# Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World*

Dan Boneh Mark Zhandry

Stanford University
{dabo,zhandry}@cs.stanford.edu

**Abstract**

We initiate the study of *quantum*-secure digital signatures and *quantum* chosen ciphertext security. In the case of signatures, we enhance the standard chosen message query model by allowing the adversary to issue *quantum* chosen message queries: given a superposition of messages, the adversary receives a superposition of signatures on those messages. Similarly, for encryption, we allow the adversary to issue *quantum* chosen ciphertext queries: given a superposition of ciphertexts, the adversary receives a superposition of their decryptions. These adversaries model a natural ubiquitous quantum computing environment where end-users sign messages and decrypt ciphertexts on a personal quantum computer.

We construct classical systems that remain secure when exposed to such quantum queries. For signatures, we construct two compilers that convert classically secure signatures into signatures secure in the quantum setting and apply these compilers to existing post-quantum signatures. We also show that standard constructions such as Lamport one-time signatures and Merkle signatures remain secure under quantum chosen message attacks, thus giving signatures whose quantum security is based on generic assumptions. For encryption, we define security under quantum chosen ciphertext attacks and present both public-key and symmetric-key constructions.

*Keywords*: Quantum computing, signatures, encryption, quantum security

## 1 Introduction

Recent progress in building quantum computers [IBM12] gives hope for their eventual feasibility. Consequently, there is a growing need for quantum-secure cryptosystems, namely classical systems that remain secure against quantum computers. Post-quantum cryptography generally studies the settings where the adversary is armed with a quantum computer, but users only have classical machines. In this paper, we go a step further and study the eventuality where end-user machines are quantum. In these settings, an attacker may interact with honest parties using quantum queries, as discussed below, potentially giving the attacker more power. The challenge is to construct cryptosystems that remain secure when exposed to such quantum queries. We emphasize that all the systems we consider are classical and can be easily implemented on a classical computer. Our

*This article is based on an earlier article: Dan Boneh and Mark Zhandry. Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World. In *Proceedings of CRYPTO, 2013*. 

goal is to construct classical systems that remain secure even when implemented on a quantum computer, thereby potentially giving the attacker the ability to issue quantum queries.

Along these lines, Zhandry [Zha12b] showed how to construct pseudorandom functions (PRFs) that remain secure even when the adversary is allowed to issue *quantum* queries to the PRF. A quantum query is a superposition of inputs $\sum_x \psi_x \ |x\rangle$ of the attacker's choice. The response is a superposition $\sum_x \psi_x \ |x, F(k,x)\rangle$ where $F(k,x)$ is the value of the PRF at a point $x$ under key $k$. Zhandry showed that certain PRFs are secure even under such a powerful query model. More recently, Boneh and Zhandry [BZ13] showed how to construct message authentication codes (MACs) that remain secure even when the attacker is allowed to issue *quantum* chosen message queries. That is, for a superposition of messages $\sum_m \psi_m \ |m\rangle$ of the attacker's choice, the attacker is given $\sum_m \psi_m \ |m, S(k,m)\rangle$ where $S(k,m)$ is the tag on message $m$ using key $k$. They showed that some classically secure MACs become insecure under quantum chosen message queries and they constructed several quantum-secure MAC families.

**Our contributions.** In this paper, we construct the first quantum-secure signatures and quantum-secure chosen ciphertext encryption systems.

We begin by defining security for digital signatures under a *quantum* chosen message attack. A quantum chosen message query [BZ13] gives the attacker the signatures on all messages in a quantum superposition. In more detail, a quantum chosen message query is the transformation

$$\sum_m \psi_m \ |m\rangle \qquad \longrightarrow \qquad \sum_m \psi_m \ |m, \ S(\mathsf{sk}, m)\rangle$$

where $S(\mathsf{sk}, x)$ is the signature on $x$ using signing key $\mathsf{sk}$. The attacker can sample the response to such a query and obtain one valid message-signature pair. After $q$ such queries, it can obtain $q$ valid message-signature pairs. We say that a signature scheme is existentially unforgeable under a *quantum* chosen message attack if, after $q$ quantum chosen message queries, the attacker cannot produce $q+1$ valid message-signature pairs.

Next, we present several compilers that convert a signature scheme that is secure under *classical* queries into one secure under *quantum* queries. In particular, we give the following constructions:

- Using a chameleon hash [KR00], we show how to transform any signature that is existentially unforgeable under a *classical random* message into a signature scheme that is existentially unforgeable under a *quantum chosen* message attack. We apply this conversion to several existing signature schemes, giving constructions whose quantum security is based on the quantum hardness of lattice problems.

- We show that any *universally* unforgeable signature under a *classical random* message attack can be made *existentially* unforgeable under a *quantum chosen* message attack in the random oracle model. For example, this conversion applies to a randomized variant of GPV signatures [GPV08], proving security of the scheme even under a *quantum* chosen message attack. We also separately show that the basic deterministic GPV scheme is secure in this setting.

- Finally, we prove that classical constructions such as Lamport one-time signatures and Merkle signatures are existentially unforgeable under a *quantum* chosen message attack. These results show how to build quantum-secure signatures from any collision resistant hash function. We leave open the problem of basing security on one-way functions. We also note that the

version of Lamport signatures that we prove secure is non-optimized, and can potentially be made more efficient using standard combinatorial techniques. Unfortunately, we cannot prove quantum-security of an optimized Lamport signature and leave that as an interesting open problem.

Turning to encryption, we first explain how to adapt the chosen ciphertext security game to the quantum setting. In the classical game, the attacker is given classical access to a decryption oracle used to answer chosen ciphertext queries and to an encryption oracle used to create challenge ciphertexts. In the quantum setting, the decryption oracle accepts a superposition of ciphertexts and returns a superposition of their decryptions:

$$\sum_m \psi_c \, |c\rangle \qquad \longrightarrow \qquad \sum_c \psi_c \, |c, \; D(\mathsf{sk}, c)\rangle \; .$$

One might also try to allow quantum access to the encryption oracle; however, we show that the resulting concept is unsatisfiable. We therefore restrict the encryption oracle to be classical.

Armed with this definition of security, we construct quantum-secure chosen ciphertext systems in both the public-key and symmetric-key settings:

- Our symmetric-key construction is built from any secure PRF, and follows the encrypt-then-MAC paradigm. The classical proof that encrypt-then-MAC is secure for generic encryption and generic MAC schemes does not carry over to the quantum setting, but we are able to prove security for our specific construction.

- We show that public-key quantum chosen ciphertext security can be obtained from any identity-based encryption scheme that is selectively secure under a quantum chosen identity attack. Such an identity-based encryption scheme can, in turn, be built from lattice assumptions. This construction is the quantum analogue of the CHK transformation from identity-based encryption to public-key chosen ciphertext security [BCHK04].

**Motivation.** Allowing the adversary to issue quantum queries is a natural and conservative security model and is therefore an interesting one to study. Constructing signature and encryption schemes that remain secure in these models gives confidence in the event that end-user computing devices eventually become quantum. Nevertheless, one might imagine that in a future where all computers are quantum, the last step in a signature or decryption procedure is to sample the final quantum state. This ensures that the results are always classical, thereby preventing quantum superposition attacks. Security in this case relies on a physical hardware assumption, namely that the final "classicalization" step is implemented correctly and cannot be circumvented by a quantum adversary. In contrast, using systems that are inherently secure against superposition attacks frees the hardware designer from worrying about the security of the classicalization step.

As further motivation, we note that our results are the tip of a large emerging area with many open questions. For any cryptographic primitive modeled as an interactive game, one can ask how to design primitives that remain secure when the interaction between the adversary and its given oracles is quantum. For example, can we design quantum-secure threshold signatures and group signatures? Can we construct a quantum-secure PRF for a large domain from a quantum-secure PRF for a small domain? In particular, do the CBC-MAC or NMAC constructions give quantum-secure PRFs?

**Other related work.** Several recent works study the security of cryptographic primitives when the adversary can issue quantum queries. Boneh et al. [BDF+11] and Zhandry [Zha12a] prove the classical security of signatures, encryption, and identity-based encryption schemes in the *quantum* random oracle model, where the adversary can query the random oracle on superpositions of inputs. In these papers, the interaction with the challenger is classical. These results show that many, but not all, random oracle constructions remain secure in the quantum random oracle model. The quantum random oracle model has also been used to prove security of Merkle's Puzzles in the quantum setting [BS08, BHK+11]. Damgård et al. [DFNS11] examine secret sharing and multiparty computation in a model where an adversary may corrupt a superposition of subsets of players, and build zero knowledge protocols that are secure, even when a dishonest verifier can issue challenges on superpositions.

Some progress toward identifying sufficient conditions under which classical protocols are also quantum immune has been made by Unruh [Unr10] and Hallgren et al. [HSS11]. Unruh shows that any scheme that is statistically secure in Cannetti's universal composability (UC) framework [Can01] against classical adversaries is also statistically secure against quantum adversaries. Hallgren et al. show that for many schemes, this is also true in the computational setting. These results, however, do not apply to cryptographic primitives such as signatures and encryption and do not consider quantum superposition attacks.

## 2 Preliminaries: Background and Techniques

We will let $[n]$ denote the set $\{1, ..., n\}$. Functions will be denoted by capital letters (such as $F$), and sets by capital script letters (such as $\mathcal{X}$). We will let $x \xleftarrow{R} D$ for some distribution $D$ denote drawing $x$ according to $D$, and $x \xleftarrow{R} \mathcal{X}$ for some set $\mathcal{X}$ denote drawing a random element from $\mathcal{X}$. Given a function $F : \mathcal{X} \to \mathcal{Y}$ and a subset $\mathcal{S} \subseteq \mathcal{X}$, the restriction of $F$ to $\mathcal{S}$ is the function $F_{\mathcal{S}} : \mathcal{S} \to \mathcal{Y}$ where $F_{\mathcal{S}}(x) = F(x)$ for all $x \in \mathcal{S}$. A distribution $D$ on $F$ induces a distribution $D_{\mathcal{S}}$ on $F_{\mathcal{S}}$. We say that $D$ is $k$-wise independent if each of the distributions $D_{\mathcal{S}}$ are truly random distributions on functions from $\mathcal{S}$ to $\mathcal{Y}$, for all sets $\mathcal{S}$ of size at most $k$. A set $\mathcal{F}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is $k$-wise independent if the uniform distribution on $\mathcal{F}$ is $k$-wise independent. A non-negative function $f(n)$ is negligible if, for any $c$, $f(n) < 1/n^c$ for all sufficiently large $n$. If a function $g(n)$ can be written as $h(n) \pm f(n)$ where $f(n)$ is negligible, we write $g(n) = h(n) \pm \mathsf{negl}$.

### 2.1 Quantum Computation

We give a short introduction to quantum computation. A quantum system $A$ is a complex Hilbert space $\mathcal{H}$ together with and inner product $\langle \cdot | \cdot \rangle$. The state of a quantum system is given by a vector $|\psi\rangle$ of unit norm ($\langle \psi | \psi \rangle = 1$). Given quantum systems $\mathcal{H}_1$ and $\mathcal{H}_2$, the joint quantum system is given by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$. Given $|\psi_1\rangle \in \mathcal{H}_1$ and $|\psi_2\rangle \in \mathcal{H}_2$, the product state is given by $|\psi_1\rangle|\psi_2\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$. Given a quantum state $|\psi\rangle$ and an orthonormal basis $B = \{|b_0\rangle, ..., |b_{d-1}\rangle\}$ for $\mathcal{H}$, a measurement of $|\psi\rangle$ in the basis $B$ results in the value $i$ with probability $|\langle b_i | \psi \rangle|^2$, and the quantum state collapses to the basis vector $|b_i\rangle$. If $|\psi\rangle$ is actually a state in a joint system $\mathcal{H} \otimes \mathcal{H}'$, then $|\psi\rangle$ can be written as

$$|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i \, |b_i\rangle |\psi_i'\rangle$$

for some complex values $\alpha_i$ and states $|\psi_i'\rangle$ over $\mathcal{H}'$. Then, the measurement over $\mathcal{H}$ obtains the value $i$ with probability $|\alpha_i|^2$ and in this case the resulting quantum state is $|b_i\rangle|\psi_i'\rangle$.

A unitary transformation over a $d$-dimensional Hilbert space $\mathcal{H}$ is a $d \times d$ matrix $\mathbf{U}$ such that $\mathbf{U}\mathbf{U}^\dagger = \mathbf{I}_d$, where $\mathbf{U}^\dagger$ represents the conjugate transpose. A quantum algorithm operates on a product space $\mathcal{H}_{in} \otimes \mathcal{H}_{out} \otimes \mathcal{H}_{work}$ and consists of $n$ unitary transformations $\mathbf{U}_1, ..., \mathbf{U}_n$ in this space. $\mathcal{H}_{in}$ represents the input to the algorithm, $\mathcal{H}_{out}$ the output, and $\mathcal{H}_{work}$ the work space. A classical input $x$ to the quantum algorithm is converted to the quantum state $|x, 0, 0\rangle$. Then, the unitary transformations are applied one-by-one, resulting in the final state

$$|\psi_x\rangle = \mathbf{U}_n ... \mathbf{U}_1 |x, 0, 0\rangle \ .$$

The final state is then measured, obtaining the tuple $(a, b, c)$ with probability $|\langle a, b, c|\psi_x\rangle|^2$. The output of the algorithm is $b$. We say that a quantum algorithm is efficient if each of the unitary matrices $\mathbf{U}_i$ come from some fixed basis set, and $n$, the number of unitary matrices, is polynomial in the size of the input.

**Quantum-accessible Oracles.** We will implement an oracle $O : \mathcal{X} \to \mathcal{Y}$ by a unitary transformation $\mathbf{O}$ where

$$\mathbf{O}|x, y, z\rangle = |x, y + O(x), z\rangle$$

where $+ : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ is some group operation on $\mathcal{X}$. Suppose we have a quantum algorithm that makes quantum queries to oracles $O_1, ..., O_q$. Let $|\psi_0\rangle$ be the input state of the algorithm, and let $\mathbf{U}_0, ..., \mathbf{U}_q$ be the unitary transformations applied between queries. Note that the transformations $\mathbf{U}_i$ are themselves possibly the products of many simpler unitary transformations. The final state of the algorithm will be

$$\mathbf{U}_q \mathbf{O}_q ... \mathbf{U}_1 \mathbf{O}_1 \mathbf{U}_0 |\psi_0\rangle$$

We can also have an algorithm make classical queries to $O_i$. In this case, the input to the oracle is measured before applying the transformation $\mathbf{O}_i$. We call a quantum oracle algorithm efficient if the number of queries $q$ is a polyomial, and each of the transformations $\mathbf{U}_i$ between queries can be written as the product polynomially many unitary transformations from some fixed basis set.

**Tools.** Next we state several lemmas and definitions that we will use throughout the paper. Some have been proved in other works, and the rest are proved in Appendix B. The first concerns partial measurements, and will be used extensively throughout the paper:

**Lemma 2.1.** *Let $A$ be a quantum algorithm, and let $\Pr[x]$ be the probability that $A$ outputs $x$. Let $A'$ be another quantum algorithm obtained from $A$ by pausing $A$ at an arbitrary stage of execution, performing a partial measurement that obtains one of $k$ outcomes, and then resuming $A$. Let $\Pr'[x]$ be the probability $A'$ outputs $x$. Then $\Pr'[x] \geq \Pr[x]/k$.*

This lemma means, for example, that if you measure just one qubit, the probability of a particular output drops by at most a factor of two. We also make use of the following lemma, proved by Zhandry [Zha12a], which allows us to simulate random oracle efficiently using $k$-wise independent functions:

**Lemma 2.2** ([Zha12a])**.** *Let $H$ be an oracle drawn from a $2q$-wise independent distribution. Then the advantage any quantum algorithm making at most $q$ queries to $H$ has in distinguishing $H$ from a truly random function is identically 0.*

The next definition and lemma are given by Zhandry [Zha12b] and allow for the efficient simulation of an exponentially-large list of samples, given only a polynomial number of samples:

**Definition 2.3** (Small-range distributions [Zha12b])**.** *Fix sets $\mathcal{X}$ and $\mathcal{Y}$ and a distribution $D$ on $\mathcal{Y}$. Fix an integer $r$. Let $\mathbf{y} = (y_1, ..., y_r)$ be a list of $r$ samples from $D$ and let $P$ be a random function from $\mathcal{X}$ to $[r]$. The distributions on $\mathbf{y}$ and $P$ induce a distribution on functions $H : \mathcal{X} \to \mathcal{Y}$ defined by $H(x) = y_{P(x)}$. This distribution is called a* small-range distribution *with $r$ samples of $D$.*

**Lemma 2.4** ([Zha12b])**.** *There is a universal constant $C_0$ such that, for any sets $\mathcal{X}$ and $\mathcal{Y}$, distribution $D$ on $\mathcal{Y}$, any integer $\ell$, and any quantum algorithm $A$ making $q$ queries to an oracle $H : \mathcal{X} \to \mathcal{Y}$, the following two cases are indistinguishable, except with probability less than $C_0 q^3/\ell$:*

- *$H(x) = y_x$ where $\mathbf{y}$ is a list of samples of $D$ of size $|\mathcal{X}|$.*
- *$H$ is drawn from the small-range distribution with $\ell$ samples of $D$.*

We use Lemma 2.4 to prove the following corollary:

**Lemma 2.5.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and for each $x \in \mathcal{X}$, let $D_x$ and $D'_x$ be distributions on $\mathcal{Y}$ such that $|D_x - D'_x| \leq \epsilon$ for some value $\epsilon$ that is independent of $x$. Let $O : \mathcal{X} \to \mathcal{Y}$ be a function where, for each $x$, $O(x)$ is drawn from $D_x$, and let $O'(x)$ be a function where, for each $x$, $O'(x)$ is drawn form $D'_x$. Then any quantum algorithm making at most $q$ queries to either $O$ or $O'$ cannot distinguish the two, except with probability at most $\sqrt{8C_0 q^3 \epsilon}$.*

Zhandry [Zha12b] proves this corollary for the special case where all of the $D_x$ distributions are the same and all of the $D'_x$ distributions are the same. Lastly, we need the following lemma:

**Lemma 2.6.** *Fix sets $\mathcal{X}$ and $\mathcal{Y}$, and distributions $D_x$ on $\mathcal{Y}$ for each $x \in \mathcal{X}$. Let $H$ be a function from $\mathcal{X}$ to $\mathcal{Y}$ where, for each $x$, $H(x)$ is drawn independently according to $D_x$. Then any quantum algorithm making $q$ quantum queries to $H$ can only produce $q+1$ input/output pairs of $H$ with probability at most $(q+1)/\left\lfloor 2^{H_\infty} \right\rfloor$, where $H_\infty$ be the minimum over all $x \in \mathcal{X}$ of the min-entropy of the distribution $D_x$.*

A special case of this theorem is when $F$ is a constant function and each of the distirbutions $D_x$ are the uniform distribution. In this case, Lemma 2.6 reduces to the following result of Boneh and Zhandry [BZ13]: any quantum algorithm making $q$ queries to a random oracle $H$ from $\mathcal{X}$ to $\mathcal{Y}$ can output $q+1$ input/output pairs of $H$ with probability at most $(q+1)/|\mathcal{Y}|$. We prove Lemma 2.6 by reducing the general case to this special case with $|\mathcal{Y}| = \left\lfloor 2^{H_\infty} \right\rfloor$.

# 3 Quantum-Secure Signatures

Our goal is to construct signatures that are resistant to a *quantum* chosen message attack, where the adversary submits quantum superpositions of messages and receives the corresponding superpositions of signatures in return. First, we need a suitable definition of what a signature scheme is in our setting, and what it means for such a scheme to be secure. Correctness for a stateless signature scheme is identical to the classical setting: any signature produced by the signing algorithm must verify. There is some subtlety, however, for stateful signature schemes. If the state of the signing algorithm depends on the messages signed, and if the adversary mounts a quantum chosen message

attack, the signing algorithm and adversary will become entangled. To keep the state of the signing algorithm classical and unentangled with the adversary, we therefore restrict the state to be independent of the messages signed so far. We note that many stateful signature schemes, such as stateful Merkle signatures, satisfy this requirement. We arrive at the following definition:

**Definition 3.1.** *A signature scheme $\mathcal{S}$ is a tuple of efficient classical algorithms* $(\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ *where*

- $\mathsf{G}(\lambda)$ *generates a private/public key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $\mathsf{Sign}(\mathsf{sk}, m, \mathsf{state})$ *outputs a signature* $\sigma$ *and new state* $\mathsf{state}'$. *If the output* $\mathsf{state}$ *is ever non-empty, we say that algorithm* $\mathsf{Sign}$ *is stateful and we require that the state does not depend in any way on the messages that have been signed so far. If the output* $\mathsf{state}$ *is always empty, we say that* $\mathsf{Sign}$ *is stateless and we drop the* $\mathsf{state}$ *variables altogether.*
- $\mathsf{Ver}(\mathsf{pk}, m, \sigma)$ *either accepts or rejects. We require that valid signatures are always accepted, that is if* $\sigma$ *is the output of* $\mathsf{Sign}(\mathsf{sk}, m, \mathsf{state})$ *then* $\mathsf{Ver}(\mathsf{pk}, m, \sigma)$ *accepts.*

For security, we use a notion similar to that for message authentication codes defined by Boneh and Zhandry [BZ13]. There are two issues in defining security under a quantum chosen message attack:

- **Randomness.** When using a randomized signature scheme, there are several choices for how the randomness is used. One option is to choose a single randomness value for each chosen message query, and sign every message in the superposition with that randomness. Another approach is to choose fresh randomness for each message in the superposition. The drawback of the second approach is that whomever is implementing the scheme on a quantum device needs to guarantee that every message in the superposition is signed with fresh independent randomness.

  The first approach, where the same randomness is used to sign all messages in a superposition, is much simpler for implementers and we therefore design signature schemes secure in this setting. Fortunately, there is a simple transformation that converts a scheme requiring independent randomness for every message into a scheme that is secure when a single randomness value is used for an entire query: when signing, choose a fresh random key $k$ for a quantum pseudorandom function (QPRF). This will be the single per-query randomness value. To sign a superposition of messages, sign each message $m$ in the superposition using randomness obtained by applying the QPRF to $m$ using the key $k$. From the adversary's point of view, this is indistinguishable from choosing independent randomness for each message. Using Lemma 2.2, we can replace the QPRF with a function drawn from a pairwise independent function family, which is far more efficient than using a QPRF. Hence, requiring global randomness per query does not complicate the signature scheme much, but greatly simplifies its implementation.

- **Forgeries.** Each quantum chosen message query can be a superposition of every message in the message space. Sampling the returned superposition will result in a single message/signature pair for a random message. Therefore, the classical notion of existential forgery being a signature on a *new* message is ill-defined when we allow quantum access. Instead, for security we require that the adversary cannot produce $q + 1$ valid message/signature pairs with $q$

quantum chosen message queries. Security definitions in this style were previously used in the context of blind signatures [PS96].

We arrive at the following definition of security:

**Definition 3.2** (Quantum Security)**.** *A signature scheme* $\mathcal{S} = (\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ *is strongly existentially unforgeable under a quantum chosen-message attack (EUF-qCMA secure) if, for any efficient quantum algorithm* $A$ *and any polynomial* $q$*,* $A$*'s probability of success in the following game is negligible in* $\lambda$*:*

**Key Gen** *The challenger runs* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{G}(\lambda)$*, and gives* $\mathsf{pk}$ *to* $A$*.*

**Signing Queries** *The adversary makes a polynomial* $q$ *chosen message queries. For each query, the challenger chooses randomness* $r$*, and responds by signing each message in the query using* $r$ *as randomness:*

$$\sum_{m,t} \psi_{m,t} |m,\ t\rangle \qquad \longrightarrow \qquad \sum_{m,t} \psi_{m,t} |m,\ t \oplus \mathsf{Sign}(\mathsf{sk}, m; r)\rangle$$

**Forgeries** *The adversary is required to produce* $q+1$ *message/signature pairs. The challenger then checks that all the signatures are valid, and that all message/signature pairs are distinct. If so, the challenger reports that the adversary wins.* □

In this paper, we will also be using several weaker notions of security. The first is for a classical chosen message attack:

**Definition 3.3.** *$\mathcal{S}$ is existentially unforgeable under a* classical *random message attack (EUF-CMA secure) if every signing query is measured before signing, so that only a single classical message is signed per query.*

Next, we define random message security:

**Definition 3.4.** *$\mathcal{S}$ is existentially unforgeable under a random message attack (EUF-RMA secure) if the adversary is not allowed any signing queries, but instead receives $q$ message/signature pairs for uniform random messages at the beginning of the game.*

We can weaken the security definition even further, to get universal unforgeability:

**Definition 3.5.** *$\mathcal{S}$ is universally unforgeable under a random message attack (UUF-RMA secure) if, along with receiving $q$ message/signature pairs for random messages, the adversary receives $n$ additional random messages, and all of the $q+1$ messages for which a signature is forged must be among the $q+n$ messages received.*

All of the above security definitions also have weak variants, where in addition to requiring that message/signature forgery pairs be distinct, we also require that the messages themselves be distinct. Finally, all of the above security definitions also have $k$-time variants for any constant $k$, where the value of $q$ is bounded to at most $k$. When the distinction is required, we refer to the standard unbounded $q$ notion as many-time security.

### 3.1 A Separation Example

Next we show that quantum chosen message queries give the adversary more power than classical chosen message queries. In particular, we present a signature scheme that is secure under classical queries, but completely insecure once an adversary can make quantum queries. Let $\mathcal{S}_c = (\mathsf{G}_c, \mathsf{Sign}_c, \mathsf{Ver}_c)$ be a signature scheme that is secure under classical chosen message queries. We augment $\mathcal{S}_c$ by choosing a random secret prime $p$ and storing $p$ in the secret signing key. We modify the signature scheme so that the signature on the message $m = p$ includes the entire secret key. As long as the adversary does not learn $p$, she should not be able to learn the secret key. We also add some auxiliary information to the signatures such that, under classical queries, $p$ is hidden, but a single quantum query suffices to recover $p$. Our signature scheme is as follows:

**Construction 3.6.** *Fix positive integers $N, N'$. Let $\mathcal{M}$ be the interval $[0, N)$. Let $\mathcal{S}_c = (\mathsf{G}_c, \mathsf{Sign}_c, \mathsf{Ver}_c)$ be a signature scheme that signs messages in $\mathcal{M}$ and $\mathsf{PRF}$ be a pseudorandom function with domain $\mathcal{M}$. Let $\mathsf{RPrime}(N')$ denote a procedure that samples a random prime in the interval $[N'/2, N')$. We build a new signature scheme $\mathcal{S} = (\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ as follows:*

$$\begin{aligned}
\mathsf{G}(\lambda) &: (\mathsf{sk}_c, \mathsf{pk}_c) \xleftarrow{R} \mathsf{G}_c(\lambda),\ k \xleftarrow{R} \{0,1\}^\lambda,\ p \xleftarrow{R} \mathsf{RPrime}(N') \\
&\quad \textit{output}\ \mathsf{sk} = (\mathsf{sk}_c, k, p),\ \mathsf{pk} = \mathsf{pk}_c \\
\mathsf{Sign}((\mathsf{sk}_c, k, p), m) &: s_1 \leftarrow \mathsf{PRF}(k,\ m \bmod p) \\
&\quad s_2 \leftarrow \begin{cases} \mathsf{sk} & \text{if } m = p \\ 0 & \text{if } m \neq p \end{cases} \\
&\quad \sigma \leftarrow \mathsf{Sign}_c(\mathsf{sk}_c, (m, s_1, s_2)) \\
&\quad \textit{output}\ (\sigma, s_1, s_2) \\
\mathsf{Ver}(pk, m, (\sigma, s_1, s_2)) &: \textit{output}\ \mathsf{Ver}(\mathsf{pk}, (m, s_1, s_2), \sigma)
\end{aligned}$$

**Theorem 3.7.** *If $\mathcal{S}_c$ is existentially unforgeable under a* classical *chosen message attack, and* $\mathsf{PRF}$ *is secure against classical queries, then for an appropriate choice of $N, N'$, $\mathcal{S}$ is also existentially unforgeable under a classical chosen message attack, but is totally broken under a quantum chosen message attack.*

**Proof**. Let $N'$ be an integer that grows exponentially in the security parameter, and let $N$ be the smallest power of 2 greater than $4N'^2$. The proof is in two parts: first we argue the classical security of our scheme, and then we break the scheme using quantum queries. Suppose we have an adversary that breaks the scheme with probability $\epsilon$. We will prove classical security though a sequence of games:

**Game 0.** This is the standard attack game, where the adversary can ask for a polynomial number of signatures on messages of his choice, and must produce a signature on a new message. By assumption, the adversary succeeds with probability $\epsilon$.

**Game 1.** Instead of using $\mathsf{PRF}$ to compute $s_1$, choose a random function $H$ at the start of the game, and let $s_1 = H(m \mod p)$. The classical security of $\mathsf{PRF}$ implies that the adversary's success probability is still negligibly close to $\epsilon$.

Define Bad as the event that the adversary either queries on $p$, or on two messages differing by a (non-zero) multiple of $p$. We now analyze the probability that Bad occurs. Suppose that, after the $i$th query, Bad has not occurred. Then all of the $s_1$ values are drawn independently at random for different $m$, and all the $s_2$ values are 0. The adversary learns nothing about $p$ other than the fact that $p$ is not equal to one of his queries, and does not divide the differences between any of his queries. Since each message is at most $O(N'^2)$, but $p$ is at least $\Omega(N')$, each difference is divisible between at most 2 different $p$. Since there are $\binom{i}{2}$ differences after $i$ queries, the adversary has ruled out at most $2\binom{i}{2} + i \in O(i^2)$ different values for $p$. There are $\Omega(N'/\log N')$ primes in the interval $[N'/2, N')$, so the fraction of primes ruled out is negligible. By similar logic, when the adversary makes query $i+1$, she can attempt to rule out at most $2i + 1 \in O(i)$ additional values of $p$. Bad occurs for query $i+1$ exactly when the actual $p$ is one of these values. Therefore, the probability that Bad occurs at query $i+1$, given that it hasn't occurred yet, is at most $O(\ (i \log N')/N')$. Therefore, the probably that Bad occurs in any query is then $O(\ (q^2 \log N')/N')$, which is negligible.

**Game 2.** Now choose a random oracle $H'$, and let $s_1 = H'(m)$ and $s_2 = 0$. As long as Bad does not occur in Game 1, Games 1 and 2 are identical. Since Bad only occurs with negligible probability, the adversary still succeeds in Game 2 with probability negligibly close to $\epsilon$.

Now a signature on $m$ is just a triple $(\mathsf{Sign}_c(\mathsf{sk}_c, (m, O(m), 0)),\ O(m), 0)$. It is straightforward to show that any adversary breaking the security of this signature scheme can be modified to break the security of $\mathcal{S}_c$. Therefore, $\epsilon$ must be negligible, showing that $\mathcal{S}$ is secure.

Now we explain how quantum queries can be used to recover sk. We can turn our quantum signing oracle outputting $(\sigma, s_1, s_2)$ into an oracle that outputs only $s_1$ by using standard tricks for quantum oracles. Then, our choices for $N$ and $N'$ allow us to use the period finding algorithm of Boneh and Lipton [BL95] to recover $p$ with only a single quantum query. Once we have $p$, we can easily recover sk by making a single classical query on the message $p$. Therefore, $\mathcal{S}$ is completely insecure under quantum chosen message queries. □

Since classically, signatures and pseudorandom functions can be built from one-way functions, we immediately get the following corollary:

**Corollary 3.8.** *Assuming the existence of one-way functions, there exists a signature scheme $\mathcal{S}$ that is existentially unforgeable under a classical chosen message attack, but is totally broken under a quantum chosen message attack.*

### 3.2 Quantum-Secure Signatures from Classically-Secure Signatures

Now we move to actually building signature schemes that are secure against quantum chosen message attacks. In this section, we show a general transformation from classically secure signatures to quantum secure signatures. The building blocks for our construction are chameleon hash functions and signatures that are secure against a classical random message attack. First, we will define a chameleon hash function. The definition we use is slightly different from the original definition from Krawczyk and Rabin [KR00], but is satisfied by the known lattice constructions:

**Definition 3.9.** *A chameleon hash function $\mathcal{H}$ is a tuple of efficient algorithms* $(\mathsf{G}, \mathsf{H}, \mathsf{Inv}, \mathsf{Sample})$ *where:*

- $\mathsf{G}(\lambda)$ *generates a secret/public key pair* $(\mathsf{sk}, \mathsf{pk})$.

- $\mathsf{H}(\mathsf{pk}, m, r)$ *maps messages to some space* $\mathcal{Y}$
- $\mathsf{Sample}(\lambda)$ *samples* $r$ *from some distribution such that, for every* $\mathsf{pk}$ *and* $m$, $\mathsf{H}(\mathsf{pk}, m, r)$ *is uniformly distributed.*
- $\mathsf{Inv}(\mathsf{sk}, h, m)$ *produces an* $r$ *such that* $\mathsf{H}(\mathsf{pk}, m, r) = h$, *and* $r$ *is distributed negligibly-close to* $\mathsf{Sample}(\lambda)$ *conditioned on* $\mathsf{H}(\mathsf{pk}, m, r) = h$

We say that a chameleon hash function is collision resistant if no efficient quantum algorithm, given only $\mathsf{pk}$, can find collisions in $\mathsf{H}(\mathsf{pk}, \cdot, \cdot)$. Cash et al. [CHKP10] build a simple lattice-based chameleon hash function, and prove that it is collision resistant, provided that the *Shortest Integer Solution* problem ($\mathsf{SIS}$) is hard for an appropriate choice of parameters. The idea behind our construction is to first hash the message with the chameleon hash function and then sign the hash. In order to be secure against quantum queries, care has to be taken in how the randomness for the hash and the signature scheme is generated. In what follows, for any randomized algorithm $A$, we let $A(x; r)$ denote running $A$ on input $x$ with randomness $r$.

**Construction 3.10.** *Let* $\mathcal{H} = (\mathsf{G}_H, \mathsf{H}, \mathsf{Inv}, \mathsf{Sample})$ *be a chameleon hash function, and* $\mathcal{S}_c = (\mathsf{G}_c, \mathsf{Sign}_c, \mathsf{Ver}_c)$ *a signature scheme. Let* $\mathcal{Q}$ *and* $\mathcal{R}$ *be families of pairwise independent functions mapping messages to randomness used by* $\mathsf{Inv}$ *and* $\mathsf{Sign}_c$, *respectively. We define a new signature scheme* $\mathcal{S} = (\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ *where:*

$$\begin{aligned}
\mathsf{G}(\lambda) : {} & (\mathsf{sk}_H, \mathsf{pk}_H) \xleftarrow{R} \mathsf{G}_H(\lambda),\ (\mathsf{sk}_c, \mathsf{pk}_c) \xleftarrow{R} \mathsf{G}_c(\lambda) \\
& \textit{output } \mathsf{sk} = (\mathsf{pk}_H, \mathsf{sk}_c),\ \mathsf{pk} = (\mathsf{pk}_H, \mathsf{pk}_c) \\
\mathsf{Sign}((\mathsf{pk}_H, \mathsf{sk}_c), m) : {} & Q \xleftarrow{R} \mathcal{Q}, R \xleftarrow{R} \mathcal{R} \\
& r \leftarrow \mathsf{Sample}(\lambda; R(m)),\ s \leftarrow Q(m),\ h \leftarrow \mathsf{H}(\mathsf{pk}_H, m, r) \\
& \sigma \leftarrow \mathsf{Sign}(\mathsf{pk}_c, h; s),\ \textit{output } (r, \sigma) \\
\mathsf{Ver}((\mathsf{pk}_H, \mathsf{pk}_c), m, (r, \sigma)) : {} & h \leftarrow \mathsf{H}(\mathsf{pk}_H, m, r),\ \textit{output } \mathsf{Ver}(\mathsf{pk}_c, h, \sigma)
\end{aligned}$$

We note that the chameleon secret key is not used in Construction 3.10, though it will be used in the security proof. Classically, this method of hashing with a chameleon hash and then signing converts any non-adaptively secure scheme into an adaptive one. We show that the resulting scheme is actually secure against an adaptive *quantum* chosen message attack.

**Theorem 3.11.** *If* $\mathcal{S}_c$ *is weakly (resp. strongly) EUF-RMA secure and* $\mathcal{H}$ *is a secure chameleon hash function, then* $\mathcal{S}$ *in Construction 3.10 is weakly (resp. strongly) EUF-qCMA secure. Moreover, if* $\mathcal{S}_c$ *is only one-time secure, then* $\mathcal{S}$ *is also one-time secure.*

Theorem 3.11 shows that we can take a classically EUF-RMA secure signature scheme, combine it with a a chameleon hash, and obtain a quantum-secure signature scheme. In particular, the following constructions will be quantum secure, assuming $\mathsf{SIS}$ is hard:

- A slight modification to the signature scheme of Cash et al. [CHKP10], which combines their chameleon hash function with an EUF-RMA secure signature scheme. The only difference in their scheme is that the values $r$ and $s$ are sampled directly, rather than setting them to be the outputs of pairwise independent functions.

- A modification of the signature scheme of Agrawal, Boneh, and Boyen [ABB10], where we hash the message using a chameleon hash before applying the signature.

We now prove Theorem 3.11:

**Proof**. We first sketch the proof idea. Given an $\mathcal{S}_c$ signature $\sigma$ on a random hash $h$, we can construct an $\mathcal{S}$ signature on any given message $m$: use the chameleon secret key $\mathsf{sk}_H$ to compute a randomness $r$ such that $\mathsf{H}(\mathsf{pk}_H, m, r) = h$, and output the signature $(r, \sigma)$. Thus, we can respond to a classical chosen message attack, given only signatures on random messages.

If the adversary issues a *quantum* chosen message query, we need to sign each of the exponentially many messages in the query superposition. Therefore, using the above technique directly would require signing an exponential number of random hashes. Instead, we use small-range distributions and Lemma 2.4 to reduce the number of signed hashes required to a polynomial. The problem is that the number of hashes signed is still a very large polynomial, whereas the number of signatures produced by our adversary is only $q + 1$, so we cannot rely on the pigeon-hole principle to argue that one of the $\mathcal{S}$ forgeries is in fact a $\mathcal{S}_c$ forgery. We can, however, argue that two of the forgeries must, in some sense, correspond to the same query. If we knew which query, we could perform a measurement, observing which of the (polynomially many) random hashes were signed. Lemma 2.1 shows that the adversary's advantage is reduced by only a polynomial factor. For this query, we now only sign a single random hash, but the adversary produces two forgeries. Therefore, one of these forgeries must be a forgery for $\mathcal{S}_c$. Of course, we cannot tell ahead of time which query to measure, so we just pick the query at random, and succeed with probability $1/q$.

We now give the complete proof. There are four variants to the theorem (one-time vs many time, strong vs weak). We will prove the many-time strong security variant, the other proofs being similar. Let $A$ be an adversary breaking the EUF-qCMA security of $\mathcal{S}$ in Construction 3.10 with non-negligible probability $\epsilon$. We prove security through a sequence of games.

**Game 0.** This is the standard attack experiment, where $A$ receives $\mathsf{pk}_c$ and $\mathsf{pk}_H$, and is allowed to make a polynomial number of quantum chosen message queries. For query $i$, the challenger produces pairwise independent functions $R^{(i)}$ and $Q^{(i)}$, and responds to each message in the query superposition as follows:

- Let $r_m^{(i)} = \mathsf{Sample}(\lambda; R^{(i)}(m))$ and $s_m^{(i)} = Q^{(i)}(m)$.
- Compute $h_m^{(i)} = \mathsf{H}(\mathsf{pk}_H, m, r_m^{(i)})$
- Compute $\sigma_m^{(i)} = \mathsf{Sign}_c(\mathsf{sk}_c, h_m^{(i)}; s_m^{(i)})$
- Respond with the signature $(r_m^{(i)}, \sigma_m^{(i)})$.

In the end, $A$ must produce $q+1$ distinct triples $(m_k^*, r_k^*, \sigma_k^*)$ such that $\mathsf{Ver}(\mathsf{pk}_c, \mathsf{H}(\mathsf{pk}_H, m_k^*, r_k^*), \sigma_k^*)$ accepts. By definition, $A$ wins with probability $\epsilon$, which is non-negligible. Therefore, there is some polynomial $p = p(\lambda)$ such that $p(\lambda) > 1/\epsilon(\lambda)$ for infinitely-many $\lambda$.

**Game 1.** We make two modifications: first, we choose $R^{(i)}$ and $Q^{(i)}$ as truly random functions, which amounts to generating $r_m^{(i)} \leftarrow \mathsf{Sample}(\lambda)$ and picking $s_m^{(i)}$ at random for each $i, m$. According to Lemma 2.2, the view of the adversary is unchanged. Second, we modify the conditions in which $A$ wins by requiring that no two $(m_k^*, r_k^*)$ pairs form a collision for $\mathsf{H}$. The security of $\mathcal{H}$ implies that $A$ succeeds in Game 1 with probability at least $\epsilon - \mathsf{negl}$.

**Game 2.** Generate $s_m^{(i)}$ as before, but now draw $h_m^{(i)}$ uniformly at random. Additionally, draw uniform randomness $t_m^{(i)}$. We will sample $r_m^{(i)}$ from the set of randomness making $\mathsf{H}(\mathsf{pk}, m, r_m^{(i)}) = h_m^{(i)}$. That is, let $r_m^{(i)} = \mathsf{Inv}(\mathsf{sk}, h_m^{(i)}, m; t_m^{(i)})$. The only difference from $A$'s perspective is the distribution of the $r_m^{(i)}$ values. For each $m$, the distribution of $r_m^{(i)}$ is negligibly-close to that of Game 1, so the oracles mapping $m$ to $r_m^{(i)}$ are indistinguishable from those in Game 1 by Lemma 2.5. Therefore, the success probability is at least $\epsilon - \mathsf{negl}$.

**Game 3.** Let $\ell = 2C_0qp$ where $C_0$ is the constant from Lemma 2.4. At the beginning of the game, for $i = 1, ..., q$ and $j = 1, ..., \ell$, sample values $\hat{h}_j^{(i)}$ and let $\hat{\sigma}_j^{(i)} = \mathsf{Sign}_c(\mathsf{sk}_c, \hat{h}_j^{(i)})$. Also pick $q$ random functions $O_i$ mapping $m$ to $[\ell]$. Then let $h_m^{(i)} = \hat{h}_{O_i(m)}^{(i)}$ and $\sigma_m^{(i)} = \hat{\sigma}_{O_i(m)}^{(i)}$. Let $T_i$ be random functions, and let $t_m^{(i)} = T_i(m)$. The only difference between Game 2 and Game 3 is that the $h_m^{(i)}$ and $\sigma_m^{(i)}$ values were generated by $q$ small-range distributions on $\ell$ samples. Each of the small-range distributions is only queried once, so Lemma 2.4 implies that the success probability is still at least $\epsilon - \mathsf{negl} - 1/2p$.

**Game 4.** Let the $O_i$ and $T_i$ be pairwise independent functions. The adversary cannot tell the difference.

Notice that Game 4 can now be simulated efficiently, and $A$ wins in this game with probability $\epsilon - \mathsf{negl} - 1/2p$. Let $h_k^* = \mathsf{H}(\mathsf{pk}, m_k^*, r_k^*)$ be the hashes of the forgeries. Since we have no collisions in $\mathsf{H}$, the pairs $(h_k^*, \sigma_k^*)$ are distinct. Let $\mathcal{H}^{(i)} = \{\hat{h}_j^{(i)}\}$ be the set of $\hat{h}$ values used to answer query $i$, and $\mathcal{H}$ be the union of the $\mathcal{H}^{(i)}$. There are two possibilities:

- At least one of the $h_k^*$ is not in $\mathcal{H}$, or two of them are equal. In this case, we can obtain a forger $B_0$ for $\mathcal{S}_c$, which is given $\mathsf{pk}_c$ and simulates Game 4 exactly: To generate the $(\hat{h}_j^{(i)}, \hat{\sigma}_j^{(i)})$ pairs, $B_0$ asks its own challenger for signatures on $q\ell$ random messages. When $A$ responds with forgeries $(m_k^*, r_k^*, \sigma_k^*)$, $B_0$ computes $h_k^* = \mathsf{H}(\mathsf{pk}_H, m_k^*, r_k^*)$, and finds the $k$ value such that $h_k^* \notin \mathcal{H}$, or the $k_0, k_1$ such that $h_{k_0}^* = h_{k_1}^*$. In the latter case, one of the $\sigma_{k_b}^*$ was not the result of a signing query, so let $k = k_b$. It then outputs the pair $(h_k^*, \sigma_k^*)$. Then $B_0$ never received the signature $\sigma_k^*$ on $h_k^*$, so this is a valid forgery. Therefore, this event happens with negligible probability.
- All of the $h_k^*$ values are distinct and lie in $\mathcal{H}$. In this case, there is some $i$ such that two $h_k^*$ values are in $\mathcal{H}^{(i)}$ for the same $i$. Notice that this event happens, and all the forgeries are valid, with probability $\epsilon - \mathsf{negl} - 1/2p$.

**Game 5.** Now we guess a random query $i^*$ and add a check that all the $h_k^*$ values lie in $\mathcal{H}$, and that two of them are distinct and lie in $\mathcal{H}^{(i^*)}$. Without loss of generality, assume these two $h^*$ values

are $h_0^*$ and $h_1^*$. $A$ then wins in this game with probability $\epsilon/q - \mathsf{negl} - 1/2pq$. Let $j_b^*$ be the $j$ such that $h_b^* = \hat{h}_{j_b^*}^{(i^*)}$ for $b = 0, 1$.

**Game 6.** On query $i^*$, measure the value of $O_i(m)$, to get a value $j^*$. $O_i$ takes values in $[\ell]$, so Lemma 2.1 says the adversary's success probability is still at least $\epsilon/q\ell - \mathsf{negl} - 1/2pq\ell$. Notice now that for query $i^*$, the challenger only needs to sign $\hat{h}_{j^*}^{(i^*)}$, and therefore, one of the $h_b^* = \hat{h}_j^{(i^*)}$ values was never signed.

**Game 7.** Now guess at the beginning of the game the value of $j^*$, and at the end, check that the guess was correct. The adversary still wins with probability $\epsilon/q\ell^2 - \mathsf{negl} - 1/2pq\ell^2$.

We now describe an adversary $B_1$ that breaks the security of $\mathcal{S}_c$. Ask the RMA challenger for $(q-1)\ell + 1$ random messages and corresponding signatures. For $j \neq j^*$, choose $\hat{h}_j^{(i^*)}$ randomly. Set the rest of the $\hat{h}_j^{(i)}$ values to be the signed messages, and set $\hat{\sigma}_j^{(i)}$ to be the corresponding signatures. Now play the role of challenger to $A$ in Game 7 using these values for $\hat{h}_j^{(i)}$ and $\hat{\sigma}^{(i)}$. As discussed above, $B_1$ will never have to sign a message it does not have a signature for. Now if $A$ wins, it means that it it produced an $\mathcal{S}_c$ signature for some $\hat{h}_j^{(i^*)}$ with $j \neq j^*$. Since $B_1$ never saw a signature on $\hat{h}_j^{(i^*)}$, this is a valid forgery. The security of $\mathcal{S}_c$ implies therefore that $\epsilon/q\ell^2 - \mathsf{negl} - 1/2pq\ell^2$ is negligible. Thus $\epsilon - 1/2p$ is negligible. Since $\epsilon > 1/p$ infinitely often, we then have $1/2p < \mathsf{negl}$ infinitely often, a contradiction. Therefore, $\epsilon$ is negligible. □

We note that for one-time security, this security reduction signs only a single message, so we only need to rely on the one-time security of $\mathcal{S}_c$.

### 3.3 Signatures in the Quantum Random Oracle Model

In this section we present a generic conversion from any classical signature scheme to a scheme secure against quantum chosen message attacks in the quantum random oracle model. We also show that the deterministic signature scheme of Gentry, Peikert, and Vaikuntanathan [GPV08] is secure in this model.

Recall that when a random oracle scheme is implemented in the real-world, the random oracle is replaced by a concrete hash function $H$, thereby enabling a quantum adversary to evaluate $H$ on a superposition of inputs. Therefore, security proofs in the random oracle model must allow all parties, including the adversary, to issue *quantum* queries to $H$. This model is called the *quantum* random oracle model [BDF+11] and is the one we use here.

#### 3.3.1 A Generic Conversion

First, we demonstrate a simple generic conversion from a classical signature scheme to one that is secure against an adaptive *quantum* chosen message attack in the quantum random oracle model. The construction is quite simple: use the random oracle to hash the message along with a random salt, and send the signature on the hash, together with the salt. This construction is very appealing since messages are often hashed anyway before signing. The results in this section then show that only minor modifications to existing schemes are necessary to make them quantum immune.

**Construction 3.12.** *Let $\mathcal{S}_c = (\mathsf{G}_c, \mathsf{Sign}_c, \mathsf{Ver}_c)$ be a signature scheme, $H$ be a hash function, and $\mathcal{Q}$ be a family of pairwise independent functions mapping messages to the randomness used by $\mathsf{Sign}_c$, and $k$ some polynomial in $\lambda$. Define $\mathcal{S} = (\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ where:*

$$\mathsf{G}(\lambda) \ = \mathsf{G}_c(\lambda)$$

$$\mathsf{Sign}(\mathsf{sk}, m) : Q \xleftarrow{R} \mathcal{Q},\ r \xleftarrow{R} \{0,1\}^k$$
$$s \leftarrow Q(m),\ h \leftarrow H(m, r),\ \sigma \leftarrow \mathsf{Sign}_c(\mathsf{sk}, h; s),\ \textit{output}\ (r, \sigma)$$

$$\mathsf{Ver}(\mathsf{pk}, m, (r, \sigma)) : h \leftarrow H(m, r),\ \textit{output}\ \mathsf{Ver}_c(\mathsf{pk}, h, \sigma)$$

We note that Construction 3.12 is similar to Construction 3.10: instead of the chameleon hash $\mathsf{H}(\mathsf{pk}, \cdot, \cdot)$ we have a random oracle $H(\cdot, \cdot)$, and instead of generating a different $r$ for each message in the superposition, we just generate a single $r$ for the entire superposition. We can achieve security for Construction 3.12, assuming only a very weak form of security for $\mathcal{S}_c$, namely, universal unforgeability under a random message attack (UUF-RMA security):

**Theorem 3.13.** *If $\mathcal{S}_c$ is strongly (resp. weakly) UUF-RMA secure, then $\mathcal{S}$ in Construction 3.12 is strongly (resp. weakly) EUF-qCMA secure in the quantum random oracle model. Moreover, if $\mathcal{S}_c$ is only one-time secure, then $\mathcal{S}$ is also one-time secure.*

Before proving Theorem 3.13, we explain how to realize the strong UUF-RMA notion of security. We note that any strongly EUF-RMA or EUF-CMA secure signature scheme satisfies this security notion. We also note that some weaker primitives do as well. The first is pre-image sampleable functions, defined by Gentry et al. [GPV08]:

**Definition 3.14** (PSF)**.** *A* pre-image sampleable trapdoor function (PSF) *is a tuple of algorithms $\mathsf{PSF} = (\mathsf{G}, \mathsf{Sample}, F, F^{-1})$ with the following properties:*

- $\mathsf{G}(\lambda)$ *generates a secret/public key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $F(\mathsf{pk}, \cdot)$ *is a function from set $\mathcal{X}_\lambda$ to set $\mathcal{Y}_\lambda$.*
- $\mathsf{Sample}(\lambda)$ *samples an $x$ from some on $\mathcal{X}_\lambda$, such that $F(\mathsf{pk}, x)$ is uniform over $\mathcal{Y}_\lambda$.*
- $F^{-1}(\mathsf{sk}, y)$ *takes an image $y \in \mathcal{Y}_\lambda$, and outputs an $x$ such that $F(\mathsf{pk}, x) = y$, and $x$ is distributied negligibly-close to $\mathsf{Sample}(\lambda)$ conditioned on $F(\mathsf{pk}, x) = y$.*

The two general notions of security we are interested in for PSFs are one-wayness and collision resistance. If we let $\mathsf{Sign}(\mathsf{sk}, m) = F^{-1}(\mathsf{sk}, m)$ and $\mathsf{Ver}(\mathsf{pk}, m, \sigma) = F(\mathsf{pk}, \sigma) == m$, then one-wayness plus collision resistance implies strong UUF-RMA security.

**Corollary 3.15.** *If* PSF *is a collision resistant and one-way PSF, then Construction 3.12 instantiated with* PSF *is strongly EUF-qCMA secure in the quantum random oracle model.*

Gentry et al. [GPV08] show how to construct a PSF that is collision-resistant and one-way under the assumption that SIS is hard. Therefore, we can construct efficient signatures in the quantum random oracle model based on SIS. Later, we also show that the basic GPV signature scheme is secure in the quantum random oracle model, though the proof is very different.

A trapdoor permutation is a PSF where $D_\lambda$ is the uniform distribution and $F(\mathsf{pk}\cdot)$ is bijective. Trapdoor permutations are trivially collision resistant, since they have no collisions.

**Corollary 3.16.** *If $\mathcal{F}$ is a one-way trapdoor permutation, then Construction 3.12 instantiated with $\mathcal{F}$ is strongly EUF-qCMA secure in the quantum random oracle model.*

Next, we observe that any adversary $A$ breaking the universal unforgeability of $\mathcal{S}_c$ by mounting a random message attack can easily be transformed into an adversary $B$ breaking Construction 3.12 under a *classical* chosen message attack in the *classical* random oracle model:

- When $B$ receives the public key $\mathsf{pk}$ for $\mathcal{S}$ in Consutrction 3.12, $B$ forwards the public key to $A$.
- $A$ requests $q$ message/signature pairs for random messages, and $n$ additional random messages. To respond, $B$ queries its signing oracle on $q$ arbitrary distinct points $m_i$, obtaining $q$ pairs $(r_i, \sigma_i)$, where $\sigma_i$ is a valid $\mathcal{S}_c$ signature of $h_i = H(m_i, r_i)$. $B$ queries its random oracle on $m_i, r_i$ to obtain $h_i$, and sends the $q$ pairs $(h_i, \sigma_i)$ as the message/tag pairs to $A$. Additionally, $B$ queries its random oracle on $n$ additional arbitrary points $m_i^*, r_i^*$, obtaining $h_i^*$, and sends the $h_i^*$ to $A$ as the $n$ additional messages.
- Finally, $A$ outputs a new signature $\sigma_i^*$ for one on the messages $h_i^*$, or potentially one of the $h_i$ if we are interested in strong security. $B$ simply figures out which pre-image $(m_i^*, r_i^*)$ this forgery corresponds to, and outputs the tuple $(m_i^*, r_i^*, \sigma^*)$.

Together with Theorem 3.13, this roughly means that quantum chosen message queries and quantum random oracle queries do not help the adversary break Construction 3.12. Therefore, if a scheme matches the form of Construction 3.12, it is only necessary to prove classical security. This is formalized by the following corollary:

**Corollary 3.17.** *If $\mathcal{S}$ in Construction 3.12 is weakly (resp. strongly) existentially unforgeable under a* classical *chosen message attack performed by a quantum adversary, then it is also weakly (resp. strongly) exististentially unforgeable under a* quantum *chosen message attack.*

We now sketch the proof of Theorem 3.13. The complete proof is in Appendix A.1.

**Proof sketch.** Given the similarities between Constructions 3.10 and 3.12, the proof is similar to that of Theorem 3.11. For classical security in the classical random oracle model, the adversary only sees a polynomial number of outputs of $H$. We can set these outputs to be exactly the messages produced by the $\mathcal{S}_c$ challenger. Moreover, we can set the outputs in a way so that we can answer signing queries using the signatures provided by the $\mathcal{S}_c$ challenger with non-negligible probability. For quantum security in the quantum random oracle model, using this approach directly would require the $\mathcal{S}_c$ challenger to output exponentially many random messages, and sign an exponential number of them. Similar to the proof of Theorem 3.11, we can overcome this difficulty using small-range distributions. However, now the number of signatures received from the $\mathcal{S}_c$ challenger is a large polynomial, whereas the adversary only produces $q+1$ $\mathcal{S}$ forgeries. To show that one of the forgeries still corresponds to an $\mathcal{S}_c$ forgery, we perform a partial measurement on one of the queries, so that the adversary only sees a single signature for that query. Since the adversary produced $q+1$ forgeries, two of them must correspond to the same query, so one of the $\mathcal{S}$ forgeries must actually be an $\mathcal{S}_c$ forgery. □

### 3.3.2 Deterministic GPV Signatures

Now we show that the basic deterministic GPV signature scheme is secure. For completeness, we present the GPV signature scheme built from pre-image sampleable functions and PRFs, and prove its security:

**Construction 3.18.** *Let* $\mathsf{PSF} = (\mathsf{G}_{psf}, \mathsf{Sample}, F, F^{-1})$ *be a pre-image sampleable function,* $\mathsf{PRF}$ *be a pseudorandom function, and* $H$ *a hash function. Let* $\mathcal{S} = (\mathsf{G}, \mathsf{Sign}, \mathsf{Ver})$ *where*

$$\mathsf{G}(\lambda) : (\mathsf{sk}', \mathsf{pk}') \xleftarrow{R} \mathsf{G}_{psf}(\lambda),\ k \xleftarrow{R} \{0,1\}^\lambda$$
$$\textit{output } \mathsf{sk} = (\mathsf{sk}', k), \mathsf{pk} = \mathsf{pk}'$$

$$\mathsf{Sign}((\mathsf{sk}, k), m) : r \leftarrow \mathsf{PRF}(k, m)\ h \leftarrow H(m),\ \textit{output } \sigma = F^{-1}(\mathsf{sk}, h; r)$$

$$\mathsf{Ver}(\mathsf{pk}, m, \sigma) : h \leftarrow H(m),\ h' \leftarrow F(\mathsf{pk}, \sigma),\ \textit{accept if and only if } h = h'$$

We say that $\mathsf{PSF}$ has large pre-image min-entropy if, for all $\mathsf{pk}$,

$$\max_{y \in \mathcal{Y}} \Pr[x \leftarrow \mathsf{Sample}(\lambda) : F(\mathsf{pk}, x) = y] < 2^{-\omega(\log \lambda)}$$

We note that the PSF given by Gentry et al. [GPV08] has large pre-image min-entropy.

**Theorem 3.19.** *If* $\mathsf{PSF}$ *is collision resistant and has large pre-image min-entropy, then* $\mathcal{S}$ *from Construction 3.18 is EUF-qCMA secure.*

**Proof**. We prove security via a sequence of games:

**Game 0.** This is the standard security game. The adversary wins with probability $\epsilon$.

**Game 1.** Replace $\mathsf{PRF}$ with a truly random function. The security of $\mathsf{PRF}$ implies that the adversary wins with probability at least $\epsilon - \mathsf{negl}$.

**Game2.** We change the way we answer signing queries and oracle queries as follows: Pick a random function $J$ that maps messages to the randomness used by $\mathsf{Sample}(\lambda)$. We implement the signing oracle as $S(m) = \mathsf{Sample}(\lambda; J(m))$. That is, signatures are random samples from $D_\lambda$, where the randomness used in the sampling is obtained by $J(m)$. We implement the random oracle as $H(m) = F(\mathsf{pk}, S(m))$. The adversary wins if he can produce $q+1$ $(m_i, \sigma_i)$ pairs where $H(m_i) = F(\mathsf{pk}, \sigma_i)$. This corresponds to $F(\mathsf{pk}, S(m_i)) = F(\mathsf{pk}, \sigma_i)$. In other words, $S(m_i)$ and $\sigma_i$ form a collision. By the collision resistance of $\mathsf{PSF}$, we must have $S(m_i) = \sigma_i$ for all $i$, except with negligible probability. This means that we make $q$ queries to the oracle $S$ and a polynomial number of queries to the oracle $F(\mathsf{pk}, S(\cdot))$, and output $q+1$ input/output pairs of $S$ with probability $\epsilon - \mathsf{negl}$.

Even if the adversary is able to completely learn the oracle $H(\cdot) = F(\mathsf{pk}, S(\cdot))$, the oracle $S(\cdot)$ is unpredictable to the adversary. In particular, $S(m)$ is a random pre-image of $H(m)$, which has minentropy at least $H_\infty = \omega(\log \lambda)$. Therefore, Game 2 satisfies the conditions of Lemma 2.6, meaning the probability $A$ wins in Game 2 is at most $(q+1)/\left\lfloor 2^{H_\infty} \right\rfloor < (q+1)2^{-\omega(\log \lambda)}$, which is negligible. Therefore, $A$ wins in Game 0 with negligible probability, as desired. □

### 3.4 Signatures from Generic Assumptions

In this section, we show how to construct signatures from generic assumptions. We first construct one-time signatures from one-way functions using the basic Lamport construction [Lam79]. We then expand the message space to handle arbitrary-length messages using collision resistance, and finally plug these one-time signatures into the Merkle signature scheme [Mer87]. The end result is a signature scheme whose quantum security relies only on the existence of collision-resistant functions:

**Theorem 3.20.** *If there exists a collision-resistant hash function, then there exists a strongly EUF-qCMA secure signature scheme.*

**Lamport Signatures.** We now give the basic Lamport scheme [Lam79] and prove its security:

**Construction 3.21.** *Let $F$ be a one-way function. We define the following signature scheme for $n$-bit messages:*

$$\begin{aligned}
\mathsf{G}(\lambda) &: \textit{for each } i \in [n], b \in \{0,1\} : x_{i,b} \xleftarrow{R} \{0,1\}^\lambda,\ y_{i,b} \xleftarrow{R} F(x_i) \\
&\quad \textit{output } \mathsf{sk} = (x_{i,b})_{i\in[n],b\in\{0,1\}}, \mathsf{pk} = (y_{i,b})_{i\in[n],b\in\{0,1\}} \\
\mathsf{Sign}(\mathsf{sk}, m) &: \textit{write } \mathsf{sk} = (x_{i,b})_{i\in[n],b\in\{0,1\}} \\
&\quad \textit{output } (x_{i,m_i})_{i\in[n]} \\
\mathsf{Ver}(\mathsf{pk}, m, \sigma) &: \textit{write } \mathsf{pk} = (y_{i,b})_{i\in[n],b\in\{0,1\}}, \sigma = (x'_i)_{i\in[n]} \\
&\quad \textit{accept if and only if } F(x'_i) = y_{i,m_i} \textit{ for all } i \in [n]
\end{aligned}$$

**Theorem 3.22.** *If $F$ is one-way (resp. second pre-image resistant), then the Lamport signature scheme built from $F$ is weakly (resp. strongly) one-time EUF-CMA secure.*

**Proof**. We prove the weak security case; the strong security case is almost identical. Let $A$ be an adversary that makes a single quantum query to $\mathsf{Sign}$ and outputs a pair of valid message/signature pairs for different messages with probability $\epsilon$. We prove security through a sequence of games.

**Game 0.** This is the standard attack game, where $A$ wins with probability $\epsilon$.

**Game 1.** Pick a random value $i^* \in [n]$. Abort if both messages in $A$'s forgery are the same for index $i^*$. $A$ still wins with probability $\epsilon/n$.

**Game 2.** For the quantum chosen message query, measure the bit $i^*$ of the message superposition. Lemma 2.1 shows that $A$ still wins with probability $\epsilon/2n$.

**Game 3.** At the beginning of the game, guess a bit $b^*$ at random, and abort if the outcome of the measurement in Game 2 is $b^*$. $A$ still wins with probability $\epsilon/4n$.

We can now describe an adversary $B$ that inverts $F$. On input $y$, $B$ guesses $i^* \in [n]$ and $b^* \in \{0,1\}$, and sets $y_{i^*,b^*} = y$. For $(i,b) \neq (i^*,b^*)$, $B$ picks $x_{i,b}$ at random, and lets $y_{i,b} = F(x_{i,b})$. Now $B$ simulates Game 3. With probability at least $\epsilon/4n$, $B$ is able to answer $A$'s query, and $A$ produces valid forgeries whose messages differ on bit $i^*$. This means $A$ produces pre-images $x'_{i^*,0}, x'_{i^*,1}$ for $y_{i^*,0}, y_{i^*,1}$. $B$ outputs $x'_{i^*,b^*}$, which is a valid pre-image for $y_{i*,b^*} = y$.

□

The signatures from Construction 3.21 have public keys that are much longer than the messages being signed. In order to use Lamport signatures in the Merkle signature scheme, we need to be able to sign much larger messages. In the classical setting, it is possible to expand the message space using target collision resistant functions. These can in turn be built from one-way functions, showing that classical signatures can be built from the minimal assumption of one-way functions.

Unfortunately, the notion of target collision resistance no longer makes sense in the quantum setting, and we therefore have to resort to collision resistance. We can thus build one-time signatures for arbitrary-length messages assuming only collision resistance.

**Merkle Signatures.** Now we show how to use such signatures to build Merkle many-time signatures [Mer87]. For completeness, we give the construction. We will have a tree of depth $d$, where each non-leaf node contains a pair of private/public key pairs for the one-time signature scheme, one for each child. The private/public keys for the system will be the keys for the root. To sign a message, a random leaf node is chosen. For each non-leaf node in the path from root to leaf, sign the node's public keys with the corresponding secret key of the parent. Then use the correct secret key from the leaf's parent to sign the message. This tree is exponential in size, so we will use a PRF to generate the keys. In more detail:

**Construction 3.23.** *Let $\mathcal{S}_{ot} = (\mathsf{G}_{ot}, \mathsf{Sign}_{ot}, \mathsf{Ver}_{ot})$ be a one-time signature scheme. Also let $F$ be a secure PRF. The stateless Merkle signature scheme $\mathcal{S} = (\mathsf{G}, \mathsf{SignVer})$ is defined as follows:*

- $\mathsf{G}(\lambda)$*: run $\mathsf{G}_{ot}$ twice to get two secret/public key pairs $(\mathsf{sk}_b, \mathsf{pk}_b)$ for $b = 0, 1$. Also choose a random $\lambda$-bit string $k$. The secret key is $\mathsf{sk} = (\mathsf{sk}_0, \mathsf{sk}_1, k)$ and the public key is $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$.*
- $\mathsf{Sign}(\mathsf{sk}, m)$*: to sign a message $m$, first pick a random bit string $\mathbf{a}$ in $\{0,1\}^d$. Then for $i = 1, ..., d-1$,*
  - *Let $(r_{(\mathbf{a}_{[1,i]},0)}, r_{(\mathbf{a}_{[1,i]},1)}, s_{\mathbf{a}_{[1,i]}}) = F(k, \mathbf{a}_{[1,i]})$.*
  - *Let $(\mathsf{sk}_{(\mathbf{a}_{[1,i]},b)}, \mathsf{pk}_{(\mathbf{a}_{[1,i]},b)}) = \mathsf{G}_{ot}(\lambda; r_{(\mathbf{a}_{[1,i]},b)})$ for $b = 0, 1$.*
  - *Let $\sigma_{\mathbf{a}_{[1,i]}} = \mathsf{Sign}_{ot}(\mathsf{sk}_{\mathbf{a}_{[1,i]}}, (\mathsf{pk}_{(\mathbf{a}_{[1,i]},0)}, \mathsf{pk}_{(\mathbf{a}_{[1,i]},1)}); s_{\mathbf{a}_{[1,i]}})$*

  *Let $\Sigma_{\mathbf{a}} = (\mathbf{a}, (\mathsf{pk}_{(\mathbf{a}_1,0)}, \mathsf{pk}_{(\mathbf{a}_1,1)}, \sigma_{\mathbf{a}_1}), ..., (\mathsf{pk}_{(\mathbf{a}_{[1,d-1]},0)}, \mathsf{pk}_{(\mathbf{a}_{[1,d-1]},1)}))$,*
  *and let $\sigma_{\mathbf{a}}(m) = \mathsf{Sign}_{ot}(\mathsf{sk}_{\mathbf{a}}, m)$. Output the signature $(\Sigma_{\mathbf{a}}, \sigma_{\mathbf{a}}(m))$.*
- $\mathsf{Ver}(\mathsf{pk}, m, \Sigma, \sigma)$*: parse $\Sigma$ as $(\mathbf{a}, (\mathsf{pk}_{(\mathbf{a}_1,0)}, \mathsf{pk}_{(\mathbf{a}_1,1)}, \sigma_{\mathbf{a}_1}), ..., (\mathsf{pk}_{(\mathbf{a}_{[1,d-1]},0)}, \mathsf{pk}_{(\mathbf{a}_{[1,d-1]},1)}))$. For $i = 1, ..., d-1$,*
  - *If $\mathsf{Ver}_{ot}(\mathsf{pk}_{\mathbf{a}_{[1,i]}}, (\mathsf{pk}_{(\mathbf{a}_{[1,i]},0)}, \mathsf{pk}_{(\mathbf{a}_{[1,i]},1)}))$ rejects, then reject and stop.*

  *Then output the output of $\mathsf{Ver}_{ot}(\mathsf{pk}_{\mathbf{a}}, m, \sigma)$.*

We note that if we allow state, we can pick the random bit string $\mathbf{a}$ incrementally for each query. Then we can actually save the $\mathsf{sk}_{(\mathbf{a}_{[1,i]},b)}, \mathsf{pk}_{(\mathbf{a}_{[1,i]},b)}, \sigma_{\mathbf{a}_{[1,i]}}$ values until we do not need them any more, and remove the need for a PRF to generate randomness. In this way, we obtain the stateful Merkle signature scheme.

**Theorem 3.24.** *If $\mathcal{S}_{ot}$ is weakly (resp. strongly) one-time EUF-qCMA secure, then both the stateless and stateful Merkle Signatures built from $\mathcal{S}_{ot}$ are weakly (resp. strongly) EUF-qCMA secure.*

**Proof sketch.** The proof is very similar to the classical proof. Notice that each secret key in all but the bottom level are used to sign exactly one message: the pair of public keys in the corresponding child. Moreover, the secret keys on the bottom level are used to answer only one (potentially quantum) signature query. Therefore, the security of the one-time signature scheme implies that no adversary can forge messages for the Merkle signature scheme. For completeness, we give the complete proof in Appendix A.2. □

# 4 Quantum-Secure Encryption Schemes

We now turn to encryption schemes where we first discuss an adequate notion of security under quantum queries. In what follows, we will discuss symmetric key schemes; the discussion for public key schemes is similar. At a high level, our notion of security allows quantum encryption and decryption queries, but requires challenge queries to be *classical*. One might hope for an entirely quantum game, where challenge queries are quantum as well, but we show that such fully-quantum security definitions are unsatisfiable.

We start by developing a notion of CPA security where encryption queries are allowed to be quantum. Since finding an attainable definition is non-trivial we first present a few alternatives and then converge to a workable definition (Definition 4.5). Once we arrive at a suitable definition for CPA security we will also obtain a corresponding definition for CCA security. Our first attempt at defining quantum CPA security is as follows:

**Definition 4.1.** *A symmetric key encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is indistinguishable under a fully quantum chosen plaintext attack (IND-fqCPA secure) if no efficient adversary $A$ can win the following game, except with probability at most $1/2 + \mathsf{negl}$:*

**Key Gen** *The challenger picks a random key $k$ and a random bit $b$.*

**Encryption Queries** *$A$ is allowed to make chosen message queries on superpositions of message pairs. For each such query, the challenger chooses randomness $r$, and encrypts the appropriate message in each pair using $r$ as randomness:*

$$\sum_{m_0,m_1,c} \psi_{m_0,m_1,c} |m_0,\ m_1,\ c\rangle \quad \longrightarrow \quad \sum_{m_0,m_1,c} \psi_{m_0,m_1,c} |m_0,\ m_1,\ c \oplus \mathsf{Enc}(k, m_b; r)\rangle$$

**Guess** *$A$ produces a bit $b'$, and wins if $b = b'$.*

Definition 4.1 captures a scheme where we can encrypt a superposition of messages by encrypting each message in the superposition separately, and no efficient adversary can learn anything about the plaintext superposition. Unfortunately, this definition is not achievable:

**Theorem 4.2.** *No encryption scheme $\mathcal{E}$ satisfies the security notion of Definition 4.1.*

**Proof**. We construct a generic adversary $A$. $A$ prepares three registers: two plaintext registers and a ciphertext register. $A$ puts a uniform superposition of all messages in the first register, and 0 in the second plaintext and ciphertext registers. $A$ submits these three registers as a chosen message query. If $b = 0$, the ciphertext register will contain the encryptions of the messages in the superposition. If $b = 1$, it will contain the encryption of 0. $A$ then measures the ciphertext register. If $b = 0$, the resulting state will be the purely classical state $(m, 0, \mathsf{Enc}(k, m))$ for a random message $m$. If $b = 1$, the measurement does nothing, so the first register still contains a superposition of all messages. $A$ now performs the quantum Fourier transform to the first message register and measures. If $b = 0$, the transform will place a uniform superposition of all messages in the first register, and measuring will give a random message. If $b = 1$, the transform will place 0 in the first register. Thus, $A$ distinguishes $b = 0$ from $b = 1$ with probability exponentially-close to 1. □

The problem with Definition 4.1 is that the message query is entangled with the ciphertext response, and this entanglement depends on which register gets encrypted. Another reasonable idea is to encrypt *both* message registers, but flip which register each ciphertext is written to depending on the value of $b$:

**Definition 4.3.** *A symmetric key encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is indistinguishable under a fully quantum chosen left-right plaintext attack (IND-lrCPA secure) if no efficient adversary $A$ can win in the following game, except with probability at most $1/2 + \mathsf{negl}$:*

**Key Gen** *The challenger picks a random key $k$ and a random bit $b$.*

**Encryption Queries** *$A$ is allowed to make chosen message queries. For each such query, the challenger chooses randomness $r_0, r_1$, and responds with the encryptions of both messages in the pair, but in an order determined by $b$:*

$$\sum_{m_0,m_1,c_1,c_2} \psi_{m_0,m_1,c_1,c_2} |m_0,\ m_1,\ c_1,\ c_2\rangle \quad \longrightarrow$$
$$\sum_{m_0,m_1,c_1,c_2} \psi_{m_0,m_1,c_1,c_2} |m_0,\ m_1,\ c_1 \oplus \mathsf{Enc}(k, m_b; r_0),\ c_2 \oplus \mathsf{Enc}(k, m_{1-b}; r_1)\rangle$$

**Guess** *$A$ produces a bit $b'$, and wins if $b = b'$.*

Unfortunately, this definition turns out to be at least as strong as Definition 4.1, and so it is also unattainable:

**Theorem 4.4.** *No encryption scheme $\mathcal{E}$ satisfies the notion of security in Definition 4.3. In particular, any encryption scheme that is secure in the sense of Definition 4.3 is also secure in the sense of Definition 4.1.*

**Proof**. Suppose we have an adversary $A$ for Definition 4.1. We will convert it into an adversary $B$ for Definition 4.3. $B$ simulates $A$ forwarding encryption queries as follows: When $A$ makes an encryption query, $B$ adds a second ciphertext register, and puts into it a uniform superposition over all ciphertexts. $B$ then sends the resulting state to its challenger as its encryption query. The answer to this query does not affect the second ciphertext register, so $B$ can uncompute it. $B$ then passes the resulting state back to $A$. $B$ perfectly simulates $A$'s view, and therefore $B$ breaks the security of $\mathcal{E}$ under Definition 4.3. □

Our attempts to make the entire security game quantum lead to an adversary that can always win. Therefore, we must force encryption queries to be classical. We do, however, wish to allow the adversary to encrypt superpositions of messages, but not have the response depend in any way on $b$. Therefore, we propose separating encryption queries into classical challenge queries and quantum encryption queries. This gives the following definition:

**Definition 4.5.** *A symmetric key encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is indistinguishable under a quantum chosen message attack (IND-qCPA secure) if no efficient adversary $A$ can win in the following game, except with probability at most $1/2 + \mathsf{negl}$:*

**Key Gen** *The challenger picks a random key $k$ and a random bit $b$.*

**Queries** *$A$ is allowed to make two types of queries:*

- **Challenge queries** *$A$ sends two messages $m_0, m_1$, to which the challenger responds with $c^* = \mathsf{Enc}(k, m_b)$.*
- **Encryption queries** *For each such query, the challenger chooses randomness $r$, and encrypts each message in the superposition using $r$ as randomness:*

$$\sum_{m,c} \psi_{m,c} |m,\ c\rangle \quad \longrightarrow \quad \sum_{m,c} \psi_{m,c} |m,\ c \oplus \mathsf{Enc}(k, m; r)\rangle$$

**Guess** *A produces a bit $b'$, and wins if $b = b'$.*

This definition has another advantage: since challenge queries are classical, when we move to CCA security, we can check if a ciphertext was the result of a challenge query and reject decryption queries for these ciphertexts. This gives us the following notion of CCA security:

**Definition 4.6.** *A symmetric key encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ is indistinguishable under a quantum chosen message attack (IND-qCCA secure) if no efficient adversary A can win in the following game, except with probability at most $1/2 + \mathsf{negl}$:*

**Key Gen** *The challenger picks a random key $k$ and a random bit $b$. It also creates a list $\mathcal{C}$ which will store challenger ciphertexts.*

**Queries** *A is allowed to make three types of queries:*

**Challenge queries** *A sends two messages $m_0, m_1$, to which the challenger responds with $c^* = \mathsf{Enc}(k, m_b)$. The challenger also adds $c^*$ to $\mathcal{C}$.*

**Encryption queries** *For each such query, the challenger chooses randomness $r$, and encrypts each message in the superposition using $r$ as randomness:*

$$\sum_{m,c} \psi_{m,c} |m,\ c\rangle \qquad \longrightarrow \qquad \sum_{m,c} \psi_{m,\ c} |m, c \oplus \mathsf{Enc}(k, m; r)\rangle$$

**Decryption queries** *For each such query, the challenger decrypts all ciphertexts in the superposition, except those that were the result of a challenge query:*

$$\sum_{c,m} \psi_{c,m} |c,\ m\rangle \qquad \longrightarrow \qquad \sum_{c,m} \psi_{c,m} |c,\ m \oplus f(c)\rangle$$

*where*

$$f(c) = \begin{cases} \perp & \text{if } c \in \mathcal{C} \\ \mathsf{Dec}(k, c) & \text{otherwise} \end{cases}$$

**Guess** *A produces a bit $b'$, and wins if $b = b'$.*

In the above definition, we need to define the operation $m \oplus \perp$. Since the query responses will xor $\perp$ with different messages, we need a convention that makes this operation reversible. Taking $\perp$ to be some bit string that lies outside of the message space, and $\perp \oplus m$ to be bitwise xor will suffice.

Note that we implicitly assume that the decryption algorithm is deterministic. This will be true of our encryption schemes. We note that this is not a limiting assumption since one can always make the decryption algorithm deterministic by deriving the randomness for decryption from a PRF applied to the ciphertext. Also, as in the classical case, a simple hybrid argument shows that the above definition is equivelent to the case where the number of encryption queries is limited to 1. Lastly, it is straightforward to modify the above definition for public key encryption schemes.

## 4.1 A Separation Example

Here we show that quantum chosen ciphertext queries give the adversary more power than classical queries. In particular, we present a public key encryption scheme that is secure under classical queries, but completely insecure once an adversary can make quantum queries. Let $\mathcal{E}_c = (\mathsf{G}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ be an encryption scheme that is secure under classical chosen ciphertext queries. The idea of our construction is similar in spirit to that for signatures. The construction is as follows:

**Construction 4.7.** *Fix positive integers $N, N'$. Let $\mathcal{E}_c = (\mathsf{G}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ be an encryption scheme and $\mathsf{PRF}$ a pseudorandom function with domain $[0, N)$. Let $\mathsf{RPrime}(N')$ denote a procedure that samples a random prime let than $N'$. We build a new encryption scheme $\mathcal{E} = (\mathsf{G}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*

$$\begin{aligned}
\mathsf{G}(\lambda) &: (\mathsf{sk}_c, \mathsf{pk}_c) \xleftarrow{R} \mathsf{G}_c(\lambda),\ k \xleftarrow{R} \{0,1\}^\lambda,\ p \xleftarrow{R} \mathsf{RPrime}(N') \\
&\quad \textit{output } \mathsf{sk} = (\mathsf{sk}_c, k, p),\ \mathsf{pk} = \mathsf{pk}_c \\
\mathsf{Enc}(\mathsf{pk}_c, m) &: c \leftarrow \mathsf{Enc}_c(\mathsf{pk}, m) \\
&\quad \textit{output } (c, 0, 0) \\
\mathsf{Dec}((\mathsf{sk}_c, k, p), (c, a, b)) &: \begin{cases} D(\mathsf{sk}_c, c) & \textit{if } a = 0 \textit{ and } b = 0 \\ \bot & \textit{if } a = 0 \textit{ and } b \neq 0 \\ \mathsf{PRF}(k, b \mod p) & \textit{if } a = 1 \\ \mathsf{sk} & \textit{if } a = 2 \textit{ and } b = p \\ \bot & \textit{if } a = 2 \textit{ and } b \neq p \end{cases}
\end{aligned}$$

**Theorem 4.8.** *If $\mathcal{E}_c$ is secure under a classical chosen ciphertext attack, and $\mathsf{PRF}$ is secure against classical queries, then $\mathcal{E}$ in Construction 4.7 is secure under a classical chosen ciphertext attack, but totally insecure under a quantum chosen ciphertext attack.*

**Proof**. The proof is very similar to the separation for signature schemes: first we need to argue the classical security of our scheme, and then we must break the scheme using quantum queries.

For security, similar to signatures, we can modify our decryption oracle so that it always outputs $\bot$ when $a = 2$, and outputs $O(b)$ when $a = 1$, for a random oracle $O$. Any adversary breaking the original scheme will also win with this decryption oracle. However, now the decryption oracle for the cases $a = 2, 3$ is completely independent of the original encryption scheme $\mathcal{E}_c$, so such an adversary can be modified to break $\mathcal{E}_c$. Since $\mathcal{E}_c$ is secure, $\mathcal{E}$ must be secure as well.

Now we explain how quantum queries can be used to recover $\mathsf{sk}$. In phase 1, the adversary makes a single quantum query with $a = 1$ to recover $p$, and then makes a classical query with $a = 2$ on the ciphertext $(0, 2, p)$ to recover $\mathsf{sk}$.

□

## 4.2 Symmetric CCA Security

In this section, we construct symmetric-key CCA secure encryption. We will follow the encrypt-then-MAC paradigm. Ideally, we would like to show that encrypt-then-MAC, when instantiated with any IND-qCPA-secure encryption scheme and any EUF-qCMA MAC, would be CCA secure. However, it is not obvious how to prove security, as the reduction algorithm has no way to tell which ciphertexts the adversary received as the result of an encryption query, and no way to decrypt the ciphertexts if it has received them. To remedy these problems, we choose a specific encryption scheme and MAC and leave the general security proof as an open question. The encryption scheme allows us to efficiently check if the adversary has seen a particular ciphertext as a result of an encryption query, and to decrypt in this case. The construction is as follows:

**Construction 4.9.** *Let $F$ and $G$ be pseudorandom functions. We construct the following encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ where:*

$$\begin{aligned}
\mathsf{Enc}((k_1,k_2),m) : & \; r \xleftarrow{R} \{0,1\}^\lambda \\
& \; c_1 \leftarrow F(k_1,r) \oplus m, \; c_2 \leftarrow G(k_2,(r,m)) \\
& \; \textit{output } (r,c_1,c_2) \\
\mathsf{Dec}((k_1,k_2),(r,c_1,c_2)) : & \; m \leftarrow c_1 \oplus F(k_1,r), \; c_2' \leftarrow G(k_2,(r,m)) \\
& \; \textit{if } c_2 \neq c_2', \textit{ output } \perp \\
& \; \textit{otherwise, output } m
\end{aligned}$$

For security, we require $F$ to be a classically secure PRF, and $G$ to be quantum secure — secure against queries on a superposition of inputs. Zhandry [Zha12b] shows how to construct PRFs meeting this strong notion of security.

**Theorem 4.10.** *If $F$ and $G$ are quantum-secure pseudrandom functions, then $\mathcal{E}$ in Construction 4.9 is qCCA-secure.*

As demonstrated by Zhandry [Zha12b], quantum-secure pseudorandom functions can be built from any one-way function. Therefore, Theorem 4.10 shows that quantum chosen ciphertext security can be obtained from the minimal assumption that one-way functions exist. We now give the proof of Theorem 4.10:

**Proof**. We first sketch the proof: we can replace $F$ and $G$ with random functions and only negligibly affect the success probability. Since each encryption query receives a single $r$ for the entire query superposition, we can answer any encryption query by making a single query to $F$ on $r$. It is easy to check if a ciphertext $(r', c_1, c_2)$ was computed during an encryption query: just check if $r = r'$. We can also decrypt such a ciphertext, since we have seen $F(k_1, r)$. Including $c_2 = G(k_2, (r, m))$ in the ciphertext guarantees with overwhelming probability that the adversary can only submit valid ciphertexts if they were ciphertexts received during an encryption query, so we might as well reject all ciphertexts $(r', c_1, c_2)$ where $r'$ was not the randomness used in any encryption query. Now, the value of $m_b$ in the challenge query becomes perfectly hidden, which means that the distinguishing probability is 0.

We now give the complete security proof: assume we have an adversary $A$ that breaks the indistinguishability of $\mathcal{E}$ in Construction 4.9 with probability $\epsilon$. We prove security through a sequence of games.

**Game 0.** This is the standard attack game where $A$ makes $q_e$ encryption queries which are answered using randomness values $r_i$, $q_c$ challenge queries which are answered using randomness $r_i^*$, and $q_d$ decryption queries. Let $(m_{i,0}^*, m_{i,1}^*)$ denote the $i$th challenger query, and $(r_i^*, c_i^*, d_i^*)$ be the response.

**Game 1.** Replace $F$ and $G$ with truly random functions. That is, answer the $i$th encryption query by mapping $m$ to $(r_i, F(r_i) \oplus m, G(r_i, m))$, the $i$th challenge query with $(r_i^*, F(r_i^*) \oplus m_{i,b}^*, G(r_i^*, m_{i,b}^*))$, and answer decryption queries accordingly. Since $F$ and $G$ are quantum-secure pseudorandom functions, the advantage of $A$ in Game 1 is at least $\epsilon - \mathsf{negl}$.

**Game 2.** Now we abort if there is a collision among any of the $r_i$ or $r_i^*$. The probability of a collision is at most $(q_e + q_c)^2/2|\mathcal{R}|$ where $\mathcal{R}$ is the randomness space. This quantity is negligible, so $A$'s advantage is still $\epsilon - \mathsf{negl}$.

Notice that we can pick the $r_i$ values and $r_i^*$ value at the start of the game, and query $F$ on these values. Let $\mathcal{T}_i = \{r_j : j \leq i\}$ and $\mathcal{T}_i^* = \{r_j^* : j \leq i\}$. Also let $\mathcal{T} = \mathcal{T}_{q_e}$ and $\mathcal{T}^* = \mathcal{T}_{q_c}^*$. Notice that at any point, $A$ never gets to see $G(r, m)$ for any $m$ if $r \notin \mathcal{T}_i \bigcup \mathcal{T}_j^*$ where $i$ is the number of encryption queries made so far and $j$ is the number of challenge queries made so far. Note also that $A$ only gets to see $G(r_k^*, m)$ where $m = m_{k,b}^*$).

**Game 3.** For a decryption query on a superposition of ciphertexts $(r, c, d)$, let $n_e$ be the number of encryption queries made so far and $n_c$ the number of challenge queries. Check that $r \in \mathcal{T}_{n_e}$, and respond with $\perp$ for that slot otherwise. We now consider the ciphertexts that would be accepted in Game 2 but rejected in Game 3. Such ciphertexts come in two forms:

- $r \in \mathcal{T}_{n_c}^*$: Then $r = r_i^*$ for some $i$. In order to not be rejected in Game 2, we must have $c \neq c_i^*$ or $d \neq d_i^*$. In the first case, $(r, c, d)$ is an encryption of a message $m \neq m_{i,b}^*$, so the value of $G(r_i^*, m)$ is hidden to the adversary. Therefore, the probability $(r, c, d)$ is a valid ciphertext is negligible. In the second case, $(r, c, d)$ is an encryption of $m_{i,b}^*$, but then $d$ is not a valid MAC, so decryption fails.

- $r \notin \mathcal{T}_{n_e} \bigcup \mathcal{T}_{n_c}^*$: Then the value of $G(r, m)$ is completely hidden from the adversary, so the probability $d$ is a valid MAC is negligible.

Therefore, the probability of rejection for any ciphertext in Game 3 is only negligibly higher than that in Game 2. This means that with overwhelming probability, we only changed the decryption oracle on a negligible fraction of inputs, so $A$ can only distinguish Games 2 and 3 with negligible probability. Therefore, $A$'s advantage is still $\epsilon - \mathsf{negl}$.

**Game 4.** Now notice that $F$ is never queried except on the points $r_i$ and $r_i^*$. Therefore, at the start of the game, we can pick random values $f_i$ and $f_i^*$ to correspond to $F(r_i)$ and $F(r_i^*)$. We can also pick random values $g_i^*$ that correspond to $G(r_i^*, m_{i,b}^*)$ (since we only query $G$ on this point once). The adversary's view in this game is unchanged, so $A$'s advantage is at least $\epsilon - \mathsf{negl}$.

Notice that we answer the $i$th challenge query with $(r_i^*, f_i^* \oplus m_{i,b}^*, g_i^*)$, and that the values of $f_i^*$ and $g_i^*$ are never used again. This means that $m_{i,b}^*$ is statistically hidden from the adversary. Therefore, $A$'s advantage in Game 4 is identically 0, so $\epsilon = \mathsf{negl}$. □

### 4.3 Public-key CCA Security

In this section, we construct CCA-secure signatures in the public-key setting. The basic idea is to first build a selectively secure identity-based encryption scheme — whose security can be based on the Learning With Errors (LWE) Problem — and then adapt the generic transformation to CCA-security to the quantum setting:

Let $\mathcal{E}_{ibe} = (\mathsf{G}_{ibe}, \mathsf{Enc}_{ibe}, \mathsf{Dec}_{ibe}, \mathsf{Ext})$ be an IBE scheme that is selectively secure against quantum queries. It is straightforward to show that the basic IBE scheme of Agrawal, Boneh, and Boyen [ABB10] meets this security notion assuming LWE is hard. Let $\mathcal{S} = (\mathsf{G}_s, \mathsf{Sign}, \mathsf{Ver})$ be a strongly EUF-CMA secure one-time signature scheme (quantum security is unnecessary). We now

construct an encryption scheme using the generic transformation from IBE to CCA security due to Boneh et al.[BCHK04]:

**Construction 4.11.** $\mathcal{E} = (\mathsf{G}, \mathsf{Enc}, \mathsf{Dec})$ *where*

$$\begin{aligned}
\mathsf{G}(\lambda) &: \mathsf{G}_{ibe}(\lambda) \\
\mathsf{Enc}(\mathsf{mpk}, m) &: (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{G}_s(\lambda) \\
&\quad c \leftarrow \mathsf{Enc}_{ibe}(\mathsf{mpk}, \mathsf{vk}, m),\ \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, c) \\
&\quad \textit{output } (\mathsf{vk}, c, \sigma) \\
\mathsf{Dec}(\mathsf{msk}, (\mathsf{vk}, c, \sigma)) &: \textit{if } \mathsf{Ver}(\mathsf{vk}, c, \sigma) \textit{ rejects, output } \perp \\
&\quad \mathsf{sk}_{\mathsf{vk}} \leftarrow \mathsf{Ext}(\mathsf{msk}, \mathsf{vk}),\ m \leftarrow \mathsf{Dec}_{ibe}(\mathsf{sk}_{\mathsf{vk}}, c), \quad \textit{output } m
\end{aligned}$$

It is not difficult to adapt the classical security proof to the quantum setting, showing that the above construction achieves quantum CCA security:

**Theorem 4.12.** *If the* LWE *problem is hard for quantum computers, then there exists a public key encryption scheme that is IND-qCCA secure.*

# 5 Conclusion and Open Problems

We defined the notions of a quantum chosen message attack for signatures and quantum chosen ciphertext attack for encryption. We gave the first constructions of signatures and encryption schemes meeting these strong notions of security. For signatures, we presented two simpler compilers that transform classically secure schemes into quantum-secure schemes. We also showed that signatures can be built from any collision resistant hash function. For encryption, we presented both a symmetric-key and a public-key construction.

There are many directions for future work. First, can we base quantum security for signatures on the minimal assumption of one-way functions? Also, it may be possible to mount quantum superposition attacks against many cryptographic primitives. For example, can we build identification protocols or functional encryption that remain secure in the presence of such attacks?

# Acknowledgments

We thank Luca Trevisan and Amit Sahai for helpful conversations about this work. This work was supported by NSF, DARPA, the Air Force Office of Scientific Research (AFO SR) under a MURI award, Samsung, and a Google Faculty Research Award. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

# References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. *Proceedings of EUROCRYPT*, pages 1–40, 2010.

[BCHK04] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. *Proceedings of EUROCRYPT*, pages 1–31, 2004.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In *Proceedings of ASIACRYPT*, 2011.

[BHK+11] Gilles Brassard, Peter Høyer, Kassem Kalach, Marc Kaplan, Sophie Laplante, and Louis Salvail. Merkle Puzzles in a Quantum World. *Proceedings of CRYPTO*, pages 391–410, 2011.

[BL95] Dan Boneh and Richard J. Lipton. Quantum Cryptanalysis of Hidden Linear Functions. *Proceedings of CRYPTO*, 1995.

[BS08] Gilles Brassard and Louis Salvail. Quantum Merkle Puzzles. *Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008)*, pages 76–79, February 2008.

[BZ13] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. In *Proceedings of Eurocrypt*, 2013. Full version available at the Electronic Colloquium on Computational Complexity: http://eccc.hpi-web.de/report/2012/136.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of FOCS*. IEEE, 2001.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. *Proceedings of EUROCRYPT*, pages 523–552, 2010.

[DFNS11] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. *CoRR*, abs/1108.6313, 2011.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. *Proceedings of the 40th Annual ACM symposium on Theory of computing (STOC)*, page 197, 2008.

[HSS11] Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In *Proceedings of CRYPTO*, LNCS. Springer, 2011.

[IBM12] IBM Research. IBM research advances device performance for quantum computing, Feb. 2012. http://www-03.ibm.com/press/us/en/pressrelease/36901.wss.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon hashing and signatures. In *Proc. of NDSS*, pages 1–22, 2000.

[Lam79] Leslie Lamport. Constructing digital signatures from a one-way function. *Technical Report SRI-CSL-98*, 1979.

[Mer87] Ralph Merkle. A Digital Signature Based on a Conventional Encryption Function. In *Advances in Cryptology — Crypto 1987*, 1987.

[PS96] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. *Advances in Cryptology — ASIACRYPT 1996*, 96:1–12, 1996.

[Unr10] Dominique Unruh. Universally Composable Quantum Multi-Party Computation. *Proceedings of EUROCRYPT*, pages 486–505, 2010.

[Zha12a] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In *Proceedings of CRYPTO*, 2012. Full version available at the Cryptology ePrint Archives: http://eprint.iacr.org/2012/076/.

[Zha12b] Mark Zhandry. how to construct quantum random functions. In *Proceedings of FOCS*, 2012. Full version available at the Cryptology ePrint Archives: http://eprint.iacr.org/2012/182/.

# A Signature Proofs

## A.1 Proof of Theorem 3.13

**Proof**. Suppose we have an adversary $A$ that breaks the security of $\mathcal{S}$ from Construction 3.12. Let $q_S$ be the number of signing queries made by $A$, and $q_H$ be the number of hash queries (including those used in signing). We will prove security through a sequence of games.

**Game 0.** This is the standard attack game. $A$ makes $q$ quantum chosen message queries, and succeeds if it produces $q+1$ valid message/signature tuples $(m_j^*, r_j^*, \sigma_j^*)$. Let $r_i$ be the random value produced in the $i$th query, and $Q_i$ be the pair-wise independent functions. $A$'s success probability is, by assumption, some non-negligible quantity $\epsilon$. Then there is some polynomial $p(\lambda)$ such that $p(\lambda) > 1/\epsilon(\lambda)$ infinitely often.

**Game 1.** Replace the $R_i$ with a truly random function, and abort if any of the $r_i$ values are identical. Then the success probability is at least $\epsilon - q^2/2^{k+1} \geq \epsilon - \mathsf{negl}$. Notice that if all the $r_i$ are distinct, we can replace $Q_i(m)$ with $Q(m, r_i)$ for a random oracle $Q$ that is fixed across all queries. That is, we sign the $i$th query with the oracle that maps $m$ to $(r_i, \mathsf{Sign}_c(\mathsf{sk}, H(m, r_i); Q(m, r_i)))$. Notice that the function $H'(m, r) = (H(m, r), Q(m, r))$ is a random function.

**Game 2.** Let $\ell = 6C_0 p q_H^3$. We now change how $H'$ is generated. Pick three random oracles $U$, $V$ and $W$, where the codomain of $U$ and $V$ is $[\ell]$ , and let $H'(m, r) = W(U(m, V(r)), V(r))$. What this distribution represents is, for each $V(r)$ value, picking a random small-range function on $\ell$ samples. In essence, we have a small-range distribution on small-range distributions. A simple generalization of Lemma 2.4 shows that this is indistinguishable from Game 1 except with probability $C_0 q_H^3/\ell = 1/3p$.

**Game 3.** Pick the $r_i$ values up front, and let $\mathcal{R}$ be the set of $r_i$ values. Abort if $V(r_i) = V(r_j)$ for any $i \neq j$. We can assume without loss of generality that $V(r_i) = i$. The probability of this abort is at most $q_S^2/2\ell \leq 1/12C_0 q_H p < 1/3p$. Therefore, $A$ wins in Game 3 with probability at least $\epsilon - 2/3p$.

The following modifications are indistinguishable to the adversary: before the start of the game, draw $\ell^2$ different $\hat{h}_{i,j}$ values. Sign each of them with $i \leq q_S$ using $\mathcal{S}$ to get $\hat{\sigma}_{i,j}$. Then let $H(m, r) = \hat{h}_{V(r),U(m,V(r))}$ and sign the $i$th query my mapping $m$ to $(r_i, \hat{\sigma}_{i,U(m,i)})$. We can also generate $V$ and $U$ from $2q_H$-wise independent functions, and the adversary cannot tell.

**Game 4.** Pick a random $r_{i_0}$ from $\mathcal{R}$. Add the condition that if the $r_j^*$ all lie in $\mathcal{R}$, that the two that are equal must be $r_{i_0}$. This condition is independent of the view of the adversary, so the adversary wins with probability at least $(\epsilon - 2/3p)/q_S$.

**Game 5.** Measure the value of $U(m, i_0)$ for the $i_0$th query. The adversary still wins with probability at least $(\epsilon - 2/3p)/q_S\ell$.

**Game 6.** Pick a random $j_0 \in [\ell]$, and abort if the result of the measurement in Game 5 does not yield $j_0$. We guess right with probability $1/\ell$, so the adversary still wins with probability at least $(\epsilon - 2/3p)/q_S\ell^2$. Now, if we succeed, we never need the values of $\hat{\sigma}_{i_0,j}$ except for $\hat{\sigma}_{i_0,j_0}$, so we don't need to ever sign the others.

We can now describe an adversary $B$ that that attacks the UUF-RMA security of $\mathcal{S}_c$. $B$ simulates the entire Game 6, except for generating the $\hat{h}_{i,j}$ and $\hat{\sigma}_{i,j}$. For these, $B$ asks its $\mathcal{S}$ challenger for $q = (q_S - 1)\ell + 1$ random message/signature pairs, and $n = \ell^2 - q$ additional random messages. It assigns the $q$ message/signature pairs to $\hat{h}_{i,j}$ and $\hat{\sigma}_{i,j}$ for $i \in [q_S] \setminus \{i_0\}$ and $\hat{\sigma}_{i_0,j_0}$. The rest of the $\hat{h}_{i,j}$ it sets to the $n$ additional messages. When $A$ outputs its $q_S + 1$ forgery candidates, there are several possibilities:

- $r_{j_1}^*$ lies outside $\mathcal{R}$ for some $j_1$. In this case, since there are no collisions among the $(m_j^*, r_j^*)$, $h_{j_1}^* = H(m_{j_1}^*, r_{j_1}^*)$ was never signed. Therefore, $\sigma_{j_1}^*$ is a signature on a fresh message, so $B$ wins.
- All of the $r_j^*$ lie in $\mathcal{R}$, and two of them are equal. Assume without loss of generality that $r_0^* = r_1^* = r_{i_0}$. If $m_0^* = m_1^*$, then we must have $\sigma_0^* \neq \sigma_1^*$, so one of these is a fresh signature. If $m_0^* \neq m_1^*$, then $h_0^* \neq h_1^*$, so one of $h_0^*$ and $h_1^*$ was never signed. Therefore, $B$ also wins.

Since $\mathcal{S}$ is secure, $B$ wins with negligible probability, meaning $(\epsilon - 2/3p)/q_S\ell^2 < \mathsf{negl}$. This is equivalent to $\epsilon - 2/3p < \mathsf{negl}$. Since $\epsilon$ is bounded by $1/p$ infinitely often, we have that $1/3p < \mathsf{negl}$ infinitely often, contradicting the fact that $p$ is a polynomial. □

We note that, for one-time security, $q = 1$, so we only need to rely on the one-time security of $\mathcal{S}_c$.

## A.2 Proof of Theorem 3.24

**Proof**. We prove security for the stateless case, the stateful case being almost identical. Suppose we have an adversary $A$ that breaks the EUF-qCMA security of $\mathcal{S}$ with non-negligible probability $\epsilon$. We will prove security through a sequence of games.

**Game 0.** This is the standard attack game, where $A$ makes $q$ quantum queries. For $j = 1, ..., q$, let $\mathbf{a}^j$ be the vector generated for query $j$.

Notice that for any $\mathbf{b}$ with $|\mathbf{b}| \leq d - 1$, we only use the secret key $\mathsf{sk}_\mathbf{b}$ to sign a single classical message.

**Game 1.** We now replace $F(k, \cdot)$ with a truly random function. The security of $F$ implies that $A$ still wins with probability negligibly-close to $\epsilon$.

**Game 2.** Now re remove the random function all together. Instead, we keep a table mapping strings $\mathbf{b}$ to tuples $(\mathsf{sk}_{(\mathbf{b},0)}, \mathsf{sk}_{(\mathbf{b},1)}, \mathsf{pk}_{(\mathbf{b},0)}, \mathsf{pk}_{(\mathbf{b},1)}, \sigma_{\mathbf{b}})$. To answer the $j$th query, pick a random $\mathbf{a}^j$. For each $i = 1, ..., d-1$,

- Let $\mathbf{b} = \mathbf{a}^j_{[1,i]}$, and look up the tuple for $\mathbf{b}$.
- If the tuple exists, we do nothing.
- If the tuple does not exist:
  - Sample $(\mathsf{sk}_{(\mathbf{b},b)}, \mathsf{pk}_{(\mathbf{b},b)}) = \mathsf{G}_{ot}(\lambda)$ for $b = 0, 1$.
  - Obtain $\mathsf{sk}_{\mathbf{b}}$ by looking up the tuple for $\mathbf{b}_{[1,i-1]}$.
  - Generate $\sigma_{\mathbf{b}} = \mathsf{Sign}_{ot}(\mathsf{sk}_{\mathbf{b}}, (\mathsf{pk}_{(\mathbf{b},0)}, \mathsf{pk}_{(\mathbf{b},1)}))$
  - Associate $\mathbf{b}$ with the tuple $(\mathsf{sk}_{(\mathbf{b},0)}, \mathsf{sk}_{(\mathbf{b},1)}, \mathsf{pk}_{(\mathbf{b},0)}, \mathsf{pk}_{(\mathbf{b},1)}, \sigma_{\mathbf{b}})$.

**Game 3.** In this game, we abort if we ever have $\mathbf{a}^j = \mathbf{a}^{j'}$ for $j' \neq j$. There are a total of $q$ different $\mathbf{a}^j$ vectors, and they are drawn form a set of size $2^d$. Therefore, the probability of abort is at most $q^2/2^{d+1}$, which is negligible. Therefore, $A$ still wins with probability negligibly close to $\epsilon$.

Notice that in Game 3, since all of the $\mathbf{a}^j$ are distinct, we are only using any particular $\mathsf{sk}_{\mathbf{a}}$ key at most once. The adversary produces $q+1$ distinct $(m_\ell, \Sigma_\ell, \sigma_\ell)$ pairs. There are two distinct possibilities:

- One or more of the $\Sigma_\ell$ is outside all of the $\Sigma_{\mathbf{a}^j}$. In this case, one of the signatures in $\Sigma_\ell$ is a forgery for one of the public keys generated in answering the signing queries. We can construct a forger for $\mathcal{S}_{ot}$ by randomly guessing which of the public keys will be forged, plugging the given public key into that key, and randomly generating all of the other keys ourselves. Such an forger will successfully forge with probability only polynomially smaller than the probability $\Sigma_\ell$ lies outside of the $\Sigma_{\mathbf{a}^j}$. The assumption that $\mathcal{S}_{ot}$ is secure shows that this probability is therefore negligible.
- Two of the $\Sigma_\ell$ are identical. In this case, there is an $\mathbf{a}$ such that we have two forgeries relative to $\mathsf{sk}_{\mathbf{a}}$. We can similarly construct a forger for $\mathcal{S}_{ot}$ by guessing a random $\mathbf{a}$, and plugging in the given public key as $\mathsf{pk}_{\mathbf{a}}$, and generating the rest of the keys itself. Such a forger will successfully forge with probability only polynomially-smaller than the probability that two of the $\sigma_\ell$ are identical. The security of $\mathcal{S}_{ot}$ shows that this probability is also negligible.

Therefore, the probability that $A$ wins in Game 3 is negligible, meaning $\epsilon$ is negligible. Hence, $\mathcal{S}$ is secure.

□

# B Technical Proofs

## B.1 Proof of Lemma 2.1

We prove Lemma 2.1, which states that performing a partial measurement obtaining one of $k$ outcomes during a computation only decreases any output's probability by at most a factor of $k$.

**Proof**. Let $|\psi\rangle$ be the final state of $A$, and let $\left|\psi_y\right\rangle$ be the final state of $A'$ when the outcome of the partial measurement is $y$. Let $\Pr[y]$ be the probability that the partial measurement obtains $y$. It is straightforward to show that $|\psi\rangle = \sum_y \sqrt{\Pr[y]}\alpha_y\left|\psi_y\right\rangle$ for some $\alpha_y$ of unit norm. Then we have

$$\Pr[x] = |\langle x|\psi\rangle|^2 = \left|\sum_y \sqrt{\Pr[y]}\alpha_y\langle x\left|\psi_y\right\rangle\right|^2 \leq k\sum_y \Pr[y]\left|\langle x\left|\psi_y\right\rangle\right|^2 = k\Pr{}'[x]$$

□

## B.2 Proof of Lemma 2.5

Recall that we have two sets $\mathcal{X}$ and $\mathcal{Y}$, and for each $x \in \mathcal{X}$, distributions $D_x$ and $D'_x$ on $\mathcal{Y}$ such that $|D_x - D'_x| \leq \epsilon$ for all $x$. Let $O : \mathcal{X} \to \mathcal{Y}$ be a function where, for each $x$, $O(x)$ is drawn from $D_x$, and let $O'(x)$ be a function where, for each $x$, $O'(x)$ is drawn form $D'_x$. We wish to bound the distinguishing probability of the functions $O$ and $O'$.

We first suppose that each of the probabilities in each of the distributions $D_x$ and $D'_x$ are rational.

**Claim B.1.** *If each of the probabilities in $D_x$ and $D'_x$ are rational, then any quantum algorithm making $q$ quantum queries can only distinguish $O$ from $O'$ with probability $\sqrt{8C_0q^3\epsilon}$.*

Before proving this claim, we explain how it proves Lemma 2.5. Fix any quantum algorithm $A$. The distinguishing probability for any rational collection of distributions $D_x$ and $D'_x$ is bounded by $\sqrt{8C_0q^3 \max_x |D_x - D'_x|}$. But the distinguishing probability of $A$ is a continuous function of the probabilities in the distributions $D_x$ and $D'_x$, and the pairs of rational distributions are dense in the set of all pairs of distributions. Therefore, the bound of $\sqrt{8C_0q^3 \max_x |D_x - D'_x|}$ applies for all pairs of distributions.

Now we prove the claim:

**Proof**. Let $r$ be the smallest integer such that each of the probabilities in each of the distributions $D_x$ and $D'_x$ can be represented as a rational number with denominator $r$. Observe that we can take $\epsilon$ to be an integer times $2/r$, say $2s/r$. Let $\mathcal{Z} = [s + r]$. Let $E$ be the uniform distribution on $[r]$ and $E'$ the uniform distribution on $[r] + s = \{s + 1, ..., s + r\}$. The probabilities in $E$ and $E'$ are the same on $[s] + (r - s) = \{s, ..., r\}$, and are $1/r$ on$[s]$ and $[s] + r$ respectively. Therefore $|E - E'| = 2s/r = \epsilon$. We now construct functions $f_x$ such that if $z \leftarrow E$, $f_x(y)$ is distributed according to $D_x$ and if $z \leftarrow E'$, $f_x(y)$ is distributed according to $D'_x$. For each $y \in \mathcal{Y}$, let $p/r$ be the probability under $D_x$ and $p'/r$ the probability under $D'_x$. Suppose $p \leq p'$. Then we will choose $p$ elements of $[s] + (r - s)$ that have not been chosen before, and let $f_x$ evaluate to $y$ on those elements. We will also choose $p' - p$ elements of $[s] + r$ and let $f_x$ be $y$ on those elements as well. We treat the $p' < p$ case similarly. Then $f_x$ evaluates to $y$ with the desired probabilities, so it remains to show that we never run out elements. Since $|D_x - D'_x| \leq 2s/r$, we will never run out of elements in $[s] + r$ or $[s]$. If $|D_x - D'_x| < 2s/r$, we will run out of elements in $[s] + (r - s)$. When we run out, however, instead of picking an element in $[s] + (r - s)$, we can pick two elements, one in each of $[s]$ and $[s] + r$, and still have the correct probability.

Now that we can generate $D_x = f_x \circ E$ and $D'_x = f_x \circ E'$, we can generate $O$ and $O'$ differently. Let $P$ be the set of oracles from $\mathcal{X}$ to $\mathcal{Z}$ where each output is drawn according to $E$, and let $P'$ be the set of oracles where each output is drawn from $E'$. Then letting $O(x) = f_x(P(x))$ and

$O'(x) = f_x(P'(x))$ gives the correct distributions for $O$ and $O'$. Suppose $A$ distinguishes $O$ from $O'$ with probability $\sigma$. Then we can easily construct an algorithm $B$ that distinguishes $P$ and $P'$ with probability $\sigma$.

Let $\ell$ be some integer to be chosen later. We replace $P$ and $P'$ with small-range distributions on $\ell$ samples of $E$ and $E'$ respectively. Applying Lemma 2.4 twice shows that $B$ must still distinguish $P$ and $P'$ with probability at least $\sigma - 2C_0q^3/\ell$. But now the difference between the distribution $P$ and $P'$ is only $\ell$ samples of either $E$ or $E'$, so the distinguishing probability is at most $\ell\epsilon$. Thus $\sigma \leq \ell\epsilon + 2C_0q_4/\ell$ for any $\ell$. Setting $\ell = \sqrt{2C_0q^3/\epsilon}$ minimizes this quantity, yielding $\sqrt{8C_0q^3\epsilon}$ as desired. □

### B.3 Proof of Lemma 2.6

Recall that we have sets $\mathcal{X}$ and $\mathcal{Y}$, and distributions $D_x$ on $\mathcal{Y}$ for each $x \in \mathcal{X}$. Let $H$ be a function from $\mathcal{X}$ to $\mathcal{Y}$ where, for each $x$, $H(x)$ is drawn independently according to $D_x$. Let $H_\infty$ be the minimum over all $x \in \mathcal{X}$ of the distributions $D_x$. Let $A$ be a quantum algorithm making $q$ queries to $H$. We wish to show that $A$ can only produce $q+1$ distinct input/output pairs with probability $(q+1)/\left\lfloor 2^{H_\infty} \right\rfloor$

We proceed by converting an algorithm violating Lemma 2.6 to an algorithm violating the following lemma proved by Boneh and Zhandry [BZ13]

**Lemma B.2** ([BZ13])**.** *Fix sets $\mathcal{X}$ and $\mathcal{Y}$, and let $H$ be a random function from $\mathcal{X}$ to $\mathcal{Y}$. Then any quantum algorithm making $q$ quantum queries can only produce $q+1$ input/output pairs with probability at most $(q+1)/|\mathcal{Y}|$.*

First, we need the following technical lemma:

**Lemma B.3.** *Fix and integer $r$. Let $D$ be a distribution of a set $\mathcal{X}$ such that $\Pr[x \leftarrow D] < 1/r$ for all $x$. Then we can construct a distribution $D'$ on injective functions from $[r]$ into $\mathcal{X}$ with the property that $\Pr[x : f \xleftarrow{R} D', i \xleftarrow{R} [s], x \leftarrow f(i)] = \Pr[x : x \xleftarrow{R} D]$ for all $y$. In other words, we can generate $x$ according $D$ by drawing a random value $i$ in $[r]$, a random injective function $f$ from $D'$, and evaluating $f(i)$.*

**Proof**. Pick an arbitrary ordering of elements in $\mathcal{X}$. Then there is a one-to-one correspondence between subsets of $\mathcal{X}$ of size $r$ and strictly monotonically increasing functions from $[r]$ to $\mathcal{X}$. Therefore, it suffices to show how do sample subsets $\mathcal{T} \subseteq \mathcal{X}$ of size $r$ such that sampling $\mathcal{T}$ and then picking a random element of $\mathcal{T}$ simulates the distribution $D$. We give the algorithm *SampleSubset*, which takes as input a set $\mathcal{X}$, a distribution $D$ on $\mathcal{X}$, and an integer $r$ such that where $\Pr[x \leftarrow D] \leq 1/r$, and samples from a distribution of subsets of size $r$ with the desired properties:

We now prove that *SampleSubsets* works as promised. We need to show that $D'$ and $D''$ are distributions. Since $p^*$ is at most the smallest probability in $D$, all the probabilities in $D'$ are non-negative. Moreover, by adding up all the probabilities in $D'$, we see that they sum to 1, so $D'$ is in fact a distribution. This means all the probabilities in $D''$ are non-negative as well. Using the fact that all elements in $\mathcal{F}$ have probability $1/r$ under $D'$, we see that the probabilities in $D''$ also sum to 1, so $D''$ is also a distribution. The fact that $D'$ is a distribution also shows that $|\mathcal{F}| \leq r$, since otherwise the probabilities would sum to greater than 1.

**Algorithm 1** *SampleSubset*$(\mathcal{X}, D, r)$

If $r = 1$, draw $x \leftarrow D$, output $\{x\}$, and exit.
Otherwise, let $p_L$ be the smallest non-zero probability in $D$.
Let $p_H$ be the largest probability in $D$.
Let $p^* \leftarrow \min(p_L, \frac{1}{r} - p_H)$.
Let $\mathcal{T}$ be the set of the $r$ elements in $\mathcal{X}$ with the smallest non-zero probabilities.
With probability $rp^*$, output $\mathcal{T}$ and exit.
Otherwise, let $D'$ be the distribution where

$$\Pr[x \leftarrow D'] = \begin{cases} \frac{\Pr[x \leftarrow D] - p^*}{1 - rp^*} & \text{if } x \in \mathcal{T} \\ \frac{\Pr[x \leftarrow D]}{1 - rp^*} & \text{otherwise} \end{cases}$$

Let $\mathcal{F}$ be the set of $x$ such that $\Pr[x \leftarrow D'] = \frac{1}{r}$.
If $|\mathcal{F}| = r$, then output $\mathcal{F}$ and exit.
Otherwise, let $D''$ be the distribution where

$$\Pr[x \leftarrow D''] = \begin{cases} 0 & \text{if } x \in \mathcal{F} \\ \frac{\Pr[x \leftarrow D']}{1 - |\mathcal{F}|/r} & \text{otherwise} \end{cases}$$

Sample $T_0$ from *SampleSubset*$(\mathcal{X}, D'', r - |\mathcal{F}|)$
Output $T_0 \cup \mathcal{F}$.

Next, we explain why the recursive call to *SampleSubset* is valid. That is, that $\Pr[x \leftarrow D''] \leq \frac{1}{r'}$ where $r' = r - |\mathcal{F}|$. For $D'$, the maximum probability is at most

$$\frac{p_H}{1 - rp^*} \leq \frac{p_H}{1 - (\frac{1}{r} - p_H)} = \frac{r}{r + (r-1)/p_H} \leq \frac{r}{r + (r-1)r} = \frac{1}{r}$$

For $D''$, the maximum probability is at most (and in fact less than) $\frac{1/r}{1 - |\mathcal{F}|/r} = \frac{1}{r - |\mathcal{F}|} = \frac{1}{r'}$, as desired. Also, *SampleSubset* is never called with $r' = 0$, since in this case we would have already outputted $\mathcal{F}$.

Now, we need to show that this sampling algorithm actually terminates. We look at two cases:

- $p^* = p_L$. Let $x_L$ be an element in $\mathcal{T}$ with $\Pr[x_L \leftarrow D] = p_L$. Observe that under $D'$ and hence $D''$, $x_L$ has probability 0.
- $p^* = \frac{1}{r} - p_H$. Let $x_H$ be an element with $\Pr[x_H \leftarrow D] = p_H$. Under $D'$, $\Pr[x_H \leftarrow D'] = \frac{1}{r}$, so $x_H$ is included in $\mathcal{F}$. Therefore, under $D''$, $x_H$ has probability 0.

This means that in each recursive call to *SampleSubset*, the number of $x$ with positive probability decreases by at least 1. Since $\mathcal{X}$ is finite, eventually, the number of $x$ with positive probability will equal $r$ (it cannot be less since all probabilities in $D$ are at most than $1/r$, meaning there are at least $r$ such elements).

It remains to be proven that our sampling algorithm gives the desired distribution. In the case $r = 1$, then we just output sets $\{x\}$ where $x \leftarrow D$, which is correct. Otherwise, with probability $rp^*$, we output $\mathcal{T}$. In this case, drawing a random value from $\mathcal{T}$ gives us each element with probability $p^*$.

Since $p^*$ is at most $p_L$, we have not over-sampled any element. If we do not output $\mathcal{T}$, we then need to sample subsets to match the distribution $D'$. If any $x$ has $\Pr[x \leftarrow D'] = \frac{1}{r}$, then $x$ must be in every subset, so we set it aside in the set $\mathcal{F}$. We then need to draw $r' = r - |\mathcal{F}|$ additional elements not in $\mathcal{F}$ to match the correct distribution. It is straightforward to show that this is achieved by calling $SampleSubset(\mathcal{X}, D'', r - |\mathcal{F}|)$.

□

We are now ready to prove Lemma 2.6:

**Proof**. Recall that we have an algorithm $A$ making $q$ queries to a random oracle $H$ where outputs are drawn from distributions $D_x$ and produces $q+1$ input/output pairs with probability $\epsilon$. Additionally, for all $x \in \mathcal{X}$, the min-entropy of $D_x$ is at least $H_\infty$.

We now generate $H$ in a different way: for each $x$, we know that $D_x$ has min-entropy at least $H_\infty$. This means that the most probable element in $D_x$ has probability at most $1/2^{H_\infty} \leq 1/\lfloor 2^{H_\infty} \rfloor$. Let $r = \lfloor 2^{H_\infty} \rfloor$. Lemma B.3 shows that there is a distribution $D'_x$ on injective functions from $[r]$ to $\mathcal{Y}$ such that sampling from $D_x$ is equivalent to sampling a random $i \xleftarrow{R} [r]$, sampling a random $f \xleftarrow{R} D'_x$, and outputting $f(i)$. Therefore, if we let $F$ be a random oracle from $\mathcal{X}$ to $[r]$, and $G$ an oracle mapping each $x \in \mathcal{X}$ to a function sampled from $D'_x$, the oracle that on input $x$ computes $f = G(x)$ and outputs $y = f(x)$ is distributed identically to $H$.

We can now construct an algorithm $B$ that violates Lemma B.2. $B$ can make $q$ quantum queries to a random function $F$ from $\mathcal{X}$ into $[r]$. $B$ first builds the function $G$, and then simulates $A$, answering $A$'s queries to $H$ using $F$ and $G$ as above. Answering $A$'s queries is potentially problematic since extra information is computed — the outputs of $F$ and $G$. In order for $B$ to properly answer $A$'s queries without becoming entangled, $B$ must uncompute these extra values. Since $B$ knows $G$, it can uncompute $G$ easily. Uncomputing $F$ would normally require making a second query to $F$, but this is unacceptable since then $B$ would make $2q$ queries instead of $q$. However, the function $f$ outputted by $G$ is injective, meaning we can invert it, which allows us to uncompute the output of $F$ by applying $f^{-1}$ to the output of $H$. Therefore, each query $A$ makes requires only a single query to $F$.

With probability $\epsilon$, $A$ produces $q+1$ distinct input/output pairs $(x_i, y_i)$ for $H$. $B$ then computes the functions $f_i = G(x_i)$, and outputs the pairs $(x_i, f_i^{-1}(y_i))$. These pairs will all be distinct and valid input/output pairs of $F$. Since $F$ is a random oracle, Lemma B.2 shows that $\epsilon < (q+1)/r$. Since $r = \lfloor 2^{H_\infty} \rfloor$, this completes the proof.

□