde{x}$. We have

$$\begin{aligned}
&\sum_{\hat{s}} \sum_{z:\hat{s}\notin\langle\hat{\tau}(z)\rangle} D(\hat{s}) \left\| \left(\Pi^{-}_{F_{\langle\tau(z)\rangle}} \otimes |z\rangle\langle z|_{F_{\hat{s}}}\right) |\psi\rangle_{XYEF} \right\|^2 \\
&= \sum_{\hat{s}} \sum_{z:\hat{s}\notin\langle\hat{\tau}(z)\rangle} D(\hat{s}) \sum_{([\pi],y)\in\Gamma_q} \left\| \left(\Pi^{-}_{F_{\langle\tau(z)\rangle}} \otimes |z\rangle\langle z|_{F_{\hat{s}}}\right) |\psi([\pi],y)\rangle_{XYE} \otimes |([\pi],y)\rangle_F \right\|^2 \\
&= \sum_{([\pi],y)\in\Gamma_q} \sum_{\substack{\hat{s}\notin\langle\tau\circ\pi(x)\rangle: \\ y_{\pi(\tilde{x})}=1}} D(\hat{s}) \left\| |\psi([\pi],y)\rangle_{XYE} \right\|^2 \\
&\leq \sum_{([\pi],y)\in\Gamma_q} q\varepsilon_D \left\| |\psi([\pi],y)\rangle_{XYE} \right\|^2 \;=\; q\cdot\varepsilon_D.
\end{aligned}$$

For the first term, we have $\Sigma_F^{\leq m}|\varphi\rangle_{XYEF} = |\varphi\rangle_{XYEF}$, i.e., for any permutation π in the support of this state there are at most m values x such that $\tau\circ\pi(x) \in \langle x\rangle$. For the second term, we have $\Sigma_F^{\leq m}|\varphi\rangle_{XYEF} = |\varphi\rangle_{XYEF}$, i.e., $|\varphi\rangle$ is supported on basis states $|[\pi],y\rangle$ where π has at most m fixed points. Using essentially the same chain of inequalities as for the second term, we get

$$\sum_{\hat{s}} \sum_{z:\hat{s}\in\langle\hat{\tau}(z)\rangle} D(\hat{s}) \left\| |z\rangle\langle z|_{F_{\hat{s}}} |\psi\rangle_{XYEF} \right\|^2 \leq m\varepsilon_D\,.$$

This completes the proof. □

Combining the above claim with Eq. (3.14), taking the expectation over (D,τ), and apply-

ing Jensen's inequality one more time results in the bound

$$|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0]| \leq \sqrt{(q+m)\varepsilon}$$

for the modified resampling experiment and thus

$$|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0]| \leq \sqrt{(q+m)\varepsilon} + 11 \cdot 2^{-m}|F|\,.$$

Setting $m = \log\left(\frac{11|F|}{\sqrt{\varepsilon}}\right)$ we get

$$\begin{aligned}&|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0]| \\ &\qquad\qquad \leq \sqrt{\varepsilon}\left(1 + \sqrt{q + \log\left(11\frac{|F|}{\sqrt{\varepsilon}}\right)}\right)\,,\end{aligned}$$

matching the lemma. □

Chapter 4: Post-Quantum Security of Even-Mansour Constructions

4.1 Even-Mansour Cipher

4.1.1 Overview

The Even-Mansour cipher [41] is a well-known approach for constructing a block cipher E from a public random permutation $P : \{0,1\}^n \to \{0,1\}^n$. Figure 4.1 illustrates the construction of the Even-Mansour cipher. The cipher $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ is defined as

$$E_{k_1,k_2}(x) = P(x \oplus k_1) \oplus k_2$$

where, at least in the original construction, k_1, k_2 are uniform and independent. Security in the standard (classical) setting is well understood [41, 42]: roughly, an unbounded attacker with access to P and P^{-1} cannot distinguish whether it is interacting with E_{k_1,k_2} and $E^{-1}_{k_1,k_2}$ (for uniform k_1, k_2) or R and R^{-1} (for an independent, random permutation R) unless it makes $\approx 2^{n/2}$ queries to its oracles. The variant where k_1 is uniform and $k_2 = k_1$ has the same security [42]. These bounds are tight, and key-recovery attacks using $O(2^{n/2})$ queries are known [41, 42].

Unfortunately, the Even-Mansour construction is insecure against a fully quantum attack in which the attacker is given *quantum* access to all its oracles [43, 44]. In such a setting, the

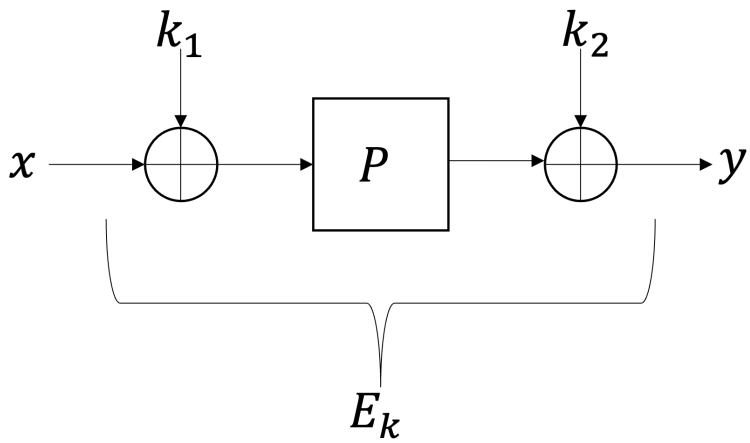


Figure 4.1: Depiction of the Even-Mansour Cipher

adversary can evaluate the unitary operators

$$U_P : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus P(x)\rangle$$

$$U_{E_{k_1,k_2}} : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus E_{k_1,k_2}(x)\rangle$$

(and the analogous unitaries for P^{-1} and $E^{-1}_{k_1,k_2}$) on any quantum state it prepares, which means the adversary is also able to evaluate the unitary operator

$$U_F : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus F(x)\rangle,$$

where $F(x) = E_{k_1,k_2}(x) \oplus P(x)$. Since

$$E_{k_1,k_2}(x) \oplus P(x) = P(x \oplus k_1) \oplus P(x) \oplus k_2,$$

we know that $F(x)$ is a periodic function with $F(x) = F(x \oplus k_1)$. Then Simon's algorithm [14] can be applied to $F(x)$ to give a key-recovery attack using only $O(n)$ queries.

To place this seemingly devastating attack in context, it is worth recalling the original motivation for considering unitary oracles of the form above in quantum-query complexity: one can always transform a classical circuit for a function f into a reversible (and hence unitary) quantum circuit for U_f. In a cryptographic context, it is thus reasonable (indeed, necessary) to consider adversaries that use U_f whenever f is a function whose circuit they know. On the other hand, if the circuit for f is *not* known to the adversary, then there is no mechanism by which it can implement U_f on its own. In particular, if f involves a private key, then the only way an adversary could possibly obtain quantum access to f would be if there were an explicit interface granting such access. In most (if not all) real-world applications, however, the honest parties using the keyed function f would implement f using a classical computer. In fact, even if they were to implement f on a quantum computer, there is no reason for them to support anything but a classical interface to f. In such cases, an adversary would have no way to evaluate the unitary operator corresponding to f.

In most real-world applications of Even-Mansour, therefore, an attacker would have only *classical* access to the keyed permutation E_{k_1,k_2} and its inverse, while retaining quantum access to P and P^{-1}. In particular, this seems to be the "right" attack model for most applications of the resulting block cipher, e.g., for constructing a secure encryption scheme from the cipher using some mode of operation. The setting in which the attacker is given quantum access to public primitives but only classical access to keyed primitives is sometimes called the "Q1 setting" [22]; we will refer to it simply as the *post-quantum* setting. The security of the Even-Mansour cipher in this setting is currently unclear. Kuwakado and Morii [43] show a key-recovery attack using the BHT collision-finding algorithm [45] that requires only $\approx 2^{n/3}$ oracle queries. Their attack uses exponential memory but this was improved in subsequent work [20, 22], culminating in

an attack using the same number of queries but with polynomial memory complexity. While these results demonstrate that the Even-Mansour construction is *quantitatively* less secure in the post-quantum setting than in the classical setting, they do not answer the *qualitative* question of whether the Even-Mansour construction remains secure as a block cipher in the post-quantum setting, or whether attacks using polynomially many queries might be possible.

In a recent work, Jaeger et al. [46] prove security of a forward-only variant of the Even-Mansour construction, as well as for the full Even-Mansour cipher against *non-adaptive* adversaries who make all their classical queries before any quantum queries. They explicitly leave open the question of proving adaptive security in the latter case.

4.1.2 Post-quantum Security of Even-Mansour

Recall that the Even-Mansour cipher is defined as $E_k(x) := P(x \oplus k_1) \oplus k_2$, where $P : \{0,1\}^n \to \{0,1\}^n$ is a public random permutation and $k = (k_1, k_2) \in \{0,1\}^{2n}$ is a key. Our proof assumes only that the marginal distributions of k_1 and k_2 are each uniform. This covers the original Even-Mansour cipher [41] where k is uniform over $\{0,1\}^{2n}$, as well as the one-key variant [42] where k_1 is uniform and then k_2 is set equal to k_1.

For E_k to be efficiently invertible, the permutation P must itself support efficient inversion; that is, the oracle for P must be accessible in both the forward and inverse directions. We thus consider adversaries $\mathcal{A}$ who can access both the cipher E_k and the permutation P in both the forward and inverse directions. The goal of $\mathcal{A}$ is to distinguish this world from the ideal world in which it interacts with independent random permutations R, P. In this section, it will be implicit in our notation that all oracles are two-way accessible.

In the following, we let $\mathcal{P}_n$ be the set of all permutations of $\{0,1\}^n$. We write $E_k[P]$ to denote the Even-Mansour cipher using permutation P and key k; we do this both to emphasize the dependence on P, and to enable references to Even-Mansour with a permutation other than P.

As our main result, we prove a lower bound showing that $\approx 2^{n/3}$ queries are *necessary* for attacking the Even-Mansour cipher in the post-quantum setting. If q_Q denotes the number of (quantum) queries to P, P^{-1} and q_C denotes the number of (classical) queries to $E_{k_1,k_2}, E^{-1}_{k_1,k_2}$, we show that any attack succeeding with constant probability requires either $q_P^2 \cdot q_C = \Omega(2^n)$ or $q_Q \cdot q_C^2 = \Omega(2^n)$ (Equating q_Q and q_C gives the claimed result.). Formally, our main result is as follows:

Theorem 4.1. *Let D be a distribution over $k = (k_1, k_2)$ such that the marginal distributions of k_1 and k_2 are each uniform, and 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*

$$\left| \Pr_{\substack{k \leftarrow D \\ P \leftarrow \mathcal{P}_n}} \left[\mathcal{A}^{E_k[P],P}(1^n) = 1 \right] - \Pr_{R,P \leftarrow \mathcal{P}_n} \left[\mathcal{A}^{R,P}(1^n) = 1 \right] \right| \leq 10 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

A simplified version of the proof also works for the case where P is a random function. We consider the cipher $E_k(x) = P(x \oplus k)$ with k uniform, and $\mathcal{A}$ is given forward-only access to both P and E.

Optimality of the bound. Real-world attackers are usually assumed to make far fewer queries to keyed, "online" primitives than to public, "offline" primitives. (Indeed, while an offline query is just a local computation, an online query requires, e.g., causing an honest user to encrypt a

certain message.) In such a regime, where $q_C \ll q_Q$, the bound on the adversary's advantage in Theorem 4.1 simplifies to $O(q_Q\sqrt{q_C}/2^{n/2})$. In that case $q_Q^2 q_C = \Omega(2^n)$ is necessary for constant success probability, which matches the BHT and offline Simon algorithms [22, 43]. In this sense, our bound is tight with respect to the number of queries. However, it is loose with regard to the attacker's advantage, as both the BHT and offline Simon algorithms achieve advantage $\Theta(q_Q^2 q_C/2^n)$. Reducing this gap is an interesting open question.

Techniques and new technical results. Proving Theorem 4.1 required us to develop new techniques that we believe are interesting beyond our immediate application. We describe the main challenge and its resolution in what follows.

As we have already discussed, in the setting of post-quantum security adversaries may have a combination of classical and quantum oracles. This is the case, in particular, when a post-quantum security notion that involves keyed oracles is analyzed in the quantum random oracle model (QROM), such as when analyzing the Fujisaki-Okamoto transform [39, 47, 48, 49, 50, 51] or the Fiat-Shamir transform [38, 52, 53]. In general, dealing with a mix of quantum and classical oracles presents a problem: quantum-query lower bounds typically begin by "purifying" the adversary and postponing all measurements to the end of its execution, but this does not work if the adversary may decide what query to make to a classical oracle (or even whether to query that oracle at all) *based on the outcome* of an intermediate measurement. The works cited above address this problem in various ways, often by relaxing the problem and allowing quantum access to *all* oracles. This is not an option for us if we wish to prove security, because the Even-Mansour cipher is insecure when the adversary is given quantum access to all its oracles! In the concurrent work of Jaeger et al. [46], the authors overcome the above barrier for the forward-only Even-

Mansour case using Zhandry's compressed oracle technique [39], which is not currently known to be applicable to inverse-accessible permutations.

Instead, we deal with the problem by dividing the execution of an algorithm that has classical access to some oracle O_c and quantum access to another oracle O_q into *stages*, where a stage corresponds to a period between classical queries to O_c. We then analyze the algorithm stage-by-stage. In doing so, however, we introduce another problem: the adversary may adaptively choose the number of queries to O_q in each stage based on outcomes of intermediate measurements. While it is possible to upper bound the number of queries to O_q in each stage by the number of queries made to O_q overall, this will (in general) result in a loose security bound. To avoid such a loss, we use the "blinding lemma" (Lemma 3.1) so that (in addition to some other generalizations) we obtain a bound in terms of the *expected* number of queries made by a distinguisher. In addition to the "blinding lemma", we also need the "resampling lemma" (Lemma 3.3) to handle two-way accessible random permutations in our proof.

Implications for a variant of the Hidden Shift problem. In the well-studied Hidden Shift problem [54], one is asked to find an unknown shift s by querying an oracle for a (typically injective) function f on a group G along with an oracle for the shifted function $f_s(x) = f(x \cdot s)$. If both oracles are classical, this problem has query complexity superpolynomial in $\log |G|$. If both oracles are quantum, then the query complexity is polynomial [16] but the algorithmic difficulty appears to depend critically on the structure of G (e.g., while $G = \mathbb{Z}_2^n$ is easy [14], $G = S_n$ appears to be intractable [55]).

The obvious connection between the Hidden Shift problem and the security of Even-Mansour in general groups has been considered before [55, 56, 57]. In our case, it leads us

to define two natural variants of the Hidden Shift problem:

1. "post-quantum" Hidden Shift: the oracle for f is quantum while the oracle for f_s is classical;

2. "two-sided" Hidden Shift: in place of f_s, use $f_{s_1,s_2}(x) = f(x \cdot s_1) \cdot s_2$; if f is a permutation, grant access to f^{-1} and $f^{-1}_{s_1,s_2}$ as well.

These two variants can be considered jointly or separately, and for either variant, one can consider worst-case or average-case settings [55]. Our main result implies:

Theorem 4.2 (informal)**.** *Solving the post-quantum Hidden Shift problem on any group G requires a number of queries that are superpolynomial in $\log |G|$. This holds for both the one-sided and two-sided versions of the problem, and for both the worst-case and the average-case settings.*

Theorem 4.2 follows from the proof of Theorem 4.1 via a few straightforward observations. First, an inspection of the proof shows that the particular structure of the underlying group (i.e., the XOR operation on $\{0,1\}^n$) is not relevant; the proof works identically for any group, simply replacing 2^n with $|G|$ in the bounds. The two-sided case of Theorem 4.2 then follows almost immediately: worst-case search is at least as hard as average-case search, and average-case search is at least as hard as average-case decision, which is precisely Theorem 4.1 (with the appropriate underlying group). Finally, as noted earlier, an appropriate analog of Theorem 4.1 also holds in the "forward-only" case where $E_k(x) = P(x \oplus k)$ and P is a random function. This yields the one-sided case of Theorem 4.2.

FAEST. FAEST [58] is a digital signature algorithm that is under the most recent NIST Post-quantum Cryptography (PQC) standardization effort. The design of FAEST is intended to provide security against attacks by quantum computers by relying only on information-theoretic and

symmetric-key cryptographic primitives. In particular, in addition to standard PRFs and PRGs for randomness derivation, the security of FAEST is tightly linked to the security of AES128, AES192, and AES256, based on which the NIST security categories 1, 3, and 5 are defined.

A key pair $(\mathsf{pk}, \mathsf{sk})$ for the FAEST signature algorithm theorem is defined as $\mathsf{pk} = (x, y)$ and $\mathsf{sk} = k$ such that $E_k(x) = y$, where E is the block cipher k is a secret key and x is the plaintext block for E. In [58], FAEST instantiates E with two variants: the standard primitive of AES [59] (FAEST) and the Even-Mansour construction (FAEST-EM). FAEST-EM uses a one-way function (OWF) F obtained from a public cryptographic permutation π by adding a key both to the input x of the OWF and to the output of the permutation, i.e.,

$$F_k(x) = k \oplus \pi(x \oplus k),$$

where π is instantiated with AES by fixing a random key k_0 and making k_0 a public constant. FAEST-EM performs better than FAEST, which can be found in [58, Table 2.1].

Clearly, the security of the OWF F is based directly on the security of the Even-Mansour construction. More specifically, the authors use our Theorem 4.1 to prove the following corollary, which shows that single-key Even-Mansour is a post-quantum-secure OWF.

Corollary 4.3 (Corollary 1, [58])**.** Let $\mathcal{A}$ be an adversary making q quantum queries to its oracle. Let π be a uniformly random n-bit permutation, and $x, k \leftarrow \{0,1\}^\lambda$. Then

$$\Pr[\mathcal{A}^\pi(k \oplus \pi(k \oplus x)) = k] \leq c2^{-\lambda/2}(q + 2 + \sqrt{3(q+2)}) + \frac{1}{2^\lambda - 1},$$

where $\mathcal{A}$ has both forward and inverse access to its oracles.

The above corollary shows that $\Omega(2^{\lambda/2})$ are necessary to invert F, which matches the Grover attack.

Proof of Theorem 4.1. Without loss of generality, we assume $\mathcal{A}$ never makes a redundant classical query; that is, once it learns an input/output pair (x, y) by making a query to its classical oracle, it never again submits the query x (respectively, y) to the forward (respectively, inverse) direction of that oracle.

We divide an execution of $\mathcal{A}$ into $q_C + 1$ stages $0, \ldots, q_C$, where the jth stage corresponds to the time between the jth and $(j+1)$st classical queries of $\mathcal{A}$. In particular, the 0th stage corresponds to the period of time before $\mathcal{A}$ makes its first classical query, and the q_Cth stage corresponds to the period of time after $\mathcal{A}$ makes its last classical query. We allow $\mathcal{A}$ to adaptively distribute its q_Q quantum queries between these stages arbitrarily. We let $q_{P,j}$ denote the expected number of queries $\mathcal{A}$ makes in the jth stage in the ideal world $\mathcal{A}^{R,P}$; note that $\sum_{j=0}^{q_C} q_{P,j} = q_Q$.

We denote the ith classical query of $\mathcal{A}$ by (x_i, y_i, b_i), where $b_i = 0$ means that $\mathcal{A}$ queried x_i in the forward direction and received response y_i, and $b_i = 1$ means that $\mathcal{A}$ queried y_i in the inverse direction and received response x_i. Let $T_j = \big((x_1, y_1, b_1), \ldots, (x_j, y_j, b_j)\big)$ be the ordered list describing the first j classical queries made by $\mathcal{A}$. We use "$\prod$" to denote sequential composition of operations, i.e., $\prod_{i=1}^{n} f_i = f_1 \circ \cdots \circ f_n$. (Note that order matters since, in general,

the composition of operators is not commutative.) Recall that $\mathsf{swap}_{a,\,b}$ swaps a and b. Define:

$$\begin{aligned}
\overrightarrow{S}_{T_j,P,k} &\stackrel{\text{def}}{=} \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P(x_i\oplus k_1),\,y_i\oplus k_2} \\
\overrightarrow{Q}_{T_j,P,k} &\stackrel{\text{def}}{=} \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{x_i\oplus k_1,\,P^{-1}(y_i\oplus k_2)} \\
\overleftarrow{S}_{T_j,P,k} &\stackrel{\text{def}}{=} \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P(x_i\oplus k_1,\,y_i\oplus k_2}
\end{aligned}$$

$$\overleftarrow{Q}_{T_j,P,k} \stackrel{\text{def}}{=} \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{x_i\oplus k_1,\,P^{-1}(y_i\oplus k_2)}$$

where, as usual, f^0 is the identity and $f^1 = f$. Finally, define

$$P_{T_j,k} \stackrel{\text{def}}{=} \overleftarrow{S}_{T_j,P,k} \circ P \circ \overrightarrow{Q}_{T_j,P,k}\,. \tag{4.1}$$

Since, for any P, x_1, y_1, x_2, y_2, it holds that

$$\begin{aligned}
\mathsf{swap}_{P(x_1),\,P(y_1)} \circ \mathsf{swap}_{P(x_2),\,P(y_2)} \circ P &= \mathsf{swap}_{P(x_1),\,P(y_1)} \circ P \circ \mathsf{swap}_{x_2,\,y_2} \\
&= P \circ \mathsf{swap}_{x_1,\,y_1} \circ \mathsf{swap}_{x_2,\,y_2},
\end{aligned}$$

we also have

$$P_{T_j,k} = \overleftarrow{S}_{T_j,P,k} \circ \overrightarrow{S}_{T_j,P,k} \circ P = P \circ \overleftarrow{Q}_{T_j,P,k} \circ \overrightarrow{Q}_{T_j,P,k}\,. \tag{4.2}$$

Intuitively, when the $\{x_i\}$ are distinct and the $\{y_i\}$ are distinct, $P_{T_j,k}$ is a "small" modification of P for which $E_k[P_{T_j,k}](x_i) = y_i$ for all i. (Note, however, that this may fail to hold if there is an "internal collision," i.e., $P(x_i \oplus k_1) = y_j \oplus k_2$ for some $i \neq j$. But such collisions occur with

low probability over choice of k_1, k_2.)

We now define a sequence of experiments $\mathbf{H}_j$, for $j = 0, \ldots, q_C$.

Experiment $\mathbf{H}_j$. Sample $R, P \leftarrow \mathcal{P}_n$ and $k \leftarrow D$. Then:

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using P, stopping immediately *before* its $(j+1)$st classical query. Let $T_j = \big((x_1, y_1, b_1), \ldots, (x_j, y_j, b_j)\big)$ be the ordered list of classical queries/answers.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $E_k[P]$ and its quantum queries using $P_{T_j,k}$.

We can compactly represent $\mathbf{H}_j$ as the experiment in which $\mathcal{A}$'s queries are answered using the oracle sequence

$$\underbrace{P, R, P, \cdots, R, P,}_{j \text{ classical queries}} \underbrace{E_k[P], P_{T_j,k}, \cdots, E_k[P], P_{T_j,k}}_{q_C - j \text{ classical queries}} .$$

Each appearance of R or $E_k[P]$ indicates a single classical query. Each appearance of P or $P_{T_j,k}$ indicates a stage during which $\mathcal{A}$ makes multiple (quantum) queries to that oracle but no queries to its classical oracle. Observe that $\mathbf{H}_0$ corresponds to the execution of $\mathcal{A}$ in the real world, i.e., $\mathcal{A}^{E_k[P],P}$, and that $\mathbf{H}_{q_C}$ is the execution of $\mathcal{A}$ in the ideal world, i.e., $\mathcal{A}^{R,P}$.

For $j = 0, \ldots, q_C - 1$, we introduce additional experiments $\mathbf{H}'_j$:

Experiment $\mathbf{H}'_j$. Sample $R, P \leftarrow \mathcal{P}_n$ and $k \leftarrow D$. Then:

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using P, stopping immediately *after* its $(j+1)$st classical query. Let $T_{j+1} = \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$ be the ordered list indicating $\mathcal{A}$'s classical queries/answers.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $E_k[P]$ and its quantum queries using $P_{T_{j+1},k}$.

Thus, $\mathbf{H}'_j$ corresponds to running $\mathcal{A}$ using the oracle sequence

$$\underbrace{P, R, P, \cdots, R, P,}_{j \text{ classical queries}} R, P_{T_{j+1},k}, \underbrace{E_k[P], P_{T_{j+1},k} \cdots, E_k[P], P_{T_{j+1},k}}_{q_C - j - 1 \text{ classical queries}}.$$

In Lemmas 4.4 and 4.5, we establish bounds on the distinguishability of $\mathbf{H}'_j$ and $\mathbf{H}_{j+1}$, as well as $\mathbf{H}_j$ and $\mathbf{H}'_j$. For $0 \leq j < q_C$ these give:

$$\begin{aligned}
\left|\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]\right| &\leq 2 \cdot q_{P,j+1} \cdot \sqrt{\frac{2 \cdot (j+1)}{2^n}}. \\
\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| &\leq 8 \cdot \sqrt{\frac{q_Q}{2^n}} + 2q_C \cdot 2^{-n}
\end{aligned}$$

Using the above, we have

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1]| \\
&\leq \sum_{j=0}^{q_C-1} \left(8 \cdot \sqrt{\frac{q_Q}{2^n}} + 2q_C \cdot 2^{-n} + 2 \cdot q_{P,j+1}\sqrt{\frac{2 \cdot (j+1)}{2^n}}\right) \\
&\leq 2q_C^2 \cdot 2^{-n} + \sum_{j=0}^{q_C-1} \left(8 \cdot \sqrt{\frac{q_Q}{2^n}} + 2 \cdot q_{P,j+1}\sqrt{\frac{2q_C}{2^n}}\right) \\
&\leq 2q_C^2 \cdot 2^{-n} + 2^{-n/2} \cdot \left(8q_C\sqrt{q_Q} + 2 \cdot q_Q\sqrt{2q_C}\right).
\end{aligned}$$

We now simplify the bound further. If $q_Q = 0$, then E_k and R are perfectly indistinguishable and the theorem holds; thus, we may assume $q_Q \geq 1$. We can also assume $q_C < 2^{n/2}$ since otherwise the bound is larger than 1. Under these assumptions, we have $q_C^2 \cdot 2^{-n} \leq q_C \cdot 2^{-n/2} \leq$

$q_C\sqrt{q_Q} \cdot 2^{-n/2}$ and so

$$
\begin{aligned}
&2q_C^2 \cdot 2^{-n} + 2^{-n/2}\left(8q_C\sqrt{q_Q} + 2q_Q\sqrt{2q_C}\right) \\
&\leq 2 \cdot q_C\sqrt{q_Q} \cdot 2^{-n/2} + 2^{-n/2}\left(8q_C\sqrt{q_Q} + 2q_Q\sqrt{2q_C}\right) \\
&\leq 10 \cdot 2^{-n/2}\left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right) ,
\end{aligned}
$$

as claimed. □

To complete the proof of Theorem 4.1, we now show that $\mathbf{H}'_j$ is indistinguishable from to $\mathbf{H}_{j+1}$ and $\mathbf{H}_j$ is indistinguishable from $\mathbf{H}'_j$.

Lemma 4.4. For $j = 0, \ldots, q_C - 1$,

$$
\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{P,j+1}\sqrt{2 \cdot (j+1)/2^n} ,
$$

where $q_{P,j+1}$ 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. Recall we can write the oracle sequences defined by $\mathbf{H}'_j$ and $\mathbf{H}_{j+1}$ as

$$
\begin{aligned}
\mathbf{H}'_j :&\ P, R, P, \cdots, R, P,\ R,\ P_{T_{j+1},k},\ E_k[P], P_{T_{j+1},k}, \cdots, E_k[P], P_{T_{j+1},k} \\
\mathbf{H}_{j+1} :&\ \underbrace{P, R, P, \cdots, R, P}_{j \text{ classical queries}}, R,\ P, \qquad \underbrace{E_k[P], P_{T_{j+1},k}, \cdots, E_k[P], P_{T_{j+1},k}}_{q_C - j - 1 \text{ classical queries}} .
\end{aligned}
$$

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 arbitrary reprogramming experiment from Lemma 3.1:

Phase 1: $\mathcal{D}$ samples $P, R \leftarrow \mathcal{P}_n$. It then runs $\mathcal{A}$, answering its quantum queries using P and

its classical queries using R, until after it responds to $\mathcal{A}$'s $(j+1)$st classical query. Let $T_{j+1} = \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$ be the list of classical queries/answers. $\mathcal{D}$ defines $F(t, x) := P^t(x)$ for $t \in \{1, -1\}$. It also defines the following randomized algorithm $\mathcal{B}$: sample $k \leftarrow D$ and then compute the set B of input/output pairs to be reprogrammed so that $F^{(B)}(t, x) = P^t_{T_{j+1},k}(x)$ for all t, x.

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 $P^t = F_b(t, \cdot)$. Phase 2 ends when $\mathcal{A}$ makes its next (i.e., $(j+2)$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 $E_k[P]$ and its quantum queries using $P_{T_{j+1},k}$. Finally, it outputs whatever $\mathcal{A}$ outputs.

Observe that $\mathcal{D}$ is a valid distinguisher for the reprogramming experiment of Lemma 3.1. 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 $\mathbf{H}'_j$. It follows that $|\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]|$ is equal to the distinguishing advantage of $\mathcal{D}$ in the reprogramming experiment. To bound this quantity using Lemma 3.1, we bound the reprogramming probability ε and the expected number of queries made by $\mathcal{D}$ in phase 2 (when $F = F_0$.)

The reprogramming probability ε can be bounded using the definition of $P_{T_{j+1},k}$ and the fact that $F^{(B)}(t, x) = P^t_{T_{j+1},k}$. Fixing P and T_{j+1}, the probability that any given (t, x) is repro-

grammed is at most the probability (over k) that it is in the set

$$\left\{(1, x_i \oplus k_1), (1, P^{-1}(y_i \oplus k_2)), (-1, P(x_i \oplus k_1)), (-1, y_i \oplus k_2)\right\}_{i=1}^{j+1} .$$

Taking a union bound and using the fact that the marginal distributions of k_1 and k_2 are each uniform, we get $\varepsilon \leq 2(j+1)/2^n$.

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 $(j+1)$st stage in $\mathbf{H}_{j+1}$. Since $\mathbf{H}_{j+1}$ and $\mathbf{H}_{q_C}$ are identical until after the $(j+1)$st stage is complete, this is precisely $q_{P,j+1}$. □

Lemma 4.5. For $j = 0, \ldots, q_C$,

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| \leq 8 \cdot \sqrt{\frac{q_Q}{2^n}} + 2q_C \cdot 2^{-n}.$$

Proof. Recall that we can write the oracle sequences defined by $\mathbf{H}_j$ and $\mathbf{H}'_j$ as

$$\mathbf{H}_j : \; P, R, P, \cdots, R, P, \quad E_k[P], P_{T_j,k}, \quad E_k[P], P_{T_j,k} \quad , \cdots, E_k[P], P_{T_j,k}$$

$$\mathbf{H}'_j : \; \underbrace{P, R, P, \cdots, R, P}_{j \text{ classical queries}}, \; R, \qquad P_{T_{j+1},k}, \underbrace{E_k[P], P_{T_{j+1},k}, \cdots, E_k[P], P_{T_{j+1},k}}_{q_C - j - 1 \text{ classical queries}} .$$

Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j$ and $\mathbf{H}'_j$. We construct from $\mathcal{A}$ a distinguisher $\mathcal{D}$ for the reprogramming experiment of Lemma 3.3:

Phase 1: $\mathcal{D}$ is given quantum access to a permutation P. It samples $R \leftarrow \mathcal{P}_n$ and then runs $\mathcal{A}$, answering its quantum queries with P and its classical queries with R (in the appropriate

directions), until $\mathcal{A}$ submits its $(j+1)$st classical query x_{j+1} in the forward direction[1] (i.e., $b_{j+1} = 0$). Let $T_j = \big((x_1, y_1, b_1), \cdots, (x_j, y_j, b_j)\big)$ be the list of classical queries/answers thus far.

Phase 2: Now $\mathcal{D}$ receives $s_0, s_1 \in \{0,1\}^n$ and quantum oracle access to a permutation P_b. Then $\mathcal{D}$ sets $k_1 := s_0 \oplus x_{j+1}$, chooses $k_2 \leftarrow D_{|k_1}$ (where this represents the conditional distribution on k_2 given k_1), and sets $k := (k_1, k_2)$. $\mathcal{D}$ continues running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)$st one) using $E_k[P_b]$, and its remaining quantum queries using

$$(P_b)_{T_j,k} = \overleftarrow{S}_{T_j,P_b,k} \circ \overrightarrow{S}_{T_j,P_b,k} \circ P_b \,.$$

Finally, $\mathcal{D}$ outputs whatever $\mathcal{A}$ outputs.

Note that although $\mathcal{D}$ makes additional queries to P_b in phase 2 (to determine $P_b(x_1 \oplus k_1), \ldots, P_b(x_j \oplus k_1)$), the bound of Lemma 3.3 only depends on the number of quantum queries $\mathcal{D}$ makes in phase 1, which is at most q_Q.

We now analyze the execution of $\mathcal{D}$ in the two cases of the game of Lemma 3.3: $b = 0$ (no reprogramming) and $b = 1$ (reprogramming). In both cases, P and R are independent, uniform permutations, and $\mathcal{A}$ is run with quantum oracle P and classical oracle R until it makes its $(j+1)$st classical query; thus, through the end of phase 1, the above execution of $\mathcal{A}$ is consistent with both $\mathbf{H}_j$ and $\mathbf{H}'_j$.

At the start of phase 2, uniform $s_0, s_1 \in \{0,1\}^n$ are chosen. Since $\mathcal{D}$ sets $k_1 := s_0 \oplus x_{j+1}$,

[1] We assume for simplicity that this query is in the forward direction, but the case where it is in the inverse direction can be handled entirely symmetrically (using the fact that the marginal distribution of k_2 is uniform). The strings s_0 and s_1 are in that case replaced by $P_b(s_0)$ and $P_b(s_1)$. See Section 44 for details.

the distribution of k_1 is uniform and hence k is distributed according to D. The two cases ($b = 0$ and $b = 1$) now begin to diverge.

Case $b = 0$ (no reprogramming). In this case, $\mathcal{A}$'s remaining classical queries (including its $(j+1)$st classical query) are answered using $E_k[P_0] = E_k[P]$, and its remaining quantum queries are answered using $(P_0)_{T_j,k} = P_{T_j,k}$. The output of $\mathcal{A}$ is thus distributed identically to its output in $\mathbf{H}_j$ in this case.

Case $b = 1$ (reprogramming). In this case, we have

$$P_b = P_1 = P \circ \mathsf{swap}_{s_0,\, s_1} = \mathsf{swap}_{P(s_0),\, P(s_1)} \circ P = \mathsf{swap}_{P(x_{j+1} \oplus k_1),\, P(s_1)} \circ P\,. \tag{4.3}$$

The response to $\mathcal{A}$'s $(j+1)$st classical query is thus

$$y_{j+1} \stackrel{\text{def}}{=} E_k[P_1](x_{j+1}) = P_1(x_{j+1} \oplus k_1) \oplus k_2 = P_1(s_0) \oplus k_2 = P(s_1) \oplus k_2\,. \tag{4.4}$$

The remaining classical queries of $\mathcal{A}$ are then answered using $E_k[P_1]$, while its remaining quantum queries are answered using $(P_1)_{T_j,k}$. If we let Expt_j refer to the experiment in which $\mathcal{D}$ executes $\mathcal{A}$ as a subroutine when $b = 1$, it follows from Lemma 3.3 that

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathsf{Expt}_j) = 1]\right| \leq 4\sqrt{q_Q/2^n}. \tag{4.5}$$

We now define three events:

1. bad_1 is the event that $y_{j+1} \in \{y_1, \ldots, y_j\}$.

2. bad_2 is the event that $s_1 \oplus k_1 \in \{x_1, \ldots, x_j\}$.

3. bad_3 is the event that, in phase 2, $\mathcal{A}$ queries its classical oracle in the forward direction on $s_1 \oplus k_1$, or the inverse direction on $P(s_0) \oplus k_2$ (with result $s_1 \oplus k_1$).

Since $y_{j+1} = P(s_1) \oplus k_2$ is uniform (because k_2 is uniform and independent of P and s_1), it is immediate that $\Pr[\mathsf{bad}_1] \leq j/2^n$. Similarly, $s_1 \oplus k_1 = s_1 \oplus s_0 \oplus x_{j+1}$ is uniform, and so $\Pr[\mathsf{bad}_2] \leq j/2^n$. As for the last event, we have:

Claim 4.6. $\Pr[\mathsf{bad}_3] \leq (q_C - j)/2^n + 4\sqrt{q_Q/2^n}$.

Proof. Consider the algorithm $\mathcal{D}'$ that behaves identically to $\mathcal{D}$ in phases 1 and 2, but then when $\mathcal{A}$ terminates outputs 1 iff event bad_3 occurred. When $b = 0$ (no reprogramming), the execution of $\mathcal{A}$ is independent of s_1, and so the probability that bad_3 occurs is at most $(q_C - j)/2^n$. Now observe that $\mathcal{D}'$ is a distinguisher for the reprogramming game of Lemma 3.3. The claim follows. □

In Figure 4.2, we show code for Expt_j and a related experiment Expt'_j. Note that Expt_j and Expt'_j are identical until either $\mathsf{bad}_1, \mathsf{bad}_2$, or bad_3 occur, and so by the fundamental lemma of game playing[2] [60] we have

$$\begin{aligned} \left|\Pr[\mathcal{A}(\mathsf{Expt}'_j) = 1] - \Pr[\mathcal{A}(\mathsf{Expt}_j) = 1]\right| &\leq \Pr[\mathsf{bad}_1 \vee \mathsf{bad}_2 \vee \mathsf{bad}_3] \\ &\leq 2q_C/2^n + 4\sqrt{q_Q/2^n}\,. \end{aligned} \tag{4.6}$$

We complete the proof by arguing that Expt'_j is identical to $\mathbf{H}'_j$:

1. In Expt'_j, the oracle Q used in line 12 is always equal to $P_{T_{j+1},k}$. When bad_1 or bad_2 occurs this is immediate (since then Q is set to $P_{T_{j+1},k}$ in line 11). But if bad_1 does not occur then

[2] This lemma is an information-theoretic result, and can be applied in our setting since everything we say in what follows holds even if $\mathcal{A}$ is given the entire function table for its quantum oracle Q in line 12.

$P, R \leftarrow \mathcal{P}_n$
Run $\mathcal{A}$ with quantum access to P and classical access to R, until $\mathcal{A}$ makes its $(j+1)$st classical query x_{j+1}; let T_j be as in the text
$s_0, s_1 \leftarrow \{0,1\}^n,\ P_1 := P \circ \mathsf{swap}_{s_0,s_1}$
$k_1 := s_0 \oplus x_{j+1}$, $k_2 \leftarrow D_{|k_1}$, $k := (k_1, k_2)$
$y_{j+1} := E_k[P_1](x_{j+1})$
$Q := (P_1)_{T_j,k}$
if $y_{j+1} \in \{y_1, \ldots, y_j\}$ **then** $\mathsf{bad}_1 := \mathsf{true}$, $\boxed{y_{j+1} \leftarrow \{0,1\}^n \setminus \{y_1, \ldots, y_j\}}$
Give y_{j+1} to $\mathcal{A}$ as the answer to its $(j+1)$st classical query
$T_{j+1} := \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$
if $s_1 \oplus k_1 \in \{x_1, \ldots, x_j\}$ **then** $\mathsf{bad}_2 := \mathsf{true}$
if $\mathsf{bad}_1 = \mathsf{true}$ *or* $\mathsf{bad}_2 = \mathsf{true}$ **then** $\boxed{Q := P_{T_{j+1},k}}$
Continue running $\mathcal{A}$ with quantum access to Q and classical access to $\mathcal{O}/\mathcal{O}^{-1}$

$\underline{\mathcal{O}(x)}$
$y := E_k[P_1](x)$
if $x = s_1 \oplus k_1$ **then**
  $\mathsf{bad}_3 := \mathsf{true}$, $\boxed{y := E_k[P](x)}$
return y

$\underline{\mathcal{O}^{-1}(y)}$
$x := E_k^{-1}[P_1](y)$
if $x = s_1 \oplus k_1$ **then**
  $\mathsf{bad}_3 := \mathsf{true}$, $\boxed{x := E_k^{-1}[P](y)}$
return x

Figure 4.2: Expt'_j includes the boxed statements, whereas Expt_j does not.

Equation (4.4) holds, and if bad_2 does not occur then for $i = 1, \ldots, j$ we have $x_i \oplus k_1 \neq s_0$ and $x_i \oplus k_1 \neq s_1$ (where the former is because $x_{j+1} \oplus k_1 = s_0$ but $x_i \neq x_{j+1}$ by assumption, and the latter is by definition of bad_2). So $P_1(x_i \oplus k_1) = P(x_i \oplus k_1)$ for $i = 1, \ldots, j$, and thus

$$\overrightarrow{S}_{T_j,P_1,k} = \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P_1(x_i \oplus k_1),\, y_i \oplus k_2} = \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P(x_i \oplus k_1),\, y_i \oplus k_2} = \overrightarrow{S}_{T_j,P,k}$$

and

$$\overleftarrow{S}_{T_j,P_1,k} = \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P_1(x_i \oplus k_1),\, y_i \oplus k_2} = \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P(x_i \oplus k_1),\, y_i \oplus k_2} = \overleftarrow{S}_{T_j,P,k}.$$

Therefore

$$
\begin{aligned}
Q = (P_1)_{T_j,k} &= \overleftarrow{S}_{T_j,P_1,k} \circ \overrightarrow{S}_{T_j,P_1,k} \circ P_1 \\
&= \overleftarrow{S}_{T_j,P,k} \circ \overrightarrow{S}_{T_j,P,k} \circ \mathsf{swap}_{P(x_{j+1}\oplus k_1),\, y_{j+1}\oplus k_2} \circ P \\
&= \overleftarrow{S}_{T_{j+1},P,k} \circ \overrightarrow{S}_{T_{j+1},P,k} \circ P \\
&= P_{T_{j+1},k}\,,
\end{aligned}
$$

using Equations (4.3) and (4.4) and the fact that $b_{j+1} = 0$.

2. In Expt'_j, the value y_{j+1} is uniformly distributed in $\{0,1\}^n \setminus \{y_1, \ldots, y_j\}$. Indeed, we have already argued above that the value y_{j+1} computed in line 14 is uniform in $\{0,1\}^n$. But if that value lies in $\{y_1, \ldots, y_j\}$ (and so bad_1 occurs) then y_{j+1} is re-sampled uniformly from $\{0,1\}^n \setminus \{y_1, \ldots, y_j\}$ in line 7.

3. In Expt'_j, the response from oracle $\mathcal{O}(x)$ is always equal to $E_k[P](x)$. When bad_3 occurs this is immediate. But if bad_3 does not occur then $x \neq s_1 \oplus k_1$; we also know that $x \neq s_0 \oplus k_1 = x_{j+1}$ by assumption. But then $P_1(x \oplus k_1) = P(x \oplus k_1)$ and so $E_k[P_1](x) = E_k[P](x)$. A similar argument shows that the response from $\mathcal{O}^{-1}(y)$ is always $E_k^{-1}[P](y)$.

Syntactically rewriting Expt'_j using the above observations yields an experiment that is identical to $\mathbf{H}'_j$. (See Appendix 44 for further details.) Lemma 4.5 thus follows from Equations (4.5) and (4.6). □

In the proof of Lemma 4.5, we only consider the forward queries to make the proof clean and clear, and we claimed that the inverse queries are handled entirely symmetrically. In this section, we'll give detailed explanations.

23 $P, R \leftarrow \mathcal{P}_n$
24 Run $\mathcal{A}$ with quantum access to P and classical access to R, until $\mathcal{A}$ makes its $(j+1)$st classical query x_{j+1}; let T_j be as in the text
25 $s_0, s_1 \leftarrow \{0,1\}^n$
26 $k_1 := s_0 \oplus x_{j+1}$, $k_2 \leftarrow D_{|k_1}$, $k := (k_1, k_2)$
27 $y_{j+1} := P(s_1) \oplus k_2$
28 **if** $y_{j+1} \in \{y_1, \ldots, y_j\}$ **then** $y_{j+1} \leftarrow \{0,1\}^n \setminus \{y_1, \ldots, y_j\}$
29 Give y_{j+1} to $\mathcal{A}$ as the answer to its $(j+1)$st classical query
30 $T_{j+1} := \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$
31 Continue running $\mathcal{A}$ with quantum access to $P_{T_{j+1},k}$ and classical access to $E_k[P]$

32 $P, R \leftarrow \mathcal{P}_n$
33 Run $\mathcal{A}$ with quantum access to P and classical access to R, until $\mathcal{A}$ makes its $(j+1)$st classical query x_{j+1}; let T_j be as in the text
34 $k_1 \leftarrow \{0,1\}^n$, $k_2 \leftarrow D_{|k_1}$, $k := (k_1, k_2)$, $y_{j+1} \leftarrow \{0,1\}^n$
35 **if** $y_{j+1} \in \{y_1, \ldots, y_j\}$ **then** $y_{j+1} \leftarrow \{0,1\}^n \setminus \{y_1, \ldots, y_j\}$
36 Give y_{j+1} to $\mathcal{A}$ as the answer to its $(j+1)$st classical query
37 $T_{j+1} := \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$
38 Continue running $\mathcal{A}$ with quantum access to $P_{T_{j+1},k}$ and classical access to $E_k[P]$

39 $P, R \leftarrow \mathcal{P}_n$
40 Run $\mathcal{A}$ with quantum access to P and classical access to R, until $\mathcal{A}$ makes its $(j+1)$st classical query x_{j+1}; let T_j be as in the text
41 $k \leftarrow D$, $y_{j+1} \leftarrow \{0,1\}^n \setminus \{y_1, \ldots, y_j\}$
42 Give y_{j+1} to $\mathcal{A}$ as the answer to its $(j+1)$st classical query
43 $T_{j+1} := \big((x_1, y_1, b_1), \ldots, (x_{j+1}, y_{j+1}, b_{j+1})\big)$
44 Continue running $\mathcal{A}$ with quantum access to $P_{T_{j+1},k}$ and classical access to $E_k[P]$

Figure 4.3: Syntactic rewritings of Expt'_j.

Equivalence of Expt'_j and $\mathbf{H}'_j$ The code in the top portion of Figure 4.3 is a syntactic rewriting of Expt'_j. (Flags that have no effect on the output of $\mathcal{A}$ are omitted.) In line 27, the computation of y_{j+1} has been expanded (note that $E_k[P_1](x_{j+1}) = P_1(s_0) \oplus k_2 = P(s_1) \oplus k_2$). In line 31, Q has been replaced with $P_{T_{j+1},k}$ and $\mathcal{O}$ has been replaced with $E_k[P]$ as justified in the proof of Lemma 4.5.

The code in the middle portion of Figure 4.3 results from the following changes: first, rather than sampling uniform s_0 and then setting $k_1 := s_0 \oplus x_{j+1}$, the code now samples a uniform k_1.

Similarly, rather than choosing uniform s_1 and then setting $y_{j+1} := P(s_1) \oplus k_2$, the code now samples a uniform y_{j+1} (note that P is a permutation, so $P(s_1)$ is uniform). Since neither s_0 nor s_1 is used anywhere else, each can now be omitted.

The code in the bottom portion of Figure 4.3 simply chooses $k = (k_1, k_2)$ according to distribution D, and chooses uniform $y_{j+1} \in \{0,1\}^n \backslash \{y_1, \ldots, y_j\}$. It can be verified by inspection that this final experiment is equivalent to $\mathbf{H}'_j$.

Handling an Inverse Query We now discuss the case where the $(j+1)$st classical query of $\mathcal{A}$ is a inverse query in the proof of Lemma 4.5. Phase 1 is exactly as described in the proof of Lemma 4.5, though we now let y_{j+1} denote the $(j+1)$st classical query made by $\mathcal{A}$, and now $b_{j+1} = 1$.

Phase 2: $\mathcal{D}$ receives $s_0, s_1 \in \{0,1\}^n$ and quantum oracle access to a permutation P_b. First, $\mathcal{D}$ sets $t_0 := P_b(s_0)$ and $t_1 := P_b(s_1)$. It then sets $k_2 := t_0 \oplus y_{j+1}$, chooses $k_1 \leftarrow D_{|k_2}$ (where this represents the conditional distribution on k_1 given k_2), and sets $k := (k_1, k_2)$. $\mathcal{D}$ continues running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)$st one) using $E_k[P_b]$, and its remaining quantum queries using

$$(P_b)_{T_j,k} = \overleftarrow{S}_{T_j,P_b,k} \circ \overrightarrow{S}_{T_j,P_b,k} \circ P_b = P_b \circ \overleftarrow{Q}_{T_j,P_b,k} \circ \overrightarrow{Q}_{T_j,P_b,k}\,.$$

Finally, $\mathcal{D}$ outputs whatever $\mathcal{A}$ outputs.

Note that t_0, t_1 are uniform, and so k is distributed according to D. Then:

Case $b = 0$ (no reprogramming). In this case, $\mathcal{A}$'s remaining classical queries (including its $(j+1)$st classical query) are answered using $E_k[P_0] = E_k[P]$, and its remaining quantum queries

are answered using $(P_0)_{T_j,k} = P_{T_j,k}$. The output of $\mathcal{A}$ is thus distributed identically to its output in $\mathbf{H}_j$ in this case.

Case $b = 1$ (reprogramming). In this case, $k_2 = P_1(s_0) \oplus y_{j+1} = P(s_1) \oplus y_{j+1}$ and so

$$\begin{aligned} P_b^{-1} = P_1^{-1} = (P \circ \mathsf{swap}_{s_0,\, s_1})^{-1} &= (\mathsf{swap}_{P(s_0),\, P(s_1)} \circ P)^{-1} \\ &= P^{-1} \circ \mathsf{swap}_{P(s_0),\, P(s_1)} \\ &= P^{-1} \circ \mathsf{swap}_{P(s_0),\, y_{j+1} \oplus k_2}. \end{aligned}$$

The response to $\mathcal{A}$'s $(j+1)$st classical query is thus

$$x_{j+1} \stackrel{\text{def}}{=} E_k^{-1}[P_1](y_{j+1}) = P_1^{-1}(y_{j+1} \oplus k_2) \oplus k_1 = P_1^{-1}(P(s_1)) \oplus k_1 = s_0 \oplus k_1 \,.$$

The remaining classical queries of $\mathcal{A}$ are then answered using $E_k[P_1]$, while its remaining quantum queries are answered using $(P_1)_{T_j,k}$.

Now we define the following three events:

1. bad_1 is the event that $x_{j+1} \in \{x_1, \ldots, x_j\}$.

2. bad_2 is the event that $P(s_0) \oplus k_2 \in \{y_1, \ldots, y_j\}$.

3. bad_3 is the event that, in phase 2, $\mathcal{A}$ queries its classical oracle in the forward direction on $s_1 \oplus k_1$, or the inverse direction on $P(s_0) \oplus k_2$.

Comparing the above to the proof of Lemma 4.5, we see (because P is a permutation) that the situation is entirely symmetric, and the analysis is therefore the same.

4.1.3 Security of Forward-only Even-Mansour

In this section, we consider a simpler case, where $E_k[F](x) := F(x \oplus k)$ for $F : \{0,1\}^n \to \{0,1\}^n$ a uniform *function* and k a uniform n-bit string. Here, we restrict the adversary to forward queries only, i.e., the adversary has classical access to $E_k[F]$ and quantum access to F; note that $E_k^{-1}[F]$ and F^{-1} may not even be well-defined. This setting was also analyzed by Jaeger et al. [46] using different techniques.

We let $\mathcal{F}_n$ denote the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$.

Theorem 4.7. *Let $\mathcal{A}$ be a quantum algorithm making q_C classical queries to its first oracle and q_F quantum queries to its second oracle. Then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n \\ F \leftarrow \mathcal{F}_n}} \left[\mathcal{A}^{E_k[F],F}(1^n) = 1 \right] - \Pr_{R,F \leftarrow \mathcal{F}_n} \left[\mathcal{A}^{R,F}(1^n) = 1 \right] \right| \leq 2^{-n/2} \cdot \left(2q_C\sqrt{q_F} + 2q_F\sqrt{q_C}\right).$$

Proof. We make the same assumptions about $\mathcal{A}$ as in the initial paragraphs of the proof of Theorem 4.1. We also adopt analogous notation for the stages of $\mathcal{A}$, now using q_C, q_F, and $q_{F,j}$ as appropriate.

Given a function $F : \{0,1\}^n \to \{0,1\}^n$, a set T of pairs where any $x \in \{0,1\}^n$ is the first element of at most one pair in T, and a key $k \in \{0,1\}^n$, we define the function $F_{T,k} : \{0,1\}^n \to \{0,1\}^n$ as

$$F_{T,k}(x) := \begin{cases} y & \text{if } (x \oplus k, y) \in T \\ F(x) & \text{otherwise.} \end{cases}$$

Note that, in contrast to the analogous definition in Theorem 4.1, here the order of the tuples in

T does not matter and so we may take it to be a set. Note also that we are redefining the notation $F_{T,k}$ from how it was used in Theorem 4.1; this notation applies to this appendix only.

We now define a sequence of experiments $\mathbf{H}_j$, for $j = 0, \ldots, q_C$:

Experiment $\mathbf{H}_j$. Sample $R, F \leftarrow \mathcal{F}_n$ and $k \leftarrow \{0,1\}^n$. Then:

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using F, stopping immediately before its $(j+1)$st classical query. Let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the set of all classical queries made by $\mathcal{A}$ thus far and their corresponding responses.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $E_k[F]$ and its quantum queries using $F_{T_j,k}$.

We can represent $\mathbf{H}_j$ as the experiment in which $\mathcal{A}$'s queries are answered using the oracle sequence

$$\underbrace{F, R, F, \cdots, R, F}_{j \text{ classical queries}}, \underbrace{E_k[F], F_{T_j,k}, \cdots, E_k[F], F_{T_j,k}}_{q_C - j \text{ classical queries}}.$$

Note that $\mathbf{H}_0$ is exactly the real world (i.e., $\mathcal{A}^{E_k[F],F}$) and $\mathbf{H}_{q_C}$ is exactly the ideal world (i.e., $\mathcal{A}^{R,F}$.)

For $j = 0, \ldots, q_C - 1$, we define an additional experiment $\mathbf{H}'_j$:

Experiment $\mathbf{H}'_j$. Sample $R, F \leftarrow \mathcal{F}_n$ and $k \leftarrow \{0,1\}^n$. Then:

1. Run $\mathcal{A}$, answering its classical queries using R and its quantum queries using F, stopping immediately after its $(j+1)$st classical query. Let $T_{j+1} = \big((x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\big)$ be the set of all classical queries made by $\mathcal{A}$ thus far and their corresponding responses.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $E_k[F]$ and its quantum queries using $F_{T_{j+1},k}$.

I.e., $\mathbf{H}'_j$ corresponds to answering $\mathcal{A}$'s queries using the oracle sequence

$$\underbrace{F, R, F, \cdots, R, F}_{j \text{ classical queries}}, R, F_{T_{j+1},k}, \underbrace{E_k[F], F_{T_{j+1},k} \cdots, E_k[F], F_{T_{j+1},k}}_{q_C - j - 1 \text{ classical queries}}.$$

We now show that $\mathbf{H}'_j$ is close to $\mathbf{H}_{j+1}$ and $\mathbf{H}_j$ is close to $\mathbf{H}'_j$ for $0 \leq j < q_C$.

Lemma 4.8. For $j = 0, \ldots, q_C - 1$,

$$|\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{F,j+1}\sqrt{(j+1)/2^n}.$$

Proof. Given an adversary $\mathcal{A}$, we construct a distinguisher $\mathcal{D}$ for the "arbitrary reprogramming game" of Lemma 3.1 that works as follows:

Phase 1: $\mathcal{D}$ samples $F, R \leftarrow \mathcal{F}_n$. It then runs $\mathcal{A}$, answering its quantum queries with F and its classical queries with R, until it replies to $\mathcal{A}$'s $(j+1)$st classical query. Let $T_{j+1} = \{(x_1, y_1), \ldots, (x_{j+1}, y_{j+1})\}$ be the set of classical queries/answers thus far. $\mathcal{D}$ defines algorithm $\mathcal{B}$ as follows: on randomness $k \in \{0,1\}^n$, output $B = \{(x_j \oplus k, y_j)\}_{j=1}^{j+1}$. Finally, $\mathcal{D}$ outputs F and $\mathcal{B}$.

Phase 2: $\mathcal{D}$ is given quantum access to a function F_b. It continues to run $\mathcal{A}$, answering its quantum queries with F_b until $\mathcal{A}$ makes its next classical query.

Phase 3: $\mathcal{D}$ is given the randomness k used to run $\mathcal{B}$. It continues running $\mathcal{A}$, answering its classical queries with $E_k[F]$ and its quantum queries with $F_{T_{j+1},k}$. Finally, $\mathcal{D}$ outputs whatever $\mathcal{A}$ outputs.

When $b = 0$ (so $F_b = F_0 = F$), then $\mathcal{A}$'s output is identically distributed to its output in

$\mathbf{H}_{j+1}$. On the other hand, when $b = 1$ then $F_b = F_1 = F^{(B)} = F_{T_{j+1},k}$ and so $\mathcal{A}$'s output is identically distributed to its output in $\mathbf{H}'_j$. The expected number of queries made by $\mathcal{D}$ in phase 2 when $F = F_0$ is the expected number of queries made by $\mathcal{A}$ in stage $(j+1)$ in $\mathbf{H}_{j+1}$. Since $\mathbf{H}_{j+1}$ and $\mathbf{H}_{q_C}$ are identical until after the $(j+1)$st stage, this is precisely $q_{F,j+1}$. Because k is uniform, we can apply Lemma 3.1 with $\varepsilon = (j+1)/2^n$. The lemma follows. □

Lemma 4.9. For $j = 0, \ldots, q_C$,

$$|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]| \leq 1.5 \cdot \sqrt{q_F/2^n}\,.$$

Proof. From any adversary $\mathcal{A}$, we construct a distinguisher $\mathcal{D}$ for the game of Lemma 3.2. $\mathcal{D}$ works as follows:

Phase 1: $\mathcal{D}$ is given quantum access to a (random) function F. It samples $R \leftarrow \mathcal{F}_n$ and then runs $\mathcal{A}$, answering its quantum queries using F and its classical queries using R, until $\mathcal{A}$ submits its $(j+1)$st classical query x_{j+1}. At that point, let $T_j = \{(x_1, y_1), \ldots, (x_j, y_j)\}$ be the set of input/output pairs $\mathcal{A}$ has received from its classical oracle thus far.

Phase 2: $\mathcal{D}$ is given (uniform) $s \in \{0,1\}^n$ and quantum oracle access to a function F_b. Then $\mathcal{D}$ sets $k := s \oplus x_{j+1}$, and then continues running $\mathcal{A}$, answering its classical queries (including the $(j+1)$st) using $E_k[F_b]$ and its quantum queries using the function $(F_b)_{T_j,k}$, i.e.,

$$x \mapsto \begin{cases} y & \text{if } (x \oplus k, y) \in T_j \\ F_b(x) & \text{otherwise.} \end{cases}$$

Finally, $\mathcal{D}$ outputs whatever $\mathcal{A}$ outputs.

We analyze the execution of $\mathcal{D}$ in the two cases of the game of Lemma 3.2. In either case, the quantum queries of $\mathcal{A}$ in stages $0, \ldots, j$ are answered using a random function F, and $\mathcal{A}$'s first j classical queries are answered using an independent random function R. Note further that since s is uniform, so is k.

Case 1: $b = 0$**.** In this case, all the remaining classical queries of $\mathcal{A}$ (i.e., from the $(j+1)$st on) are answered using $E_k[F]$, and the remaining quantum queries of $\mathcal{A}$ are answered using $F_{T_j,k}$. The output of $\mathcal{A}$ is thus distributed identically to its output in $\mathbf{H}_j$ in this case.

Case 2: $b = 1$**.** Here, $F_b = F_1 = F_{s \to y}$ for a uniform y. Now, the response to the $(j+1)$st classical query of $\mathcal{A}$ is

$$E_k[F_b](x_{j+1}) = E_k[F_{s \to y}](x_{j+1}) = F_{s \mapsto y}(k \oplus x_{j+1}) = F_{s \to y}(s) = y.$$

Since y is uniform and independent of anything else, and since $\mathcal{A}$ has never previously queried x_{j+1} to its classical oracle, this is equivalent to answering the first $j+1$ classical queries of $\mathcal{A}$ using a random function R. The remaining classical queries of $\mathcal{A}$ are also answered using $E_k[F_{s \mapsto y}]$. However, since $E_k[F_{s \to y}](x) = E_k[F](x)$ for all $x \neq x_{j+1}$ and $\mathcal{A}$ never repeats the query x_{j+1}, this is equivalent to answering the remaining classical queries of $\mathcal{A}$ using $E_k[F]$.

The remaining quantum queries of $\mathcal{A}$ are answered with the function

$$x \mapsto \begin{cases} y' & \text{if } (x \oplus k, y') \in T_j \\ F_{s \to y}(x) & \text{otherwise.} \end{cases}$$

This, in turn, is precisely the function $F_{T_{j+1},k}$, where T_{j+1} is obtained by adding (x_{j+1}, y) to T_j

(and thus consists of the first $j+1$ classical queries made by $\mathcal{A}$ and their corresponding responses). Thus, the output of $\mathcal{A}$ in this case is distributed identically to its output in $\mathbf{H}'_j$.

The number of quantum queries made by $\mathcal{D}$ in phase 1 is at most q_F. The claimed result thus follows from Lemma 3.2. □

Using Lemmas 4.8 and 4.9, and the fact that $\sum_{j=1}^{q_C} q_{F,j} = q_F$, we have

$$\begin{aligned} |\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1]| &\leq 1.5 q_C \sqrt{q_F/2^n} + 2 \sum_{j=1}^{q_C} q_{F,j} \sqrt{j/2^n} \\ &\leq 1.5 q_C \sqrt{q_F/2^n} + 2 \sqrt{q_C/2^n} \sum_{j=1}^{q_C} q_{F,j} \\ &\leq 1.5 q_C \sqrt{q_F/2^n} + 2 q_F \sqrt{q_C/2^n}\,, \end{aligned}$$

as required. □

4.2 Tweakable Even-Mansour Cipher

4.2.1 Overview

The development of large-scale quantum computers would have a significant impact on cryptography. For symmetric-key cryptosystems—even ideal ciphers—one must at least double the key length in order to achieve the same security against quantum attackers as is enjoyed against classical adversaries, due to key-recovery attacks via Grover search [61]. In general, however, doubling the key length may not be sufficient [43, 62, 63], and it is therefore critical to understand the security of various symmetric-key constructions against quantum attackers.

Tweakable block cipher. Block ciphers are fundamental cryptographic primitives in symmetric-

key cryptography. A block cipher is a family of permutations parameterized by a key; if the key is selected uniformly at random, the resulting permutation appears pseudorandom to adversaries with appropriate query access. A *tweakable block cipher* has an additional parameter (the *tweak*) that allows for easily selecting different permutations for a fixed key. For tweakable block ciphers, security must hold even in a setting where the adversary can select (possibly related) tweaks for each query. Tweakable block ciphers are used frequently in practice, e.g., for disk encryption.

Tweakable Even-Mansour Cipher. Let $P : \{0,1\}^n \to \{0,1\}^n$ be a permutation. The tweakable Even-Mansour scheme $\mathsf{TEM}^{f_1,f_2}[P] : \{0,1\}^n \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is defined as

$$\mathsf{TEM}_k^{f_1,f_2}[P](t,x) = P(x \oplus f_1(t,k)) \oplus f_2(t,k),$$

where the key k is of length n, the set $\mathcal{T}$ is a tweak space, and f_1, f_2 are functions satisfying some structural properties:

Definition 4.10. A function $f : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is **proper** (with respect to $\mathcal{T}$) if it satisfies the following two properties:

Uniformity: For all $t \in \mathcal{T}$, the function $f(t,\cdot)$ is a permutation.

XOR-universality: For all distinct $t, t' \in \mathcal{T}$ and all $y \in \{0,1\}^n$,

$$\Pr_{k \leftarrow \{0,1\}^n}[f(t,k) \oplus f(t',k) = y] \leq 2^{-n}\,.$$

Figure 4.4 illustrates the construction of the Tweakable Even-Mansour cipher. In this section, we show the post-quantum security of the *tweakable* Even-Mansour construction, a tweak-

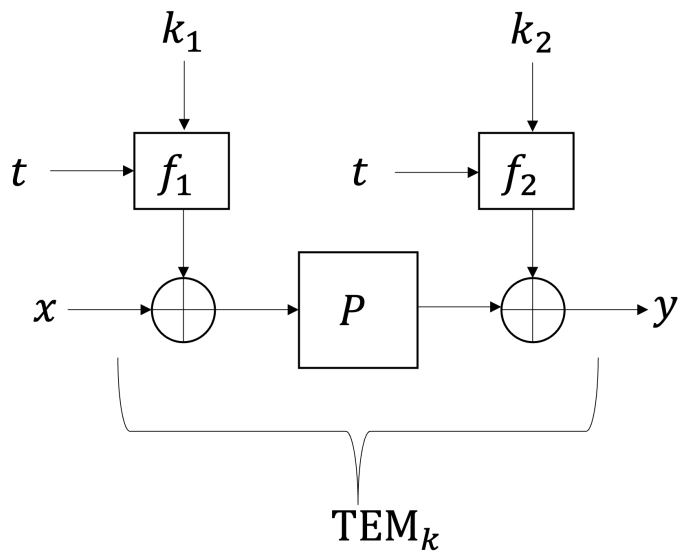


Figure 4.4: Depiction of the Tweakable Block Cipher

able block cipher constructed from a public random permutation, in the Q1 model. We then use this result to establish post-quantum security of several symmetric-key schemes. We stress that the post-quantum security of tweakable Even-Mansour does not follow from the post-quantum security of Even-Mansour. Indeed, the tweak must be incorporated in a way that satisfies several technical conditions; in addition, incorporating both tweaks and key expansion introduces dependencies and requires significant technical work to analyze. We also remark that our setting is significantly different from that of [64, 65, 66]. Those works are focused on classical-quantum query tradeoffs (for basic query complexity problems) when both the classical and the quantum oracle are for the same function; moreover, they do not consider permutations, even in the one-way-accessible setting.

In all of our results, adversaries can make adaptive queries to any permutations to which they have access (whether quantum or classical, as appropriate) in both the forward and inverse directions.

A new resampling lemma. As a key technical tool used for our results, we prove a generalization of existing "resampling lemmas" [27, 38] sufficient to handle tweakable block ciphers, something we believe to be of independent interest. A resampling lemma controls the success probability of a quantum-query adversary $\mathcal{D}$ in an experiment of the following form:

1. $\mathcal{D}$ receives quantum oracle access to a random permutation P;
2. two inputs s_0, s_1 are sampled from some distribution;
3. $\mathcal{D}$ receives quantum oracle access to either P, or P with inputs s_0 and s_1 "swapped"; it succeeds if it can correctly guess which is the case.

Prior work considered only the uniform distribution on s_0, s_1. We give a new resampling lemma that handles a wider class of (adversarially influenced) distributions, and even allows the distribution to depend on information $\mathcal{D}$ learns about P during step 1 of the above experiment. This new resampling lemma is formally stated as Lemma 3.5 and proved in Section 3.2.2.

4.2.2 Post-Quantum Security of Tweakable Even-Mansour

We let $\mathcal{P}(n)$ denote the set of all permutations on $\{0,1\}^n$. In the *public-permutation model* (or random permutation model), a permutation $P \leftarrow \mathcal{P}(n)$ is sampled uniformly and then provided as an oracle (in both the forward and inverse directions) to all parties.

A block cipher $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ is a keyed permutation, i.e., $E_k(\cdot) = E(k, \cdot)$ is a permutation of $\{0,1\}^n$ for all $k \in \{0,1\}^\kappa$. We say E is a *pseudorandom permutation* if E_k (for uniform $k \in \{0,1\}^\kappa$) is indistinguishable from a uniform permutation in $\mathcal{P}(n)$, where indistinguishability is required to hold even against adversaries who may query their oracle in both the forward and inverse directions.

For a set $\mathcal{T}$, let $\mathcal{E}(\mathcal{T}, n)$ be the set of all functions $E : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ such that $E(t, \cdot)$ is a permutation on $\{0,1\}^n$ for all $t \in \mathcal{T}$. A tweakable block cipher $\tilde{E} : \{0,1\}^\kappa \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is a family of permutations indexed by both a key $k \in \{0,1\}^\kappa$ and a tweak $t \in \mathcal{T}$, i.e., we now require that $\tilde{E}_k(t, \cdot) = \tilde{E}(k, t, \cdot)$ is a permutation of $\{0,1\}^n$ for all $k \in \{0,1\}^\kappa$ and $t \in \mathcal{T}$. A tweakable block cipher $\tilde{E}_k$ is *secure* if $\tilde{E}_k$ (for uniform choice of $k \in \{0,1\}^\kappa$) is indistinguishable from a uniform $\tilde{E} \leftarrow \mathcal{E}(\mathcal{T}, n)$.

In all 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 public-permutation model can apply the unitary operators

$$|x\rangle|y\rangle \mapsto |x\rangle|x \oplus P(y)\rangle$$

$$|x\rangle|y\rangle \mapsto |x\rangle|x \oplus P^{-1}(y)\rangle$$

to quantum registers of the adversary's choice. (We emphasize that this includes evaluating P/P^{-1} on arbitrary superpositions of inputs.) This is well-motivated, as in practice P would be instantiated by a publicly known permutation; adversaries with quantum computers would thus be able to coherently execute the reversible circuit for computing P/P^{-1}. On the other hand, secretly keyed primitives would be implemented by honest parties; if honest parties only evaluate the primitive on classical inputs then the attacker has no way to obtain quantum access to that keyed primitive.

We claim the post-quantum security of three different variants of the tweakable Even-

Mansour construction. We start with security of TEM-KX, and then prove security of TEM as a simple corollary. In addition, we also prove the security of TEM-KX1 by showing that its key-expansion function is a pseudorandom generator (PRG).

I: Security of TEM-KX

Let $P \in \mathcal{P}(n)$ be a permutation and $\mathcal{T}$ a finite set, and fix two functions $f_1, f_2 : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$. We consider a key-expanding version of the tweakable Even-Mansour construction $\mathsf{TEM\text{-}KX}^{f_1,f_2}[P] : \{0,1\}^\kappa \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ defined as

$$\mathsf{TEM\text{-}KX}_k^{f_1,f_2}[P](t,x) = P(x \oplus f_1(t, P(k||0^{n-\kappa}))) \oplus f_2(t, P(k||0^{n-\kappa}))\,.$$

We assume the tweak functions f_1, f_2 are proper as Definition 4.10.

satisfy some structural properties:

Theorem 4.11. *Let* TEM-KX *be as above, and let $\mathcal{A}$ be an adversary making q_C classical queries to its first oracle and $q_Q \geq \max\{n, \log(11 \cdot |\mathcal{T}|)\}$ quantum queries*[3] *to its second oracle. If f_1, f_2 are proper with respect to $\mathcal{T}$, then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^\kappa; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{TEM\text{-}KX}_k^{f_1,f_2}[P],P} = 1 \right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T},n); \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E},P} = 1 \right] \right| \leq 7 \cdot 2^{-\kappa/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

Proof. The high-level structure of our proof is similar to the proof of security for the Even-Mansour construction by Alagic et al. [27], though here relying heavily on our new resampling

[3]The mild assumption on q_Q can be avoided at the expense of an additive term of $\mathcal{O}(q_C \cdot 2^{-\kappa/2} \cdot (n + \log|\mathcal{T}|))$ in the bound.

lemma. For that reason, we copy some portions of their proof (with appropriate updates for our setting).

Without loss of generality, we assume $\mathcal{A}$ never makes a redundant classical query; that is, once it learns a triple (t, x, y) of tweak, input, and output by making a query to its classical oracle, it never again submits a query (t, x) (resp., (t, y)) to that oracle in the forward (resp., inverse) direction. We divide an execution of $\mathcal{A}$ into $q_C + 1$ stages $0, \ldots, q_C$, where the jth stage corresponds to the time between the jth and $(j+1)$st classical queries of $\mathcal{A}$. (The 0th stage is the period of time before $\mathcal{A}$ makes its first classical query, and the q_Cth stage is the period of time after $\mathcal{A}$ makes its last classical query.) $\mathcal{A}$ may adaptively[4] distribute its q_Q quantum queries between these stages arbitrarily, and we let $q_{Q,j}$ be the expected number of quantum queries that $\mathcal{A}^{\tilde{E},P}$ makes in the jth stage, where the expectation is taken over $\tilde{E} \leftarrow \mathcal{E}(\mathcal{T}, n)$ and $P \leftarrow \mathcal{P}(n)$ and any internal randomness/measurements of $\mathcal{A}$. Note that $\sum_{j=0}^{q_C} q_{Q,j} = q_Q$.

Fixing f_1, f_2, we write $\mathsf{TEM\text{-}KX}_k$ for $\mathsf{TEM\text{-}KX}_k^{f_1,f_2}$. In a given execution of $\mathcal{A}$, we denote its jth classical query by (t_j, x_j, y_j, b_j), where $t_j \in \mathcal{T}$ is a tweak, $(x_j, y_j) \in \{0,1\}^n \times \{0,1\}^n$ is an input/output pair, and $b_j \in \{0,1\}$ indicates the query direction, i.e., $b_j = 0$ (resp., $b_j = 1$) means that the jth classical query was in the forward (resp., inverse) direction. We let $T_j = \big((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j)\big)$ be the ordered list of the first j classical queries of $\mathcal{A}$.

Our proof involves a sequence of experiments in which $\mathcal{A}$'s oracles are modified based on the classical queries made by $\mathcal{A}$ thus far. We first establish the appropriate notation. We use the product symbol $\prod$ to denote sequential composition of operations, i.e., $\prod_{i=1}^{n} f_i = f_1 \circ \cdots \circ f_n$. Note that order matters, since function composition is not commutative in general. We use the

[4] Alternatively, the techniques of [67] can be used to turn the adversary into one that uses a fixed query schedule; the overall bound would be unchanged.

notation $\prod_{i=n}^{1} f_i = f_n \circ \cdots \circ f_1$ to denote the composition in reverse order. For a permutation P, a key k, and a list $T_j = \big((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j)\big)$ as above, define the operators

$$\begin{aligned}
\overrightarrow{S}_{T_j,P,k} &= \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))} \\
\overrightarrow{Q}_{T_j,P,k} &= \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})),\, P^{-1}(y_i \oplus f_2(t_i, P(k||0^{n-\kappa})))} \\
\overleftarrow{S}_{T_j,P,k} &= \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P(x_i \oplus f_1(t_i, P(k||0^{0-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))} \\
\overleftarrow{Q}_{T_j,P,k} &= \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})),\, P^{-1}(y_i \oplus f_2(t_i, P(k||0^{n-\kappa})))}
\end{aligned}$$

where, as usual, f^0 is the identity map and $f^1 = f$ for any function f. We define the modified permutation $P^{T_j,k}$ as

$$P^{T_j,k}(x) = \overleftarrow{S}_{T_j,P,k} \circ \overrightarrow{S}_{T_j,P,k} \circ P(x)\,.$$

Since $P \circ \mathsf{swap}_{x,\,y} = \mathsf{swap}_{P(x),\,P(y)} \circ P$ for all x, y, we have

$$\overleftarrow{S}_{j,P,k} \circ \overrightarrow{S}_{T_j,P,k} \circ P = \overleftarrow{S}_{T_j,P,k} \circ P \circ \overrightarrow{Q}_{T_j,P,k} = P \circ \overleftarrow{Q}_{T_j,P,k} \circ \overrightarrow{Q}_{T_j,P,k}\,.$$

Roughly speaking, $P^{T_j,k}$ is the minimal modification of P that is consistent with the forward ($\rightarrow$) and inverse ($\leftarrow$) queries from the transcript T_j when post-composed (S) or pre-composed (Q) with P. For compactness, we occasionally write P^j in place of $P^{T_j,k}$ when T_j and k are understood from the context.

We now define a sequence of hybrid experiments $\mathbf{H}_j$, for $j = 0, \ldots, q_C$.

Experiment $\mathbf{H}_j$. Sample uniform $\tilde{E} \in \mathcal{E}(\mathcal{T}, n)$ and $P \in \mathcal{P}(n)$, and a uniform key $k \in \{0,1\}^\kappa$.

Then:

1. Run $\mathcal{A}$, answering its classical queries using $\tilde{E}$ and its quantum queries using P, stopping immediately *before* its $(j{+}1)$st classical query. Let $T_j = \big((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j)\big)$ be the list of classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $\mathsf{TEM\text{-}KX}_k[P^{T_j,k}]$ and its quantum queries using $P^{T_j,k}$.

We can compactly represent $\mathbf{H}_j$ as the experiment in which $\mathcal{A}$'s queries are answered using the oracle sequence

$$\underbrace{P, \tilde{E}, P, \cdots, \tilde{E}, P,}_{j \text{ classical queries}} \underbrace{\mathsf{TEM\text{-}KX}_k[P^j], P^j, \cdots, \mathsf{TEM\text{-}KX}_k[P^j], P^j}_{q_C - j \text{ classical queries}}.$$

Each instance of $\tilde{E}$ or $\mathsf{TEM\text{-}KX}_k[P^j]$ represents a single classical query, while each instance of P or P^j represents a stage during which $\mathcal{A}$ makes multiple quantum queries to that oracle but no queries to its classical oracle. Observe that $\mathbf{H}_0$ corresponds to the execution of $\mathcal{A}$ in the real world, i.e., $\mathcal{A}^{\mathsf{TEM\text{-}KX}_k[P],P}$, and $\mathbf{H}_{q_C}$ is the execution of $\mathcal{A}$ in the ideal world, i.e., $\mathcal{A}^{\tilde{E},P}$.

For $j = 0, \ldots, q_C - 1$, we introduce additional experiments $\mathbf{H}'_j$:

Experiment $\mathbf{H}'_j$. Sample uniform $\tilde{E} \in \mathcal{E}(\mathcal{T}, n)$ and $P \in \mathcal{P}(n)$, and uniform $k \in \{0,1\}^\kappa$. Then:

1. Run $\mathcal{A}$, answering its classical queries using $\tilde{E}$ and its quantum queries using P, stopping immediately *after* its $(j{+}1)$st classical query. Let $T_{j+1} = \big((t_1, x_1, y_1, b_1), \ldots, (t_{j+1}, x_{j+1}, y_{j+1}, b_{j+1})\big)$ be the classical queries so far.

2. For the remainder of the execution of $\mathcal{A}$, answer its classical queries using $\mathsf{TEM\text{-}KX}_k[P^{T_{j+1},k}]$ and its quantum queries using $P^{T_{j+1},k}$.

Thus, $\mathbf{H}'_j$ corresponds to running $\mathcal{A}$ using the oracle sequence

$$\underbrace{P, \tilde{E}, P, \cdots, \tilde{E}, P,}_{j \text{ classical queries}} \tilde{E}, P^{j+1}, \underbrace{\mathsf{TEM\text{-}KX}_k[P^{j+1}], P^{j+1} \cdots, \mathsf{TEM\text{-}KX}_k[P^{j+1}], P^{j+1}}_{q_C - j - 1 \text{ classical queries}}.$$

In Lemma 4.12 and Lemma 4.13, we establish the following bounds on the distinguishability of $\mathbf{H}'_j$ and $\mathbf{H}_{j+1}$, as well as $\mathbf{H}_j$ and $\mathbf{H}'_j$, for $0 \leq j < q_C$:

$$\left|\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]\right| \leq 2^{-\kappa/2} \cdot 2 \cdot q_{Q,j+1}\sqrt{2 \cdot (j+1)}$$

and

$$\begin{aligned}
&\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| \\
&\leq 2^{-\kappa/2}\left(1 + \sqrt{q_Q + \log(11\,|\mathcal{T}|) + n + \kappa/2}\right) + \frac{4j}{2^\kappa}.
\end{aligned}$$

Using the above, we have

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_0) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C}) = 1]| \\
&\leq \sum_{j=0}^{q_C-1}\left(2^{-\kappa/2}\left(1 + \sqrt{q_Q + \log(11\,|\mathcal{T}|) + n + \kappa/2} + 2q_{Q,j+1}\sqrt{2(j+1)}\right) + \frac{4j}{2^\kappa}\right) \\
&\leq \frac{4q_C^2}{2^\kappa} + \sum_{j=0}^{q_C-1} 2^{-\kappa/2}\left(1 + \sqrt{q_Q + \log(11\,|\mathcal{T}|) + n + \kappa/2} + 2 \cdot q_{Q,j+1}\sqrt{2q_C}\right) \\
&\leq \frac{4q_C^2}{2^\kappa} + 2^{-\kappa/2}\left(q_C + q_C\sqrt{q_Q + \log(11\,|\mathcal{T}|) + n + \kappa/2} + 2\sqrt{2}q_Q\sqrt{q_C}\right).
\end{aligned}$$

The above bound can be simplified. By assumption, $q_Q \geq \log(11 \cdot |\mathcal{T}|)$ and $q_Q \geq n \geq \kappa$. So $\sqrt{q_Q + \log(11 \cdot |\mathcal{T}|) + n + \kappa/2} \leq \sqrt{7q_Q/2}$. We may also assume $q_C \leq 2^{\kappa/2}$ since otherwise the

bound is larger than 1. Under these assumptions, we have $4q_C^2 \cdot 2^{-\kappa} \leq 4q_C \cdot 2^{-\kappa/2} \leq 4q_C\sqrt{q_Q} \cdot 2^{-\kappa/2}$ and so

$$\begin{aligned}
&\frac{4q_C^2}{2^\kappa} + 2^{-\kappa/2} \cdot \left(q_C + q_C\sqrt{q_Q + \log(11 \cdot |\mathcal{T}|) + n + \kappa/2} + 2\sqrt{2}q_Q\sqrt{q_C}\right) \\
&\leq 2^{-\kappa/2} \cdot \left(5q_C + q_C\sqrt{7q_Q/2} + 2\sqrt{2}q_Q\sqrt{q_C}\right) \\
&\leq 2^{-\kappa/2} \cdot \left(\left(5 + \sqrt{\frac{7}{2}}\right) q_C\sqrt{q_Q} + 2\sqrt{2}q_Q\sqrt{q_C}\right) \\
&\leq 2^{-\kappa/2} \cdot \left(7q_C\sqrt{q_Q} + 2\sqrt{2}q_Q\sqrt{q_C}\right) \leq 7 \cdot 2^{-\kappa/2} \cdot \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right),
\end{aligned}$$

as claimed.

We now prove Lemma 4.12 and Lemma 4.13.

Lemma 4.12. For $j = 0, \ldots, q_C - 1$,

$$\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]| \leq 2 \cdot q_{Q,j+1}\sqrt{2 \cdot (j+1)/2^\kappa},$$

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 $\mathbf{H}'_j$ and $\mathbf{H}_{j+1}$. We construct a distinguisher $\mathcal{D}$ for the experiment from Lemma 3.1:

Phase 1: $\mathcal{D}$ samples uniform $\tilde{E} \in \mathcal{E}(\mathcal{T}, n)$ and $P \in \mathcal{P}(n)$. It then runs $\mathcal{A}$, answering its quantum queries using P and its classical queries using $\tilde{E}$, until after it responds to $\mathcal{A}$'s $(j+1)$st classical query. Let $T_{j+1} = \big((t_1, x_1, y_1, b_1), \ldots, (t_{j+1}, x_{j+1}, y_{j+1}, b_{j+1})\big)$ be the list of classical queries by $\mathcal{A}$ thus far. $\mathcal{D}$ defines $F(a, x) := P^a(x)$ for $a \in \{1, -1\}$.

It also defines the following randomized algorithm $\mathcal{B}$: sample $k \leftarrow \{0,1\}^\kappa$ and then compute the set B of input/output pairs to be reprogrammed so that $F^{(B)}(a,x) = (P^{T_{j+1},k})^a(x)$ for all a, 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., $(j+2)$nd) classical query.

Phase 3: $\mathcal{D}$ is given k. It resumes running $\mathcal{A}$, answering its classical queries using $\mathsf{TEM\text{-}KX}_k[P^{T_{j+1},k}]$ and its quantum queries using $P^{T_{j+1},k}$. 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 $\mathbf{H}'_j$. It follows that $|\Pr[\mathcal{A}(\mathbf{H}'_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]|$ is equal to the distinguishing advantage of $\mathcal{D}$ in the reprogramming experiment of Lemma 3.1. 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 $P^{T_{j+1},k}$ and the fact that $F^{(B)}(a,x) = (P^{T_{j+1},k})^a(x)$. Fixing P and T_{j+1}, the probability that any particular input (a,x) is reprogrammed is at most the probability (over k) that it lies in the set

$$\left\{ \begin{array}{c} (1, x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\ (1, P^{-1}(y_i \oplus f_2(t_i, P(k||0^{n-\kappa})))), \\ (-1, P(x_i \oplus f_1(t_i, P(k||0^{n-\kappa})))),\ (-1, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))) \end{array} \right\}_{i=1}^{j+1}.$$

We compute the probability that $(a,x) = (1, x_i \oplus f_1(t_i, P(k||0^{n-\kappa})))$ for some fixed i. P is a

permutation, and so is $f_1(t_i, \cdot)$. As k is uniform,

$$\Pr_k[(a,x) = (1, x_i \oplus f_1(t_i, P(k||0^{n-\kappa})))] = \begin{cases} 2^{-\kappa} & a = 1 \\ 0 & a = -1 \end{cases}.$$

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

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 $(j+1)$st stage in $\mathbf{H}_{j+1}$. Since $\mathbf{H}_{j+1}$ and $\mathbf{H}_{q_C}$ are identical until after the $(j+1)$st stage is complete, this is precisely $q_{Q,j+1}$.

□

Lemma 4.13. For $j = 0, \ldots, q_C$,

$$\begin{aligned} &\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| \\ &\leq \frac{1}{2^{\kappa/2}}\left(1 + \sqrt{q_Q + \log(11\,|\mathcal{T}|) + n + \kappa/2}\right) + \frac{4j}{2^\kappa}. \end{aligned}$$

Proof. We introduce additional experiments $\mathbf{H}^*_j$ and $\mathbf{H}^{**}_j$.

Experiment $\mathbf{H}^*_j$. Sample uniform $\tilde{E} \in \mathcal{E}(\mathcal{T}, n)$, $P \in \mathcal{P}(n)$, and $k \in \{0,1\}^\kappa$. Then

1. Run $\mathcal{A}$, answering its classical queries using $\tilde{E}$ and its quantum queries using P, until $\mathcal{A}$ makes its $(j+1)$st classical query $(t_{j+1}, x_{j+1}, b_{j+1} = 0)$, which we assume for concreteness to be in the forward direction.[5]

2. Define $s_0 = f_1(t_{j+1}, P(k||0^{n-\kappa})) \oplus x_{j+1}$ and sample uniform $s_1 \in \{0,1\}^n$. Define $P^{(1)}$

[5] As in [27], the case of an inverse query is entirely symmetric.

as $P^{(1)}(x) = (P \circ \mathsf{swap}_{s_0,\, s_1})(x)$. Then continue running $\mathcal{A}$, answering its remaining classical queries (including the $(j+1)$st) using $\mathsf{TEM\text{-}KX}_k[(P^{(1)})^{T_j,k}]$, and its quantum queries using $(P^{(1)})^{T_j,k}$.

Experiment $\mathbf{H}_j^{**}$ is the same as $\mathbf{H}_j^*$, except that the $(j+1)$st query is answered using $\tilde{E}$ to obtain $y_{j+1} = \tilde{E}(t_{j+1}, x_{j+1})$, and then we define $s_1 = (P^{T_j,k})^{-1}(y_{j+1} \oplus f_2(t_{j+1}, P(k\|0^{n-\kappa})))$. We have

$$\begin{aligned}
\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| \;\leq\; & \left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^*) = 1]\right| \\
& + \left|\Pr[\mathcal{A}(\mathbf{H}_j^*) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1]\right| \\
& + \left|\Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right|.
\end{aligned}$$

We now bound the three differences on the right-hand side.

Let $\mathcal{A}$ be a distinguisher between $\mathbf{H}_j$ and $\mathbf{H}_j^*$. We construct a distinguisher $\mathcal{D}$ for the experiment of Lemma 3.5, where $F = \{f_1(t, \cdot) \oplus x\}_{t \in \mathcal{T}, x \in \{0,1\}^n}$.

Phase 1: $\mathcal{D}$ is given quantum access to a uniform permutation P. It samples uniform $\tilde{E} \leftarrow \mathcal{E}(\mathcal{T}, n)$ and then runs $\mathcal{A}$, answering its quantum queries using P and its classical queries using $\tilde{E}$ (in the appropriate directions), until $\mathcal{A}$ submits its $(j+1)$st classical query $(t_{j+1}, x_{j+1}, b_{j+1} = 0)$. At that point, $\mathcal{D}$ has a list $T_j = \big((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j)\big)$ of the queries $\mathcal{A}$ has made to its classical oracle thus far. $\mathcal{D}$ lets $\tau \in F$ be such that $\tau(\cdot) = f_1(t_{j+1}, \cdot) \oplus x_{j+1}$, and defines the distribution D on $\{0,1\}^n$ that chooses uniform $k \in \{0,1\}^\kappa$ and outputs $k\|0^{n-\kappa}$. Finally, $\mathcal{D}$ outputs (D, τ).

Phase 2: The challenger samples $\hat{s} \leftarrow D$ with $\hat{s} = k\|0^{n-\kappa}$. Then $\mathcal{D}$ is given $\hat{s}$ and quantum

oracle access to the permutation $P^{(b)}$. It continues running $\mathcal{A}$, answering its remaining classical queries—including the $(j+1)$st—using $\mathsf{TEM\text{-}KX}_k[(P^{(b)})^{T_j,k}]$, and its remaining quantum queries using $(P^{(b)})^{T_j,k}$. $\mathcal{D}$ outputs whatever $\mathcal{A}$ does.

In phase 1, distinguisher $\mathcal{D}$ perfectly simulates experiments $\mathbf{H}_j$ and $\mathbf{H}_j^*$ for $\mathcal{A}$ until the point where $\mathcal{A}$ makes its $(j+1)$st classical query. If $b = 0$, $\mathcal{D}$ gets access to $P^{(0)} = P$ in phase 2. Since $\mathcal{D}$ answers all quantum queries using $(P^{(0)})^{T_j,k}$ and all classical queries using $\mathsf{TEM\text{-}KX}_k[(P^{(0)})^{T_j,k}]$, 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 $P^{(1)}$, where $P^{(1)}(x) = P \circ \mathsf{swap}_{s_0,\, s_1}(x)$. In this case $\mathcal{D}$ perfectly simulates $\mathbf{H}_j^*$ for $\mathcal{A}$. Applying Lemma 3.5 thus gives

$$\begin{aligned}\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^*) = 1]\right| &\leq \sqrt{\varepsilon}\left(1 + \sqrt{q_Q + \log\left(\frac{11\,|F|}{\sqrt{\varepsilon}}\right)}\right) \\ &= \frac{1}{2^{\kappa/2}}\left(1 + \sqrt{q_Q + \log\left(\frac{11\,|\mathcal{T}|\,2^n}{2^{-\kappa/2}}\right)}\right). \qquad (4.7)\end{aligned}$$

Next, we bound the distinguishability of $\mathbf{H}_j^*$ and $\mathbf{H}_j^{**}$. Recall that in $\mathbf{H}_j^*$ the answer to the $(j+1)$st classical query is $y_{j+1} = \mathsf{TEM\text{-}KX}_k[(P^{(1)})^{T_j,k}](t_{j+1}, x_{j+1})$, whereas in $\mathbf{H}_j^{**}$ the response is $y_{j+1} = \tilde{E}_{t_{j+1}}(x_{j+1})$. In $\mathbf{H}_j^*$, we have

$$\begin{aligned} y_{j+1} &\stackrel{\text{def}}{=} \mathsf{TEM\text{-}KX}_k[(P^{(1)})^{T_j,k}](t_{j+1}, x_{j+1}) \\ &= (P^{(1)})^{T_j,k}(s_0) \oplus f_2(t_{j+1}, P(k||0^{n-\kappa})) \\ &= P^{T_j,k}(s_1) \oplus f_2(t_{j+1}, P(k||0^{n-\kappa}))\,. \end{aligned}$$

Since s_1 is uniform and $P^{T_j,k}(\cdot) \oplus f_2(t_{j+1}, P(k||0^{n-\kappa}))$ is a permutation, we conclude that y_{j+1}

is uniform. This is not identical to the distribution of y_{j+1} in $\mathbf{H}_j^{**}$, which is uniform subject to the constraint that $\tilde{E}_{t_{j+1}}$ is a permutation. Define the set $\mathcal{Y}_{j+1} = \{y_i \mid t_i = t_{j+1}\}$, i.e., these are the outputs of $\tilde{E}$ that $\mathcal{A}$ learned from queries with the same tweak t_{j+1} used in the $(j+1)$st query. Bounding the probability that $y_{j+1} \in \mathcal{Y}_{j+1}$ when y_{j+1} is uniform gives an upper bound on the probability that $\mathcal{A}$ can distinguish $\mathbf{H}_j^*$ and $\mathbf{H}_j^{**}$. Thus,

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j^*) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1]\right| \leq \frac{|\mathcal{Y}_{j+1}|}{2^n} \leq \frac{j}{2^n} \leq \frac{j}{2^\kappa}\,. \tag{4.8}$$

Finally, we bound the distinguishability of $\mathbf{H}_j^{**}$ and $\mathbf{H}_j'$. Recall that the difference between these experiments is that from the $(j+1)$st query onward the former uses $(P^{(1)})^{T_j,k}$ while the latter uses $P^{T_{j+1},k}$ (both for the quantum queries of $\mathcal{A}$ and to instantiate TEM-KX for the classical queries of $\mathcal{A}$). Thus, the two experiments are identical if $(P^{(1)})^{T_j,k}$ and $P^{T_{j+1},k}$ are equal. In what follows we upper bound the probability that they are not equal.

Both $(P^{(1)})^{T_j,k}$ and $P^{T_{j+1},k}$ involve $j+1$ swaps: $(P^{(1)})^{T_j,k}$ involves j swaps from the first j queries plus the extra swap by the definition of $P^{(1)}$, whereas $P^{T_{j+1},k}$ involves $j+1$ swaps from the first $j+1$ queries. Since the $(j+1)$st query is a forward query, we have

$$\left(P^{(1)}\right)^{T_j,k}(x) = \overleftarrow{S}_{T_j,P^{(1)},k} \circ \overrightarrow{S}_{T_j,P^{(1)},k} \circ P^{(1)}(x)$$

and

$$(P)^{T_{j+1},k}(x) = \overleftarrow{S}_{T_{j+1},P,k} \circ \overrightarrow{S}_{T_{j+1},P,k} \circ P(x)\,.$$

Let $\mathcal{X} = \{x_1 \oplus f_1(t_1, P(k||0^{n-\kappa})), \ldots, x_j \oplus f_1(t_j, P(k||0^{n-\kappa}))\}$, i.e., $\mathcal{X}$ contains the inputs to P from the first j classical queries of $\mathcal{A}$. Let Bad_0 be the event that $x_{j+1} \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \in \mathcal{X}$

and Bad_1 be the event that $s_1 \in \mathcal{X}$. We upper bound the probabilities of Bad_0, Bad_1, and then show that $(P^{(1)})^{T_j,k} = P^{T_{j+1},k}$ when neither Bad_0 nor Bad_1 occurs.

Since s_1 is $\frac{j}{2^n}$-close to uniform by Eq. (4.8), $\Pr[\mathsf{Bad}_1] \leq \frac{2j}{2^n}$. Bounding the probability of Bad_0 is more complex since we have to consider the tweaks from the first j queries of $\mathcal{A}$. Intuitively, for queries whose tweak was the same as t_{j+1}, we rely on the assumption that $\mathcal{A}$ does not repeat queries; for queries where the tweaks are different, we use the XOR-universality of f_1, f_2. Define

$$\begin{aligned}
\mathcal{X}^{=} &= \{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})) \mid 1 \leq i \leq j,\ t_i = t_{j+1}\} \\
\mathcal{X}^{\neq} &= \{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})) \mid 1 \leq i \leq j,\ t_i \neq t_{j+1}\}.
\end{aligned}$$

These sets partition $\mathcal{X}$ into those inputs using the same tweak as in the $(j+1)$st query ($\mathcal{X}^{=}$) and those using different tweaks ($\mathcal{X}^{\neq}$). Hence,

$$\Pr[\mathsf{Bad}_0] = \Pr[\mathsf{Bad}_0^{=}] + \Pr[\mathsf{Bad}_0^{\neq}]\,,$$

where $\mathsf{Bad}_0^{=}$ is the event that $x_{j+1} \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \in \mathcal{X}^{=}$ and $\mathsf{Bad}_0^{\neq}$ is the event that $x_{j+1} \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \in \mathcal{X}^{\neq}$. For $\mathsf{Bad}_0^{=}$, we have

$$\begin{aligned}
x_{j+1} \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \in \{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})) \mid t_i = t_{j+1}\} \\
\Leftrightarrow \quad x_{j+1} \in \{x_i \mid t_i = t_{j+1}\}\,.
\end{aligned}$$

Since $\mathcal{A}$ does not repeat queries, this means $\Pr[\mathsf{Bad}_0^{=}] = 0$.

For $\mathsf{Bad}_0^{\neq}$, rewriting yields

$$\begin{aligned} & x_{j+1} \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \in \{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})) \mid t_i \neq t_{j+1}\} \\ \Leftrightarrow \quad & x_{j+1} \in \{x_i \oplus f_1(t_i, P(k||0^{n-\kappa})) \oplus f_1(t_{j+1}, P(k||0^{n-\kappa})) \mid t_i \neq t_{j+1}\}\,. \end{aligned}$$

XOR-universality of f_1, together with the fact that $f_1(t, \cdot)$ is a permutation for all t, implies that the mapping $g_{t,t'} : x \mapsto f_1(t, x) \oplus f_1(t', x)$ is a permutation whenever $t \neq t'$. Thus $g_{t_i,t_{j+1}} \circ P$ preserves the min-entropy of $k\|0^{n-\kappa}$ and $\Pr[\mathsf{Bad}_0^{\neq}] \leq |\mathcal{X}^{\neq}|/2^{\kappa} \leq j/2^{\kappa}$. Summarizing,

$$\Pr[\mathsf{Bad}_0] = \Pr[\mathsf{Bad}_0^{=}] + \Pr[\mathsf{Bad}_0^{\neq}] \leq 0 + \frac{|\mathcal{X}^{\neq}|}{2^{\kappa}} \leq \frac{j}{2^{\kappa}}\,.$$

If neither Bad_0 or Bad_1 happens, we have $P^{(1)}(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))) = P(x_i \oplus f_1(t_i, P(k||0^{n-\kappa})))$ for all $1 \leq i \leq j$; furthermore, $P^{T_j,k}(s_1) = P(s_1)$ or, in other words, $P(s_1) = y_{j+1} \oplus f_2(t_{j+1}, P(k||0^{n-\kappa}))$. Therefore,

$$\begin{aligned} \overrightarrow{S}_{T_j,P^{(1)},k} &= \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P^{(1)}(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))} \\ &= \prod_{i=1}^{j} \mathsf{swap}^{1-b_i}_{P(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))} \;=\; \overrightarrow{S}_{T_j,P,k} \end{aligned}$$

and

$$\overleftarrow{S}_{T_j,P^{(1)},k} = \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P^{(1)}(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))}$$
$$= \prod_{i=j}^{1} \mathsf{swap}^{b_i}_{P(x_i \oplus f_1(t_i, P(k||0^{n-\kappa}))),\, y_i \oplus f_2(t_i, P(k||0^{n-\kappa}))} \;=\; \overleftarrow{S}_{T_j,P,k},$$

and so

$$(P^{(1)})^{T_j,k}(x) = \overleftarrow{S}_{j,P^{(1)},k} \circ \overrightarrow{S}_{j,P^{(1)},k} \circ P^{(1)}(x)$$
$$= \overleftarrow{S}_{j,P,k} \circ \overrightarrow{S}_{j,P,k}$$
$$\circ\, \mathsf{swap}_{P(f_1(t_{j+1}, P(k||0^{n-\kappa})) \oplus x_{j+1}),\, y_{j+1} \oplus f_2(t_{j+1}, P(k||0^{n-\kappa}))} \circ P(x)$$
$$= \overleftarrow{S}_{j+1,P,k} \circ \overrightarrow{S}_{j+1,P,k} \circ P(x) \;=\; P^{T_{j+1},k}.$$

Putting everything together, we conclude that

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1] - \Pr[\mathcal{A}(\mathbf{H}'_j) = 1]\right| \leq \Pr[\mathsf{Bad}_0] + \Pr[\mathsf{Bad}_1] \leq \frac{3j}{2^\kappa}\,.$$

Combining the above with Eq. (4.7) and Eq. (4.8) concludes the proof of Lemma 4.13, and hence the proof of Theorem 4.11. □

□

II: Security of TEM

Recall that the tweakable Even-Mansour construction TEM is defined as

$$\mathsf{TEM}_k^{f_1,f_2}[P](t,x) = P(x \oplus f_1(t,k)) \oplus f_2(t,k)\,.$$

Setting $\kappa = n$ and noting that $P(k)$ is uniform when k is uniform (since P is a permutation), Theorem 4.11 yields the following as an easy corollary:

Theorem 4.14. *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. If f_1, f_2 are proper with respect to $\mathcal{T}$, then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{TEM}_k^{f_1,f_2}[P],P} = 1 \right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T},n); \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E},P} = 1 \right] \right| \leq 7 \cdot 2^{-n/2} \cdot \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

We note that Theorem 4.14 is obtained as a corollary of Theorem 4.11 only for $q_Q \geq \max(\log(11|\mathcal{T}|), n)$. While small values of q_Q are not particularly interesting to consider, the theorem can be proven for those using a resampling lemma like Lemma 3.5, but without key expansion.

III: Security of $\mathsf{TEM\text{-}KX1}$

We also consider an alternate method of expanding a key $k \in \{0,1\}^\kappa$ to an effective key of length n, in which we compute $F_P(k) = P(k \| 0^{n-\kappa}) \oplus k \| 0^{n-\kappa}$. This gives rise to $\mathsf{TEM\text{-}KX1}$, a variant of tweakable Even-Mansour defined as

$$\mathsf{TEM\text{-}KX1}_k^{f_1,f_2}[P](t,x) = P(x \oplus f_1(t, F_P(k))) \oplus f_2(t, F_P(k)))\,.$$

We obtain a tighter security bound for this variant than for TEM-KX; this allows us to give a tighter bound for Elephant in Section 4.3.2.

We first show that F_P is a pseudorandom generator, even against adversaries with quantum oracle access to P and P^{-1}.

Lemma 4.15. For any quantum algorithm $\mathcal{A}$ making q_Q quantum queries,

$$\left| \Pr_{\substack{r \leftarrow \{0,1\}^n \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^P(r) = 1 \right] - \Pr_{\substack{k \leftarrow \{0,1\}^\kappa \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^P(P(k||0^{n-\kappa}) \oplus k||0^{n-\kappa}) = 1 \right] \right| \leq \frac{4 \cdot q_Q}{2^{\kappa/2}} .$$

Proof. Given an adversary $\mathcal{A}$, we construct a distinguisher $\mathcal{D}$ for the reprogramming experiment from Lemma 3.1:

Phase 1: $\mathcal{D}$ samples uniform $P \in \mathcal{P}_n$ and $r \in \{0,1\}^n$, and defines a randomized algorithm $\mathcal{B}$ that proceeds as follows:

1. sample uniform $k \in \{0,1\}^\kappa$;
2. output a set of reprogramming pairs B so that P blinded with B is $P^{(B)}(x) = P \circ \mathsf{swap}_{P^{-1}((k||0^{n-\kappa}) \oplus r),\, k||0^{n-\kappa}}$.

Then $\mathcal{D}$ outputs P and $\mathcal{B}$.

Phase 2: $\mathcal{B}$ is run with a uniform $k \in \{0,1\}^\kappa$ to compute B. Let $P_0 = P$ and $P_1 = P^{(B)}$. A uniform $b \in \{0,1\}$ is chosen and $\mathcal{D}$ is given access to P_b (in the forward and inverse directions). $\mathcal{D}$ runs $\mathcal{A}$ with input r and oracle P_b. This phase ends when $\mathcal{A}$ has made its last query and outputs its guess.

Phase 3: $\mathcal{D}$ outputs what $\mathcal{A}$ outputs.

Note that there are at most four reprogrammed points. By construction, it holds that $\Pr_{k \leftarrow \{0,1\}^\kappa}[x \in B_1] \leq 4 \cdot 2^{-\kappa}$. By Lemma 3.1,

$$|\Pr[\mathcal{D} \text{ outputs } 1 \mid b = 0] - \Pr[\mathcal{D} \text{ outputs } 1 \mid b = 1]| \leq 4q_Q \cdot 2^{-\kappa/2} \,. \tag{4.9}$$

When $b = 0$, $\mathcal{D}$ runs $\mathcal{A}^P(r)$ for uniform and independent P, r. When $b = 1$, $\mathcal{D}$ runs $\mathcal{A}^{P_1}(r)$ where P_1 and r are each uniform but are not independent. Indeed,

$$\begin{aligned} P_1(k||0^{n-\kappa}) \oplus k||0^{n-\kappa} &= P(P^{-1}((k||0^{n-\kappa}) \oplus r)) \oplus k||0^{n-\kappa} \\ &= k||0^{n-\kappa} \oplus r \oplus k||0^{n-\kappa} = r \,. \end{aligned}$$

We prove that P_1 is uniform subject to that constraint. Let $\ell = 2^n - 1$, and let $x_1, \ldots, x_\ell$ and $y_1, \ldots, y_\ell$ be arbitrary enumerations of $X = \{0,1\}^n \setminus \{k||0^{n-\kappa}\}$ and $Y = \{0,1\}^n \setminus \{r \oplus k||0^{n-\kappa}\}$, respectively. We show that

$$\Pr[\forall i = 1, \ldots, \ell : P_1(x_i) = y_i] = \frac{1}{(2^n - 1)!} \,.$$

Letting

$$\begin{aligned} \mathbf{A} &= \Pr[P^{-1}((k||0^{n-\kappa}) \oplus r) \notin X] \\ &\quad \cdot \Pr[\forall i = 1, \ldots, \ell : P_1(x_1) = y_i \mid P^{-1}((k||0^{n-\kappa}) \oplus r) \notin X] \\ &= 2^{-n} \cdot \frac{1}{(2^n - 1)!} = \frac{1}{2^n!} \end{aligned}$$

and

$$\begin{aligned}
\mathbf{B} &= \sum_{j=1}^{\ell} \Pr[P^{-1}((k||0^{n-\kappa}) \oplus r) = x_j] \\
&\qquad \cdot \Pr[\forall i \neq j : P(k||0^{n-\kappa}) = y_j \wedge P_1(x_i) = y_i \mid P^{-1}((k||0^{n-\kappa}) \oplus r) = x_j] \\
&= \sum_{j=1}^{\ell} 2^{-n} \cdot \frac{1}{(2^n-1)!} = \frac{\ell}{2^n!} = \frac{2^n-1}{2^n!}\,,
\end{aligned}$$

we have

$$\Pr[\forall i = 1, \ldots, \ell : P_1(x_i) = y_i] = \mathbf{A} + \mathbf{B} = \frac{1}{(2^n-1)!}\,,$$

as desired. The claimed result thus follows from Eq. (4.9). □

The following is an immediate corollary of Theorem 4.14 and Lemma 4.15.

Theorem 4.16. *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. If f_1, f_2 are proper with respect to $\mathcal{T}$, then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^\kappa; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{TEM\text{-}KX1}_k^{f_1,f_2}[P],P} = 1 \right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T},n); \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E},P} = 1 \right] \right| \leq 4 \cdot q_Q 2^{-\kappa/2} + 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

4.3 Applications

In this section we use our results of Section 4.2.2 to show post-quantum security of several lightweight symmetric-key schemes: Chaskey [1], Elephant [2], and a variant of Minalpher [3].

4.3.1 Chaskey

Chaskey [1] is an ISO-standardized lightweight MAC whose construction is based on a specific permutation P that we model as a random permutation. Define $\mathsf{F}^P_{k,k'}(x) = P(x \oplus k) \oplus k'$, i.e., the Even-Mansour cipher based on P. Evaluating Chaskey using key k involves evaluating $\mathsf{F}^P_{k,k}$, $\mathsf{F}^P_{k\oplus k_1,k_1}$, and $\mathsf{F}^P_{k\oplus k_2,k_2}$, where $k_1 = 2k$, $k_2 = 4k$, and multiplication is in the field $GF(2^n)$ with respect to a particular representation of field elements as n-bit strings (see Figure 4.5). Prior work [1] shows that Chaskey is a secure MAC if these three instances of F^P are indistinguishable from three independent random permutations—a notion called *3PRP security*—and also proves 3PRP security of F when P is modeled as a public random permutation. Although this prior work considered classical adversaries only, it is not hard to verify that the proofs carry through to imply security of Chaskey against quantum adversaries making classical MAC queries, so long as 3PRP security of F holds against adversaries making classical queries to the secretly keyed ciphers and quantum queries to P.

As we now show, Theorem 4.14 readily implies 3PRP security of F in the post-quantum setting.

Theorem 4.17. *Let $\mathcal{A}$ be a quantum algorithm making q_C classical queries to its first three oracles and $q_Q \geq 1$ quantum queries to its fourth oracle. Then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n, \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{F}^P_{k,k}, \mathsf{F}^P_{k\oplus k_1,k_1}, \mathsf{F}^P_{k\oplus k_2,k_2}, P} = 1 \right] - \Pr_{R_1,R_2,R_3,P \leftarrow \mathcal{P}(n)} \left[\mathcal{A}^{R_1,R_2,R_3,P} = 1 \right] \right| \leq 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right),$$

where $k_1 = 2k$ and $k_2 = 4k$.

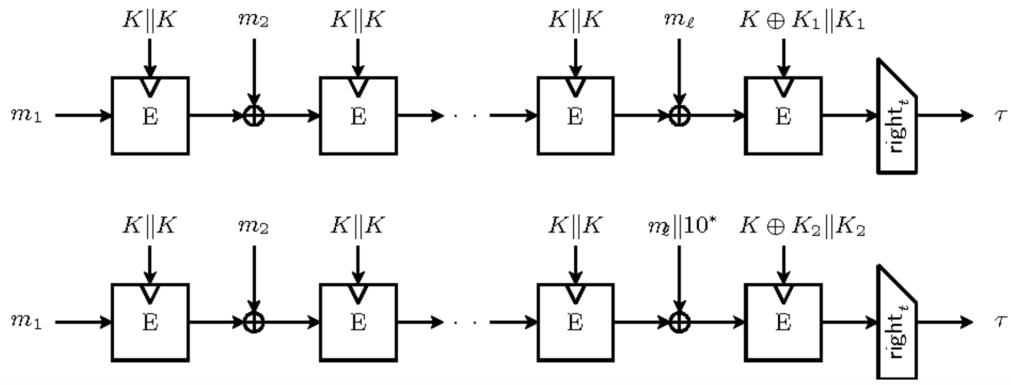


Figure 4.5: Depiction of Chaskey-B: An alternative description of Chaskey based on an Even-Mansour block cipher. The figure is adapted from [1].

Proof. Letting $\mathcal{T} = \{0, 1, 2\} \subset GF(2^n)$ and defining $f_1(t, k) = k \oplus (2tk)$ and $f_2(t, k) = 2^t \cdot k$, we see that

$$\mathsf{TEM}_k^{f_1,f_2}[P](0, x) = P(x \oplus k) \oplus k = \mathsf{F}_{k,k}(x)$$

$$\mathsf{TEM}_k^{f_1,f_2}[P](1, x) = P(x \oplus k \oplus 2k) \oplus 2k = \mathsf{F}_{k\oplus k_1,k_1}(x)$$

$$\mathsf{TEM}_k^{f_1,f_2}[P](2, x) = P(x \oplus k \oplus 4k) \oplus 4k = \mathsf{F}_{k\oplus k_2,k_2}(x)\,.$$

The theorem thus follows from Theorem 4.14 once we verify that f_1, f_2 are proper. Uniformity of f_1 and f_2 follows readily from invertibility of non-zero elements in $GF(2^n)$. Finally, note that

$$f_1(t, k) \oplus f_1(t', k) = 2 \cdot (t \oplus t') \cdot k \text{ and } f_2(t, k) \oplus f_2(t', k) = (2^t \oplus 2^{t'}) \cdot k\,,$$

with $t \oplus t'$ and $2^t \oplus 2^{t'}$ non-zero for distinct t, t'; XOR-universality follows. This concludes the proof of the theorem. □

As discussed earlier, the above theorem in combination with prior results [1] implies post-quantum security (in the random-permutation model) of Chaskey. Below we state a simple ver-

sion of the theorem, leaving out some details and parameters. We formulate MAC unforgeability in terms of a distinguishing experiment in which the adversary is equipped with the Mac_k oracle, and must distinguish the oracle implementing Ver_k from the oracle (denoted by $\perp$) that always rejects. (To exclude trivial attacks, the adversary cannot forward a message/tag pair obtained from the first oracle to the second oracle.)

Theorem 4.18. *Let* $(\mathsf{Mac}, \mathsf{Ver})$ *be the* Chaskey *MAC, and let* $\mathcal{A}$ *be a quantum algorithm making* q_C *classical queries to its first two oracles and* q_Q *quantum queries to its third oracle. Then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{Mac}_k, \mathsf{Ver}_k, P} = 1 \right] - \Pr_{\substack{k \leftarrow \{0,1\}^n \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{Mac}_k, \perp, P} = 1 \right] \right| \leq \mathcal{O}(2^{-n} \cdot q_C) + 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

4.3.2 Elephant

Elephant [2] is a lightweight authenticated encryption scheme with associated data (AEAD) that was a finalist in the NIST lightweight cryptography standardization effort [68]. It is based on a tweakable block cipher we call ELE, which is constructed from a specific permutation P. Prior work [2] proves—in the purely classical setting—that Elephant is secure if ELE is a secure tweakable block cipher, and that ELE is a secure tweakable block cipher if P is modeled as a public random permutation. As with Chaskey, it is straightforward to verify that the former result carries over to the setting of quantum adversaries with classical access to Elephant if ELE is post-quantum secure.

The tweakable block cipher $\mathsf{ELE}[P] : \{0,1\}^{n-s} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ used by Elephant

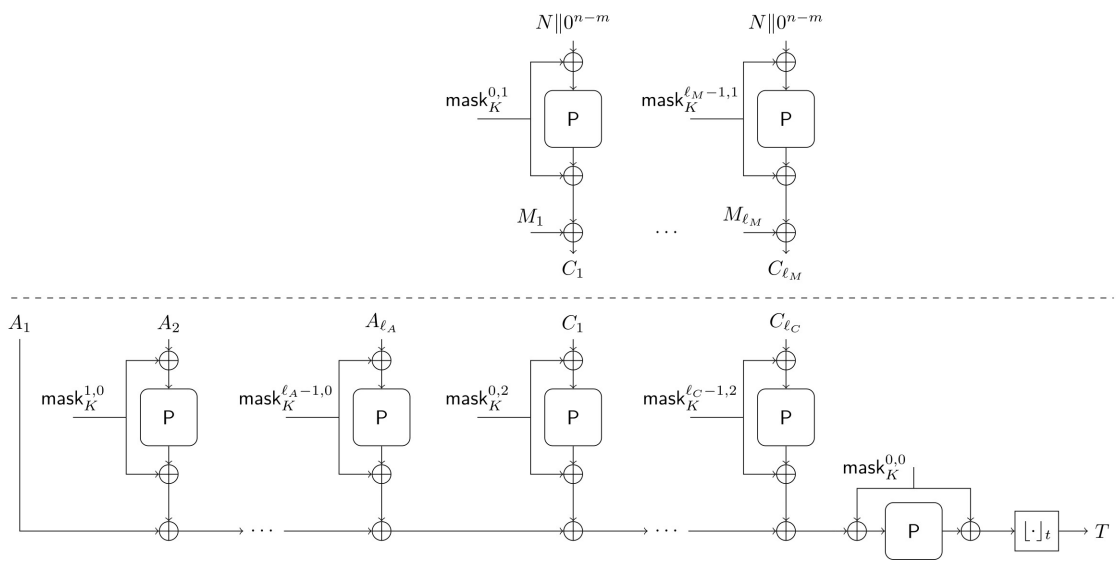


Figure 4.6: Depiction of Elephant. The figure on top illustrates encryption, while the one below depicts authentication. The figure is adapted from [2].

is defined as

$$\mathsf{ELE}[P]_k(t,x) = P(x \oplus f(t, P(k\|0^s))) \oplus f(t, P(k\|0^s))\,, \tag{4.10}$$

where $f : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is a function that is proper with respect to $\mathcal{T}$. (The particular structure of f and $\mathcal{T}$ is not relevant here.) Since ELE is a special case of TEM-KX where $f_1 = f_2 = f$, post-quantum security of ELE follows directly from Theorem 4.11.

Theorem 4.19. *Let* ELE *be as above and let $\mathcal{A}$ be an adversary making q_C classical queries to its first oracle and $q_Q \geq \max\{n, \log(11 \cdot |\mathcal{T}|)\}$ quantum queries to its second oracle. Then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{ELE}[P]_k, P} = 1\right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T},n); \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E}, P} = 1\right] \right| \leq 7 \cdot 2^{-n/2} \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right).$$

As discussed earlier, the above theorem in combination with [2, Theorem B.3] implies post-quantum security (in the random-permutation model) of Elephant. Recall that in the au-

thenticated encryption security experiment, the adversary is tasked with distinguishing the oracles $(\mathsf{Enc}_k, \mathsf{Dec}_k)$ from the pair of oracles in which the first (denoted $\$$) outputs random ciphertexts and the second (denoted $\perp$) always rejects. (Typical restrictions have to be imposed on the adversary to avoid trivial attacks; we do not state these here explicitly.) A fully flexible security theorem for Elephant involves many parameters; for simplicity, we record only a simple version below.

Theorem 4.20. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be the* Elephant *AEAD scheme, and let* $\mathcal{A}$ *be a quantum adversary making a total of* q_C *classical queries to its first two oracles and* $q_Q \geq \max\{n, \log(11 \cdot |\mathcal{T}|)\}$ *quantum queries to its third oracle. Then*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^n; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{Enc}_k, \mathsf{Dec}_k, P} = 1\right] - \Pr_{P \leftarrow \mathcal{P}(n)} \left[\mathcal{A}^{\$, \perp, P} = 1\right] \right| \leq \mathcal{O}(2^{-n} \cdot q_C) + 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C}\right).$$

A variant with a tighter security bound. Next, we consider a slight variant of Elephant for which we can give a tighter security bound. Recall that ELE expands the key via $k \| 0^s \mapsto P(k \| 0^s)$. Here, we instead expand the key via $k \mapsto k || 0^s \oplus P(k || 0^s)$. The tweakable block cipher then becomes

$$\mathsf{ELE\text{-}KX1}[P]_k(t, x) = P(x \oplus f(t, P(k \| 0^s) \oplus k || 0^s)) \oplus f(t, P(k \| 0^s) \oplus k || 0^s). \tag{4.11}$$

Security of the above is then a direct consequence of Theorem 4.16.

Theorem 4.21. *Let* ELE-KX1 *be as above 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. Then

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^{n-s}; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{ELE\text{-}KX1}[P]_k, P} = 1 \right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T}, n); \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E}, P} = 1 \right] \right| \leq 2(q_Q + q_C) \cdot \sqrt{2/2^{n-s}} + 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

The above implies post-quantum security of the variant of Elephant constructed from the cipher in Eq. (4.11) (in place of the cipher from Eq. (4.10)).

4.3.3 (A Variant of) Minalpher

Minalpher [3] is an AEAD scheme[6] that was a second-round candidate in the CAESAR competition. Minalpher is based on a single-round tweakable Even-Mansour cipher that we call MA, which is constructed from a specific permutation P. Prior work in the purely classical setting [3] first proves that MA is a secure tweakable block cipher when P is modeled as a random permutation and then proves, as a consequence, that Minalpher is a secure AEAD scheme. Just as with Elephant and Chaskey, the latter step easily translates to the post-quantum setting if MA is secure in that setting.

We specify MA in more detail. The tweak space $\mathcal{T}$ contains tweaks of the form (flag, N, i, j), where flag is an s-bit string that takes two possible values, $N \in \{0,1\}^{n/2-s}$, and i, j are non-negative integers with $i < 2^\ell$ giving an upper bound on the message length and $j \in \{0,1,2\}$. The tweakable block cipher $\mathsf{MA} : \{0,1\}^{n/2} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ used by Minalpher is then

[6] Minalpher can also be used as a MAC, but here we focus on the AEAD scheme. (see Figure 4.7 for the detailed structure.)

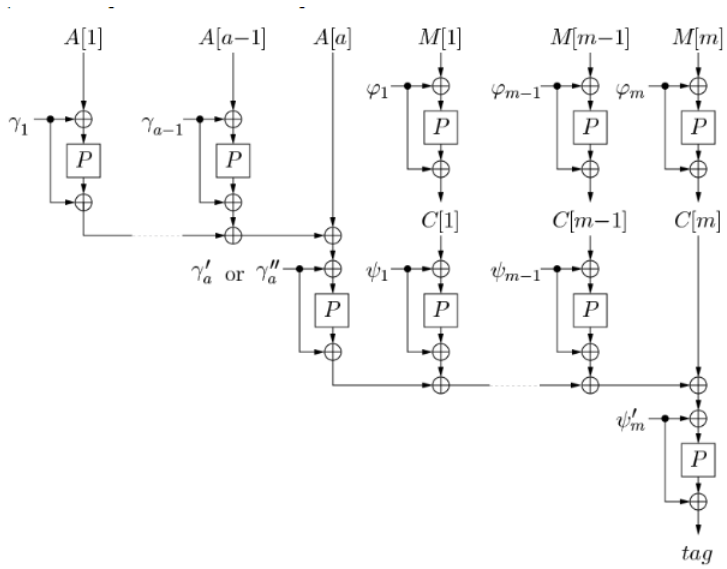


Figure 4.7: Depiction of the AEAD mode of Minalpher. The figure is adapted from [3].

given by

$$\mathsf{MA}_k(t,x) = P(x \oplus L(t,k))) \oplus L(t,k)\,,$$

where

$$L((\mathsf{flag}, N, i, j), k) = y^i(y+1)^j \cdot (P(k||\mathsf{flag}||N) \oplus (k||\mathsf{flag}||N))$$

with y some fixed element of $GF(2^n)$. Note that Minalpher pads the key with part of the tweak (in contrast to Elephant which just pads the key with 0s), which prevents us from using Theorem 4.11 to analyze MA. We thus consider a variant of Minalpher based on a different tweakable block cipher MA' in which the key is padded with 0s. Specifically, we set $s = 1$ so that flag is simply a bit, encode j using two bits, and then fix the lengths of N and i so their combined length

is $n-3$ bits. We then define

$$\mathsf{MA}'_k(t,x) = P(x \oplus f(t,k)) \oplus f(t,k)\,,$$

where

$$f(t,k) = (\mathsf{flag}||N||i||j) \cdot \left(P(k||0^{n/2}) \oplus (k||0^{n/2})\right)\ .$$

Since f is proper, Theorem 4.16 implies:

Theorem 4.22. *Let* MA' *be as above and 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*

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^{n/2};\\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{MA}'_k, P} = 1\right] - \Pr_{\substack{\tilde{E} \leftarrow \mathcal{E}(\mathcal{T},n);\\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\tilde{E}, P} = 1\right] \right| \\ \leq 2(q_Q + q_C) \cdot \sqrt{2/2^{n/2}} + 7 \cdot 2^{-n/2} \left(q_C\sqrt{q_Q} + q_Q\sqrt{q_C}\right).$$

Let $\mathsf{Minalpher}'$ be the variant of $\mathsf{Minalpher}$ constructed by using MA' in place of MA. We can combine the above with classical results about the security of $\mathsf{Minalpher}$ [3] to prove post-quantum security of $\mathsf{Minalpher}'$.

Theorem 4.23. *Let* $(\mathsf{Enc}, \mathsf{Dec})$ *be the* $\mathsf{Minalpher}'$ *AEAD scheme, and let* $\mathcal{A}$ *be a quantum adversary making a total of* q_C *classical queries to its first two oracles and* q_Q *quantum queries to its*

third oracle. Then

$$\left| \Pr_{\substack{k \leftarrow \{0,1\}^{n/2}; \\ P \leftarrow \mathcal{P}(n)}} \left[\mathcal{A}^{\mathsf{Enc}_k, \mathsf{Dec}_k, P} = 1 \right] - \Pr_{P \leftarrow \mathcal{P}(n)} \left[\mathcal{A}^{\$, \perp, P} = 1 \right] \right|$$
$$\leq \mathcal{O}(2^{-n/2} \cdot q_C) + 2(q_Q + q_C) \cdot \sqrt{2/2^{n/2}} + 7 \cdot 2^{-n/2} \left(q_C \sqrt{q_Q} + q_Q \sqrt{q_C} \right).$$

Chapter 5: Two-sided Permutation Inversion problems

5.1 Overview

The permutation inversion problem is defined as follows: given a permutation $\pi : [N] \to [N]$ and an image $y \in [N]$, output the correct pre-image $x := \pi^{-1}(y)$. 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 [21, 30], which is worst-case asymptotically optimal [30, 69, 70].

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. To make the problem nontrivial, we modify the inverse oracle to output 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 [4], as well as in the context of sponge hashing (SHA3) [71]. Moreover, we also consider an adaptive case where the adversary gets to choose part of the pre-image. In this case, the inversion algorithm consists of two phases. The first phase is given a complete description of P, and allowed to output a string $\mu \in \{0,1\}^m$ for $m < n$. The sec-

ond phase is then granted query access to P and asked to invert an image y, sampled uniformly randomly from the set of all strings whose last m bits equal μ.

In the main theorem, we consider the *average-case* setting. The average case means the permutation π and the challenge image y are randomly selected. 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. However, we also show that there exists a random self-reduction so that the average algorithm implies the worst-case algorithm, as [72].

5.2 Reduction from Unstructured Search to Two-sided Permutation Inversion

In this section, we show that given an algorithm that solves the *two-sided permutation inversion problem* (we use TPI to denote such problem in a later paragraph), we can construct another algorithm that solves the unstructured search problem (denoted as $\mathsf{UNIQUESEARCH}$). We start with a few definitions.

Definition 5.1. (TPI_n) Given a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$, let $\mathcal{A}$ be an algorithm which receives an image $y \in \{0,1\}^n$ as input, along with quantum oracle access to π and $\pi_{\perp t}^{-1}$, where $\pi_{\perp t}^{-1} : \{0,1\}^n \times \{0,1\} \to \{0,1\}^n \times \{0,1\}$ is defined by

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

$\mathcal{A}$ outputs "yes" if the first bit of $\pi^{-1}(t)$ is 1, and "no" otherwise.

Note that this is a decision version of the permutation inversion problem; for the search version, $\mathcal{A}$ outputs a n-bit string x and then checks whether $\pi(x) = t$. Such a problem is also considered in [73].

Definition 5.2. (UNIQUESEARCH$_n$) Given a function $f : \{0,1\}^n \to \{0,1\}$, such that f maps at most one element to 1. Consider an algorithm $\mathcal{A}$ having quantum query access to f, $\mathcal{A}$ outputs "yes" if $f^{-1}(1)$ is non-empty and "no" otherwise.

We consider the special case where function f is restricted to map at most one element to 1, which can be viewed as the hardest unordered search instance. Therefore, we name such a problem as UNIQUESEARCH. Moreover, this is also a decision algorithm.

It is well-known that given an algorithm for UNIQUESEARCH, it is not hard to construct an algorithm that solves TPI. More specifically, we could define a function f such that $f(i) = 1$ when $\pi(i) = t$. Classically, this function can be evaluated with one query to an oracle for π; in the quantum world, this could also be done with an additional quantum query, which we'll explain later. Using Grover's search algorithm, we could solve the problem with $O(\sqrt{2^n})$ queries to an oracle for π. In [72], Nayak gave a reduction in the inverse direction, i.e., constructing an UNIQUESEARCH algorithm from a TPI algorithm. In Nayak's reduction, the adversary $\mathcal{A}$ only has forward query access to the permutation oracle. In this work, we extended Nayak's reduction, where the adversary has both forward and inverse access to the permutation oracle. Moreover, our reduction also works for an adaptive version of the permutation inversion problem, where $\mathcal{A}$ can select part of the pre-image. Before discussing the reduction, it is worth noting that our reduction works for the average case, where both function f and permutation π are chosen from a uniform distribution, and the "yes" and "no" instances are equally distributed. We define the

distribution error of a decision algorithm.

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

Given the distributional error (p_0, p_1) of an algorithm, we let $\frac{1}{2}(p_0 + p_1)$ be the total error of that algorithm given that the "yes" and "no" cases are equally distributed. We are now ready to define our reduction.

Theorem 5.4. *Let $\mathcal{A}$ be a quantum algorithm that solves TPI_n with total error $\frac{1}{2} - \delta$ $(0 < \delta < \frac{1}{2})$ on the uniform distribution over permutations on $\{0,1\}^n$, with T quantum queries to the permutation oracles. Then there exists a quantum algorithm $\mathcal{B}$ that can solve $\mathsf{UNIQUESEARCH}_{n-1}$ with at most $2T$ quantum queries with distributional error $\left(\frac{1}{2} - \delta, \frac{1}{2}\right)$.*

Adaptive case

As we claimed above, Theorem 5.4 also works for the adaptive case, where the adversary can adaptively select part of the pre-image. To illustrate how this works, we define the adaptive version of TPI.

Definition 5.5. ($\mathsf{aTPI}_{m,n}$) Given a permutation $\pi : \{0,1\}^n \to \{0,1\}^n$. Let $\mathcal{A}$ be a quantum algorithm that does the following:

1. $\mathcal{A}$ has access to the whole codebook of π, and then $\mathcal{A}$ outputs a m bit string μ , where $m < n$. Then $\mathcal{A}$ losses access to π.

2. The challenger generates a random image $t \in \{0,1\}^n$ by first sampling a random string $x \leftarrow \{0,1\}^{n-m}$ and then letting $t = \pi(x \| \mu)$;

3. $\mathcal{A}$ receives t, along with quantum oracle access to π and $\pi^{-1}_{\perp t}$. $\mathcal{A}$ outputs "yes" if the first bit of $\pi^{-1}(t)$ is 1, and "no" otherwise.

Compare with Definition 5.1, it is straightforward that the following holds.

Corollary 5.6. TPI_n is the specially case of $\mathsf{aTPI}_{m,n}$ when $m = 0$.

Theorem 5.7. *Let $\mathcal{A}$ be a quantum algorithm that solves $\mathsf{aTPI}_{m,n}$ with total error $\frac{1}{2} - \delta$ ($0 < \delta < \frac{1}{2}$) on the uniform distribution over permutations on $\{0,1\}^n$, with T quantum queries to the permutation oracles. Then there exists a quantum algorithm $\mathcal{B}$ that can solve $\mathsf{UNIQUESEARCH}_{n-m-1}$ with at most $2T$ quantum queries with distributional error $\left(\frac{1}{2} - \delta, \frac{1}{2}\right)$.*

Proof. Given a quantum algorithm $\mathcal{A}$ as Theorem 5.7, we construct another algorithm $\mathcal{B}$ which solves the $\mathsf{UNIQUESEARCH}_{n-m-1}$ problem. For any uniform image $t \in \{0,1\}^n$, define the "no" and "yes" instances sets (corresponding to the image t) of $\mathsf{aTPI}_{m,n}$:

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

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

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

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

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

2. $\mathcal{B}$ then runs the first step of $\mathcal{A}$ and receives a string $\mu \in \{0,1\}^m$ from $\mathcal{A}$.

3. Let $t = \pi(s\|\mu)$, and then it follows that if $s|_0 = 0$, $\pi \in \pi_{t,0}$, and otherwise $\pi \in \pi_{t,1}$. Since s is uniformly sampled, we have $\Pr[\pi \in \pi_{t,0}] = \Pr[\pi \in \pi_{t,1}] = \frac{1}{2}$.

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

$$h_{f,\pi,t,\mu}(i\|j\|u) = \begin{cases} t & \text{if } i = 1 \text{ and } f(j) = 1, u = \mu, \\ \pi(i\|j\|u) & \text{otherwise.} \end{cases} \tag{5.1}$$

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

$$h_{f,\pi,t,\mu}(i\|j\|\mu) = \begin{cases} t & \text{if } i = 0 \text{ and } f(j) = 1, u = \mu, \\ \pi(i\|j\|u) & \text{otherwise.} \end{cases} \tag{5.2}$$

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

$$h^{-1*}_{f,\pi,t,\mu}(k||b) = \begin{cases} \pi^{-1}(k)\|0 & \text{if } b = 0 \text{ and } k \neq t, \\ 1\|1 & \text{otherwise.} \end{cases} \tag{5.3}$$

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

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

Let δ_1 be the error probability of $\mathcal{A}$ in the YES case and δ_0 be that in the NO case of $\mathsf{aTPI}_{m,n}$. Since s is uniform random and then $\Pr[\pi \in \pi_{t,0}] = \Pr[\pi \in \pi_{t,1}] = 1/2$, it follows that

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

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

$\mathcal{A}^{h,h^{-1*}}(t) = \mathcal{A}^{\pi,\pi^{-1}_{\perp t}}(t)$. Therefore,

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

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

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

where the third equality comes from the fact that no matter $\pi \in \pi_{t,0}$ or $\pi \in \pi_{t,1}$, the corresponding

h is uniform in $Q_{t,\mu}$ and then can be viewed as uniformly generated from $Q_{t,\mu}$. Since $\mathcal{A}$ is granted with oracle access to h, both conditions can be changed to $h_{f,\pi,t,\mu} \xleftarrow{\$} Q_{t,\mu}$.

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

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

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

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

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

operations

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

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

Proof of Theorem 5.4. The proof follows from Theorem 5.7, with $m = 0$. □

5.3 Lower Bound for the Two-sided Permutation Inversion Problem

In Theorem 5.4 and Theorem 5.7, we show that the two-sided permutation inversion problem is as hard as the unstructured search problem. This section gives a tight bound for the two-sided permutation inversion problem.

For the unstructured search problem, the optimal lower bound of $\Omega(\sqrt{2^n})$ was given by Ambainis [30] using the adversary method.

Theorem 5.8 (Theorem 1 in [30]). *Any quantum algorithm that solves the* $\mathsf{UNIQUESEARCH}_n$ *with probability* ε *uses at least* $\frac{1}{2}(1-2\sqrt{\varepsilon(1-\varepsilon)})\sqrt{2^n-1}$ *queries.*

Note that Theorem 5.8 gives the worst-case lower bound, while Theorem 5.4 and Theorem 5.7 talk about the average-case algorithm. Fortunately, Nayak [72] gave a self-reduction algorithm, showing that the average-case algorithm for the unstructured search algorithm implies the worst-case algorithm.

Lemma 5.9 (Lemma 2.2 in [70]). Suppose $\mathcal{B}$ is an algorithm for $\mathsf{UNIQUESEARCH}_n$ with distributional error at most $\frac{1}{2}-\delta$ on the "no" case and at most $\frac{1}{2}$ on the "yes" case for some $\delta \in (0, 1/2]$. Then there is an algorithm for $\mathsf{UNIQUESEARCH}_n$ that makes the same number of queries as $\mathcal{B}$, but has worst-case error at most

$$\delta' = \frac{\max\left\{\frac{1}{2}-\delta, \frac{1}{2}\right\}}{1+|\frac{1}{2}-\delta-\frac{1}{2}|} = \frac{1}{2(1+\delta)} < \frac{1}{2}.$$

Now we are ready to give the lower bound of our two-sided permutation inversion problem.

Theorem 5.10. *Let* $\mathcal{A}$ *be a quantum algorithm that solves* $\mathsf{aTPI}_{m,n}$ *with distribution error* $\frac{1}{2}-\delta$ *(*$0 < \delta < \frac{1}{2}$*) on the uniform distribution over permutations on* $\{0,1\}^n$*, with* T *quantum queries to the permutation oracles. Then* $T^2 \geq \widetilde{\Omega}(\delta \cdot 2^{n-m})$.

Proof. By Theorem 5.7, we get a $2T$-query algorithm for $\mathsf{UNIQUESEARCH}_{n-m-1}$ with distributional error $(\frac{1}{2}-\delta, \frac{1}{2})$, which implies $\mathsf{UNIQUESEARCH}_{n-m-1}$ with worst-case error $\delta' = 1/(2(1+\delta))$.

From Lemma 5.9, we have

$$T \geq \frac{1}{4}(1 - 2\sqrt{\delta'(1-\delta')})\sqrt{2^{n-m-1} - 1},$$

which concludes the proof. □

We note that with non-adaptive $\mathcal{A}$, i.e. $m = 0$, the above bound reduces to query lower bound $T^2 \geq \widetilde{\Omega}\left(\delta \cdot 2^n\right)$.

5.4 Applications

In this section, we give a plausible security model for symmetric-key encryption and a scheme whose security in that model relies on the hardness of our two-sided permutation inversion problem.

We start with the attack models. In Definition 2.7, we introduced an attack model where the adversary can choose m_0 and m_1 but only learns the ciphertext of one of those messages. Additionally, there are no further interactions. However, more potent models exist where the adversary possesses superior capabilities, resulting in more robust security notions. Indistinguishability security notions (IND) usually consider two types of attacks: chosen-plaintext attack (CPA) and chosen-ciphertext attack (CCA). Furthermore, an even stronger model exists where the adversary could choose the random coins of Enc. In their study of quantum cryptography, Boneh and Zhandry [74] introduced a quantum analog of classical CPA and CCA, known as QCPA and QCCA. For the purposes of this discussion, we will be using this model where we consider QPT adversaries. Note that if the QPT adversary with quantum oracle access is replaced by the PPT

adversary with only classical oracle access, all the concepts revert to the classical case. All the definitions of QCPA and QCCA are subject to this rule.

Quantum Chosen-plaintext Attacks (QCPA). Quantum chosen-plaintext attacks allow adversaries to exert (partial) control over the content encrypted by the honest parties. Specifically, the adversary is aware of the encrypted messages, although it lacks the authority to select the randomness. To model quantum chosen-plaintext attacks, we utilize an indistinguishability experiment structured as follows:

Definition 5.11 (The quantum CPA indistinguishability experiment $\mathsf{QCPA}_{\mathcal{A},\Pi}$(n) [4, 74])**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme and $\mathcal{A}$ be a QPT adversary, the experiment proceeds as follows:

1. $k \leftarrow \mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given input 1^n and quantum oracle access to $\mathsf{Enc}_k(\cdot)$ and outputs a pair of messages m_0, m_1 of the same length.
3. A uniform bit $b \in \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$. We refer to c as the *challenge ciphertext*.
4. $\mathcal{A}$ continues to have quantum oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a bit b'.
5. The experiment outputs 1 if $b = b'$, and outputs 0 otherwise.

Definition 5.12. A symmetric-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is QCPA-secure or $\mathsf{IND\text{-}QCPA}$ if for all QPT adversaries $\mathcal{A}$ there is a negligible function $\mathsf{negl}(n)$ such that for all n,

$$\Pr[\mathsf{CPA}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

The above experiment is usually called QCPA2, and the corresponding security notion is IND-QCPA2. Relatively, if $\mathcal{A}$ doesn't get quantum oracle access to $\mathsf{Enc}_k(\cdot)$ and just outputs m_0 and m_1 before receiving the challenge ciphertext c, it is referred to as QCPA1 and IND-QCPA1. Moreover, when Enc is randomized, the oracle uses fresh *random coins* each time it answers a query. In a later section, we will introduce a type of attack where the adversary can query and potentially select random coins.

Quantum Chosen-ciphertext Attacks (QCCA). In *quantum chosen-ciphertext attacks*, the adversary not only has quantum access to the encryption oracle (as QCPA) but also can obtain the decryption of the ciphertexts of its choice. In other words, the adversary has access to both the encryption and decryption oracle. Similar to QCPA, we employ an indistinguishability experiment to model QCCA:

Definition 5.13 (The quantum CCA indistinguishability experiment $\mathsf{QCCA}_{\mathcal{A},\Pi}$(n) [4, 74])**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme and $\mathcal{A}$ be a QPT adversary, the experiment proceeds as follows:

1. $k \leftarrow \mathsf{Gen}(1^n)$.

2. The adversary $\mathcal{A}$ is given input 1^n and quantum oracle access to $\mathsf{Enc}_k(\cdot)$ and $\mathsf{Dec}_k(\cdot)$ and outputs a pair of messages m_0, m_1 of the same length.

3. A uniform bit $b \in \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.

4. $\mathcal{A}$ continues to have quantum oracle access to $\mathsf{Enc}_k(\cdot)$ and $\mathsf{Dec}_{k,\perp_c}(\cdot)$, but is not allowed to query the decryption oracle on c. $\mathcal{A}$ outputs a bit b'.

5. The experiment outputs 1 if $b = b'$, and outputs 0 otherwise.

Definition 5.14. A symmetric-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is QCCA-secure or IND-QCCA if for all QPT adversaries $\mathcal{A}$ there is a negligible function $\mathsf{negl}(n)$ such that for all n,

$$\Pr[\mathsf{CCA}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

Similar to the notions for QCPA, Definition 5.13 and 5.14 both pertain to QCCA2. Similarly, if $\mathcal{A}$ doesn't have quantum oracle access to Enc and Dec before the challenge phase (step 3), it is denoted as QCCA1. It's important to note that $\mathcal{A}$ has full quantum access to Enc, but not Dec, after the challenge phase. If $\mathcal{A}$ gets full access to Dec, it could simply query the challenge ciphertext c and obtain the answer. Therefore, we do not permit $\mathcal{A}$ to query c directly, but rather we *blind* $\mathcal{A}$ on c. In this context, a *blinded* oracle is typically defined as follows:

$$\mathsf{Dec}_{k,\perp c}(x) = \begin{cases} \mathsf{Dec}_k(x) & \text{if } x \neq c \\ \perp & \text{otherwise.} \end{cases}$$

Quantum Chosen Randomness Attacks (QCRA). Until now we have talked about two types of attacks: *chosen-plaintext attacks* and *chosen-ciphertext attacks*. We have also seen the security model that is used to prevent certain types of attacks. In this model, the adversary has access to an oracle that encrypts a given message m using random coins that are generated uniformly, for example, by pseudorandom number generators (PRNG). This creates the potential for even stronger attacks if the random coins are inadequately generated or under partial adversarial control. In the worst case, the adversary may even have complete control over the *random coins* that will actually be used to encrypt the message. Such an attack was first introduced by Karama

and Katz [75], called *chosen-randomness attack* (CRA). Such chosen-randomness attacks can be combined with CPA and CCA, which leads to CPRA and CCRA. In this work, we study the quantum analog of chosen-randomness attacks where the adversary could have quantum oracle access to the encryption oracle (and decryption oracle for CCA). We define such attacks as QCPRA and QCCRA, which stand for "*quantum chosen-plaintext randomness-access attack*" and "*quantum chosen-ciphertext randomness-access attack*." In fact, we consider the strongest attack model, which is QCCRA2. To begin with, we define the security notions as follows:

Definition 5.15. (OW-QCCRA2-v1) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme. We say that Π is OW-QCCRA2-v1 if the advantage for any QPT adversary $\mathcal{A}$ in the following experiment is at most negligible:

1. A key k is generated by running $\mathsf{Gen}(1^n)$;
2. Uniform $x \in \mathcal{M}$ and $r \in \mathcal{R}$ are chosen, and a challenge ciphertext $c = \mathsf{Enc}_k(x; r)$ is computed and given to $\mathcal{A}$;
3. $\mathcal{A}$ gets quantum oracle access to $\mathsf{Enc}_k(\,\cdot\,;\,\cdot\,)$ and $\mathsf{Dec}_k^{\perp c}(\cdot)$. Eventually, it outputs a bit b. Suppose $\mathcal{A}$ makes $T(n)$ quantum queries.
4. The experiment outputs 1, if $b = x|_0$, and 0 otherwise.

In some scenarios, the adversary $\mathcal{A}$ also gets to choose the part of the pre-image; such an adaptive adversary is considered in the following definition.

Definition 5.16. (OW-QCCRA2-v2) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme. We say that Π is OW-QACCRA-v2 if the advantage for any QPT $\mathcal{A}$ in the following experiment is at most negligible:

1. A key k is generated by running $\mathsf{Gen}(1^n)$;

2. $\mathcal{A}$ gets access to the whole codebook of $\mathsf{Enc}_k(\,\cdot\,;\,\cdot\,)$ and $\mathsf{Dec}_k(\cdot)$, and then outputs a string $\mu \in \{0,1\}^*$;

3. Uniform $x \in \mathcal{X}$ and $r \in \mathcal{R}$ are chosen, and a challenge ciphertext $c = \mathsf{Enc}_k(x||\mu; r)$ is computed and given to $\mathcal{A}$;

4. $\mathcal{A}$ gets quantum oracle access to $\mathsf{Enc}_k(\,\cdot\,;\,\cdot\,)$ and $\mathsf{Dec}_k^{\perp c}(\cdot)$. Eventually, it outputs a bit b. Suppose $\mathcal{A}$ makes $T(n)$ quantum queries.

5. The experiment outputs 1, if $b = x|_0$, and 0 otherwise.

Note that OW denotes *one-way*. Different from the security games for standard IND- security notions (presented in Section 2), the experiments for OW- security notions aim to invert the pre-image instead of distinguishing which message is encrypted.

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

- Gen is given 1^n and outputs a uniformly random permutation π on $\{0,1\}^{2n}$;

- Enc is given $x \in \{0,1\}^n$ and $r \in \{0,1\}^n$, and outputs $c := \pi(x||r)$

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

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

- Gen takes as input a security parameter 1^n and returns a key $k \in \{0,1\}^n$ for P_k;

- Enc is given key $k \in \{0,1\}^n$, $x \in \{0,1\}^n$ and $r \in \{0,1\}^n$, and outputs $c := P_k(x||r)$;

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

Next, we will show that the PRP scheme is OW-QCCRA2-v2 by the following theorem.

Theorem 5.17. *The* PRP *scheme is* OW-QCCRA2-v2. *In other words, for any quantum polynomial time (QPT) adversary $\mathcal{A}$ who makes $T(n)$ quantum queries in the post-challenge phase, it holds that*

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

Proof. Given an adversary $\mathcal{A}$ that attacks RP scheme in the OW-QCCRA2-v2 experiment Π'. The only difference between Π and Π' is that Π uses P_k, a ε-Qsecure PRP as in Definition 2.5, while Π' uses a random permutation π. Suppose $\mathcal{A}$ makes $T(n)$ quantum queries after receiving the challenge ciphertext; we can construct another algorithm $\mathcal{B}$ which solves $\mathsf{aTPI}_{m,n}$, with $T(n)$ queries to the permutation oracles.

1. A random permutation $\pi : \{0,1\}^{2n} \to \{0,1\}^{2n}$, a random image $y \leftarrow \{0,1\}^n$, and a random string $r \leftarrow \{0,1\}^n$ are sampled;

2. $\mathcal{B}$ is given the whole permutation table of π and grants $\mathcal{A}$ unlimited access to π and π^{-1}. Then $\mathcal{B}$ gets back a string $\mu \in \{0,1\}^m$.

3. $\mathcal{B}$ receives as input an image t where $t = \pi(x||\mu||r)$ for a random $x \in \{0,1\}^{n-m}$, and quantum oracle access to π and $\pi^{-1}_{\perp t}$. And then, $\mathcal{B}$ directly passes t with two oracles to $\mathcal{A}$, runs $\mathcal{A}$, gets back a bit b, and outputs it.

4. If $b = \pi^{-1}(t)|_0$, output 1; otherwise output 0.

It trivially follows that

$$\Pr[\mathsf{Exp}^{\text{OW-QCCRA2-v2}}_{\mathcal{A},\Pi'}(1^n) = 1] \leq \Pr[\mathsf{aTPI}_{m,n}(1^n) = 1] = \frac{1}{2} + \delta.$$

Where $\delta \leq O(T^2 \cdot 2^{-(m+n)})$ by Theorem 5.10.

Moreover, for all QPT $\mathcal{A}$, there exists a negligible function ε such that

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

It directly follows that

$$\begin{aligned}\Pr[\mathsf{Exp}^{\text{OW-QCCRA2-v2}}_{\mathcal{A},\Pi}(1^n) = 1] &\leq \Pr[\mathsf{Exp}^{\text{OW-QCCRA2-v2}}_{\mathcal{A},\Pi'}(1^n) = 1] + \varepsilon \cdot \mathsf{poly}(T) \\ &\leq \Pr[\mathsf{aTPI}_{m,n}(n) = 1] + \varepsilon \cdot \mathsf{poly}(T) \\ &= \frac{1}{2} + \delta + \varepsilon \cdot \mathsf{poly}(T).\end{aligned}$$

For all QPT $\mathcal{A}$, which makes a polynomial number of quantum queries T, both δ and $\varepsilon \cdot \mathsf{poly}(n)(T)$ are negligible. □

When $m = 0$, we can directly get the one-wayness of the non-adaptive case, i.e., OW-QCCRA2-v1.

Corollary 5.18. The *PRP* scheme is OW-QCCRA2-v1.

Chapter 6: Conclusion and Outlook

6.1 Conclusion

We first studied the security of symmetric cryptographic primitives against quantum adversaries in the query complexity model. Due to Grover's algorithm, symmetric primitives suffer from reduced ideal security in the quantum world. This is much less devastating than many asymmetric primitives, which can be completely broken with Shor's algorithm. To maintain security against Grover's attack, one immediate solution is to double the key length. However, recent works have shown that generic efficient attacks exist against symmetric constructions and modes of operations. That raises a crucial question: Can we prove the security of symmetric schemes in the quantum world?

To answer this question, the first step is to understand the attack models. Quantum attacks can be categorized into two models for symmetric primitives. In the Q1 model, an adversary gets quantum access to public primitives, such as public permutations, through her local quantum computer. However, she could only get classical access to the online primitives, such as the block ciphers. This model can be understood as a *quantum computation classical communication* model. In the Q2 model, an adversary has quantum access to all the primitives, meaning that in this model, an adversary can achieve *quantum computation and quantum communication*. The Q2 model is more powerful, but the Q1 model is more realistic and achievable, at least in the near

future.

We studied the post-quantum security of symmetric primitives in the Q1 model, starting with a fundamental and very important cipher, the Even-Mansour cipher. We then studied its tweakable version and applications. We provide lower bounds of these symmetric primitives against quantum attacks by upper bounding the adversary's distinguish advantage through hybrid approaches. Our framework leads to many applications, including lightweight symmetric cryptoschemes that are candidates in the NIST lightweight standardization (Elephant and Minalpher) and ISO standardization (Chaskey) efforts. Moreover, FAEST, a digital signature scheme in the current NIST PQC standardization round, uses Even-Mansour as a variant of their encryption scheme to get better performance. Therefore, our framework ensures post-quantum security.

We then studied the query complexity of the two-sided permutation inversion problem against quantum attacks. Moreover, we also consider the adaptive case where the adversary could select part of the pre-image in advance. We get an optimal lower bound through a reduction from the unique search problem. Studying the two-sided permutation inversion problem is important due to its relevance to the security of encryption schemes against chosen-ciphertext attacks and sponge hashing. As mentioned in Section 5.4, our work implies the one-wayness of a variant of the CCA security scheme.

6.2 Future Works

In this section, we will present some future projects that we believe will be of great interest.

First, proving security in the Q1 model is still challenging; there are, at present, only a limited number of positive results about security in this model. Our framework demonstrates the

post-quantum security of Even-Mansour and its extensions, opening up new possibilities in this field. An intriguing area for future research is to expand our work to the ideal cipher model. Let $\mathcal{E}(m, n)$ be the set of all functions $E : m \times \{0,1\}^n \to \{0,1\}^n$ such that $E(t, \cdot)$ is a permutation on $\{0,1\}^n$ for all $t \in m$. In the *ideal-cipher model*, a cipher $E \leftarrow \mathcal{E}(m, n)$ is sampled uniformly and then provided as an oracle, in both the forward and inverse directions, to all parties. This model can be used to generate secure key-length extension schemes. For example, the ideal cipher extension of the (tweakable) Even-Mansour cipher is the (tweakable) FX construction [76]. FX has been considered in several applications, including NIST lightweight candidates PRINCE and PRIDE [77, 78]. Jaeger et al. [46] have proved the security of the FX construction in the Q1 model, but against a restricted adversary who could only make classical queries before querying the quantum oracles. Our immediate future work is to use the framework to prove the security of FX against a general and adaptive adversary.

Another very interesting direction is the sponge construction. Previous work [79, 80] studied the post-quantum security of the sponge construction where the block function is either a random function or a (non-invertible) random permutation. However, as the core permutation in SHA3 [81] is public and efficiently invertible, the "right setting" of theoretical study is one in which the block function consists of an invertible permutation. This setting is far less understood, and establishing the security of the sponge in this setting is a major open problem in post-quantum cryptography. Our results on two-sided permutation inversion may serve as a stepping stone towards this goal. Additionally, we can prove that PRP is OW-QCCRA2, but we cannot achieve the stronger security notion, which is IND-QCCRA2. It is worth exploring the possibility of achieving IND-QCCRA2 against quantum adversaries.

Bibliography

[1] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In *Selected Areas in Cryptography (SAC)*, volume 8781 of *LNCS*, pages 306–323. Springer, 2014.

[2] Tim Beyne, Yu Long Chen, Christoph Dobraunig, and Bart Mennink. Elephant v2. Technical report, NIST, 2021. `https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/documents/finalist-round/updated-spec-doc/elephant-spec-final.pdf`.

[3] Yu Sasaki, Yosuke Todo, Kazumaro Aoki, Yusuke Naito, Takeshi Sugawara, Yumiko Murakami, Mitsuru Matsui, and Shoichi Hirose. Minalpher v1.1, 2015. Available ay `https://competitions.cr.yp.to/caesar-submissions.html`.

[4] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[5] C. E. Shannon. Communication theory of secrecy systems*. *Bell System Technical Journal*, 28(4):656–715, Oct 1949. ISSN 0005-8580. doi: 10.1002/j.1538-7305.1949.tb00928.x. URL `http://dx.doi.org/10.1002/j.1538-7305.1949.tb00928.x`.

[6] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[7] Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[8] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 227–240. 2019.

[9] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 25–32, New York, NY, USA, 1989. ACM. ISBN 0-89791-307-8. doi: 10.1145/73007.73010. URL `http://doi.acm.org/10.1145/73007.73010`.

[10] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, page 43. ACM, 1989.

[11] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC '90: Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, New York, NY, USA, 1990. ACM Press. ISBN 0897913612. doi: 10.1145/100216.100269. URL `http://dx.doi.org/10.1145/100216.100269`.

[12] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, FOCS '94, pages 124–134, Washington, DC, USA, 1994. IEEE Computer Society. ISBN 0-8186-6580-7. doi: 10.1109/SFCS.1994.365700.

[13] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.

[14] Daniel R. Simon. On the power of quantum computation. *SIAM J. Computing*, 26(5): 1474–1483, 1997.

[15] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412):eaam9288, 2018.

[16] Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004. doi: 10.1016/j.ipl.2004.01.024.

[17] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[18] Corporate Nist. The digital signature standard. *Communications of the ACM*, 35(7):36–40, 1992.

[19] Eli Biham and Adi Shamir. Differential cryptanalysis of des-like cryptosystems. *Journal of CRYPTOLOGY*, 4:3–72, 1991.

[20] Akinori Hosoyamada and Yu Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *Topics in Cryptology—Cryptographers' Track at the RSA Conference (CT-RSA) 2018*, volume 10808 of *LNCS*, pages 198–218. Springer, 2018.

[21] 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.

[22] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. Quantum attacks without superposition queries: The offline Simon's algorithm. In *Advances in Cryptology—Asiacrypt 2019, Part I*, volume 11921 of *LNCS*, pages 552–583. Springer, 2019.

[23] Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.

[24] D Augot, L Batina, D Bernstein, J Bos, J Buchmann, W Castryck, O Dunkelman, T Guneysu, S Gueron, A Hulsing, et al. Post-quantum cryptography for long-term security. *PQCRYPTO, Eindhoven, The Netherlands, Rep. ICT-645622*, 2015.

[25] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[26] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. doi: 10.1137/S0097539796300933. URL `http://dx.doi.org/10.1137/S0097539796300933`.

[27] Gorjan Alagic, Chen Bai, Jonathan Katz, and Christian Majenz. Post-quantum security of the Even-Mansour cipher. In *Advances in Cryptology—Eurocrypt 2022, Part III*, volume 13277 of *LNCS*, pages 458–487. Springer, 2022.

[28] Gorjan Alagic, Chen Bai, Jonathan Katz, Christian Majenz, and Patrick Struck. Post-quantum security of tweakable even-mansour, and applications. Cryptology ePrint Archive, Paper 2022/1097, 2022. URL `https://eprint.iacr.org/2022/1097`. `https://eprint.iacr.org/2022/1097`.

[29] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 409–426. Springer, 2006.

[30] Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002.

[31] John S Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195, 1964.

[32] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

[33] Marc Fischlin and Arno Mittelbach. An overview of the hybrid argument. *Cryptology ePrint Archive*, 2021.

[34] Mark Zhandry. How to construct quantum random functions. In *Proceedings of the 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 679–687, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4874-6. doi: 10.1109/FOCS.2012.37.

[35] Mark Zhandry. A note on quantum-secure prps. *arXiv preprint arXiv:1611.05564*, 2016.

[36] Phillip Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 98–107, 2002.

[37] Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In *Adv. in Cryptology—Eurocrypt 2020, Part III*, volume 12107 of *LNCS*, pages 788–817. Springer, 2020. doi: 10.1007/978-3-030-45727-3_27.

[38] Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. In *Advances in Cryptology—Asiacrypt 2021, Part I*, volume 13090 of *LNCS*, pages 637–667. Springer, 2021. Available at `https://eprint.iacr.org/2020/1361`.

[39] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Adv. in Cryptology—Crypto 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, 2019. doi: 10.1007/978-3-030-26951-7_9.

[40] Ryan O'Donnell and Ramgopal Venkateswaran. The quantum union bound made easy, 2021. Available at `https://arxiv.org/abs/2103.07827`.

[41] Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–161, 1997. doi: 10.1007/s001459900025.

[42] Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The Even-Mansour scheme revisited. In *Advances in Cryptology—Eurocrypt 2012*, volume 7237 of *LNCS*, pages 336–354. Springer, 2012.

[43] Hidenori Kuwakado and Masakatu Morii. Security on the quantum-type Even-Mansour cipher. In *Proc. International Symposium on Information Theory and its Applications*, pages 312–316. IEEE, 2012.

[44] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *Adv. in Cryptology—Crypto 2016, Part II*, volume 9815 of *LNCS*, pages 207–237. Springer, 2016. doi: 10.1007/978-3-662-53008-5_8.

[45] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem, 1997. Avalable at `https://arxiv.org/abs/quant-ph/9705002`.

[46] Joseph Jaeger, Fang Song, and Stefano Tessaro. Quantum key-length extension. In *19th Theory of Cryptography Conference—TCC 2021, Part I*, volume 13042 of *LNCS*, pages 209–239. Springer, 2021. ISBN 978-3-030-90459-3.

[47] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In *14th Theory of Cryptography Conference—TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, 2016. doi: 10.1007/978-3-662-53644-5_8.

[48] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *15th Theory of Cryptography Conference—TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, 2017. doi: 10.1007/978-3-319-70500-2_12.

[49] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *17th Theory of Cryptography Conference—TCC 2019, Part II*, volume 11892 of *LNCS*, pages 61–90. Springer, 2019. doi: 10.1007/978-3-030-36033-7_3.

[50] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *Adv. in Cryptology—Eurocrypt 2020, Part III*, volume 12107 of *LNCS*, pages 703–728. Springer, 2020. doi: 10.1007/978-3-030-45727-3_24.

[51] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. Cryptology ePrint Archive, Report 2021/280, 2021. `https://eprint.iacr.org/2021/280`.

[52] Dominique Unruh. Post-quantum security of Fiat-Shamir. In *Advances in Cryptology—Asiacrypt 2017, Part I*, volume 10624 of *LNCS*, pages 65–95. Springer, 2017. doi: 10.1007/978-3-319-70694-8_3.

[53] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *Adv. in Cryptology—Eurocrypt 2018, Part III*, volume 10822 of *LNCS*, pages 552–586. Springer, 2018. doi: 10.1007/978-3-319-78372-7_18.

[54] Wim van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. *SIAM J. Computing*, 36(3):763–778, 2006.

[55] Gorjan Alagic and Alexander Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In *Adv. in Cryptology—Eurocrypt 2017, Part III*, volume 10212 of *LNCS*, pages 65–93. Springer, 2017. doi: 10.1007/978-3-319-56617-7_3.

[56] Hector Bjoljahn Hougaard. How to generate pseudorandom permutations over other groups: Even-Mansour and Feistel revisited, 2017. URL `https://arxiv.org/abs/1707.01699`. Available at `https://arxiv.org/abs/1707.01699`.

[57] Xavier Bonnetain and María Naya-Plasencia. Hidden shift quantum cryptanalysis and implications. In *Advances in Cryptology—Asiacrypt 2018, Part I*, volume 11272 of *LNCS*, pages 560–592. Springer, 2018. doi: 10.1007/978-3-030-03326-2_19.

[58] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Sebastian Ramacher, Christian Rechberger, Emmanuela Orsini, Lawrence Roy, et al. Faest: Algorithm specifications. 2023.

[59] Vincent Rijmen and Joan Daemen. Advanced encryption standard. *Proceedings of federal information processing standards publications, national institute of standards and technology*, 19:22, 2001.

[60] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Adv. in Cryptology—Eurocrypt 2006*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006. Full version available at `https://eprint.iacr.org/2004/331`.

[61] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *28th Annual ACM Symp. on Theory of Computing (STOC)*, pages 212–219. ACM Press, 1996. URL `http://doi.acm.org/10.1145/237814.237866`.

[62] Hidenori Kuwakado and Masakatu Morii. Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In *Proc. IEEE International Symposium on Information Theory*, pages 2682–2685. IEEE, 2010.

[63] Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. Beyond quadratic speedups in quantum attacks on symmetric schemes. In *Advances in Cryptology—Eurocrypt 2022, Part III*, volume 13277 of *LNCS*, pages 315–344. Springer, 2022.

[64] Ansis Rosmanis. Hybrid quantum-classical search algorithms. *arXiv preprint arXiv:2202.11443*, 2022.

[65] Yassine Hamoudi, Qipeng Liu, and Makrand Sinha. Quantum-classical tradeoffs in the random oracle model. *arXiv preprint arXiv:2211.12954*, 2022.

[66] Alexandru Cojocaru, Juan Garay, and Fang Song. Generalized hybrid search and applications. *Cryptology ePrint Archive*, 2023.

[67] Jelle Don, Serge Fehr, and Yu-Hsuan Huang. Adaptive versus static multi-oracle algorithms, and quantum security of a split-key PRF. In *20th Theory of Cryptography Conference—TCC 2022, Part I*, volume 13747 of *LNCS*, pages 33–51. Springer, 2022. doi: 10.1007/978-3-031-22318-1_2. URL `https://doi.org/10.1007/978-3-031-22318-1_2`.

[68] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Çağdaş Çalık, Lawrence Bassham, Jinkeon Kang, and John Kelsey. Status report on the second round of the NIST lightweight cryptography standardization process, 2021. NIST IR 8369.

[69] 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.

[70] Ashwin Nayak. Inverting a permutation is as hard as unordered search. *arXiv preprint arXiv:1007.2899*, 2010.

[71] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Cryptographic sponge functions. Submission to NIST (Round 3), 2011. URL `http://sponge.noekeon.org/CSF-0.1.pdf`.

[72] Ashwin Nayak. Inverting a permutation is as hard as unordered search. *Theory of Computing*, 7(1):19–25, 2011. ISSN 1557-2862. doi: 10.4086/toc.2011.v007a002. URL `http://dx.doi.org/10.4086/toc.2011.v007a002`.

[73] Gorjan Alagic, Chen Bai, Alexander Poremba, and Kaiyan Shi. On the two-sided permutation inversion problem. Cryptology ePrint Archive, Paper 2023/985, 2023. URL `https://eprint.iacr.org/2023/985`. `https://eprint.iacr.org/2023/985`.

[74] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Advances in Cryptology – CRYPTO 2013*, pages 361–379. Springer, 2013. doi: 10.1007/978-3-642-40084-1_21.

[75] Seny Kamara and Jonathan Katz. How to encrypt with a malicious random number generator. In *International Workshop on Fast Software Encryption*, pages 303–315. Springer, 2008.

[76] Joe Kilian and Phil Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *Journal of Cryptology*, 14(1):17–35, 2001.

[77] Martin R Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçın. Block ciphers–focus on the linear layer (feat. pride). In *Advances in Cryptology–CRYPTO 2014: 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I 34*, pages 57–76. Springer, 2014.

[78] 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 *Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings 18*, pages 208–225. Springer, 2012.

[79] Jan Czajkowski, Leon Groot Bruinderink, Andreas Hülsing, Christian Schaffner, and Dominique Unruh. Post-quantum security of the sponge construction. In *International Conference on Post-Quantum Cryptography*, pages 185–204. Springer, 2018.

[80] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability, 2021.

[81] Morris Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015-08-04 2015.