How to Construct Quantum Random Functions

Mark Zhandry
Stanford University, USA
mzhandry@stanford.edu

Abstract

In the presence of a quantum adversary, there are two possible definitions of security for a pseudorandom function. The first, which we call standard-security, allows the adversary to be quantum, but requires queries to the function to be classical. The second, quantum-security, allows the adversary to query the function on a quantum superposition of inputs, thereby giving the adversary a superposition of the values of the function at many inputs at once. Existing proof techniques for proving the security of pseudorandom functions fail when the adversary can make quantum queries. We give the first quantum-security proofs for pseudorandom functions by showing that some classical constructions of pseudorandom functions are quantum-secure. Namely, we show that the standard constructions of pseudorandom functions from pseudorandom generators or pseudorandom synthesizers are secure, even when the adversary can make quantum queries. We also show that a direct construction from lattices is quantum-secure. To prove security, we develop new new tools to prove the indistinguishability of distributions under quantum queries.

In light of these positive results, one might hope that all standard-secure pseudorandom functions are quantum-secure. To the contrary, we show a separation — there exist pseudorandom functions secure against adversaries with only classical access to the function, but insecure once the adversary can make quantum queries.

Keywords: Quantum, Pseudorandom Function

1 Introduction

In their seminal paper, Goldreich, Goldwasser, and Micali [GGM86] define what it means for a function to be pseudorandom, and construct pseudorandom functions (PRFs) from any pseudorandom generator. Since then, pseudorandom functions have also been built from pseudorandom synthesizers [NR95], as well as directly from hard problems [NR97, NRR00, DY05, LW09, BMR10, BPR11]. Pseudorandom functions have become an important tool in cryptography: for example, they are used in the construction of block ciphers and message authentication codes.

To define pseudorandom functions in the presence of a quantum adversary, two approaches are possible. The first is what we call standard-security: the quantum adversary can only make *classical* queries to the function, but all the computation between the queries may be quantum. The second, which we call quantum-security, allows the adversary to make *quantum* queries to the function. We call pseudorandom functions that are secure against quantum queries Quantum Pseudorandom Functions, or QPRFs. Constructing secure QPRFs will be the focus of this paper.

Quantum-secure pseudorandom functions (QPRFs) have several applications. Whenever a pseudorandom function is used in the presence of a quantum adversary, security against quantum queries captures a wider class of attacks. Thus, the conservative approach to crpytosystem design would dictate using a quantum-secure pseudorandom function. Further, in any instance where a pseudorandom function might be evaluated on a superposition, quantum-security is required. Lastly, quantum-secure pseudorandom functions can be used to simulate quantum-accessible random oracles [BDF⁺11]. Unlike the classical setting, where a random oracle can be simulated on the fly, simulating a quantum-accessible random oracle is simulated by defining the entire function up front before any queries are made. Zhandry [Zha12] observes that if the number of queries is *a-priori* bounded by q, $2q$-wise independent functions are sufficient. However, whenever the number of quantum queries is not known in advance, quantum-secure pseudorandom functions seem necessary for simulating quantum-accessible random oracles.

1.1 Proving Quantum Security

One might hope that a proof of standard-security would imply a proof of quantum-security. However, all existing proofs of security for pseudorandom functions are inherently classical in nature and do not immediately imply quantum-security. For example, we consider the Goldreich et al. construction of a pseudorandom function PRF from a pseudorandom generator G:

At a high level, implicit in the definition of PRF is a binary tree of depth $n+1$, where each leaf corresponds to an input/output pair of PRF. To evaluate PRF, we start at the root, and follow the path from root to the leaf corresponding to the input. The key to proving security is that any efficient adversary can only make a polynomial number of queries, and can thus the evaluations of PRF visit only a polynomial number of nodes in the tree. This allows any adversary A which breaks the security of PRF with non-negligible probability to be converted into an adversary breaking the security of G also with non-negligible probability.

In the quantum setting, A may query on a superposition of all inputs, so the response could "visit" all exponentially many nodes in the tree. Therefore, the argument above no longer applies, and there is no obvious way to adapt it to the quantum setting. All existing security proofs for pseudorandom functions from standard assumptions suffer from similar weaknesses.

1.2 Our Results

We investigate the quantum-security of pseudorandom functions. Our results are as follows:

- We show that there are standard-secure pseudorandom functions that are not quantum-secure, thus a standard-secure PRF may not be secure as a QPRF.
- We show that specific constructions of pseudorandom functions are quantum-secure:
  - The construction from length-doubling pseudorandom generators (PRGs) due to Goldreich, Goldwasser, and Micali [GGM86].
  - The construction from pseudorandom synthsizers due to Naor and Reingold [NR95].
  - The direct construction based on the Learning With Errors problem due to Banerjee, Peikert, and Rosen [BPR11].
- The main technical tool used in the above proofs is the following: let D_1 and D_2 be two distributions over a set $\mathcal{Y}$. Let $\mathcal{X}$ be another set, and let O_1 and O_2 be the distributions of functions from $\mathcal{X}$ to $\mathcal{Y}$ where for each $x \in \mathcal{X}$, $O_i(x)$ is chosen independently according to D_i. We show that if D_1 and D_2 are computationally (resp. statistically) indistinguishable, then the oracles O_1 and O_2 are computationally (resp. statistically) indistinguishable by any algorithm making a polynomial number of quantum queries. We this tool, for example, to argue that a stronger notion of PRG security is equivalent to the standard PRG security definition.

2 Preliminaries and Notation

We say that $\epsilon = \epsilon(n)$ is negligible if, for all polynomials $p(n)$, $\epsilon(n) < 1/p(n)$ for large enough n.

For an integer k, we will use non-standard notation and write $[k] = \{0, ..., k-1\}$ to be the set of non-negative integers less than k. We write the set of all n bit strings as $[2]^n$. Let $x = x_1...x_n$ be a string of length n. We write $x_{[a,b]}$ to denote the substring $x_a x_{a+1} ... x_b$.

2.1 Functions and Probabilities

Given two sets $\mathcal{X}$ and $\mathcal{Y}$, define $\mathcal{Y}^{\mathcal{X}}$ as the set of functions $f : \mathcal{X} \to \mathcal{Y}$. Notice that the set of functions $f : \mathcal{Z} \to \mathcal{X} \times \mathcal{Y}$ can be written both as $(\mathcal{X} \times \mathcal{Y})^{\mathcal{Z}}$ and as $\left(\mathcal{X}^{\mathcal{Z}}\right) \times \left(\mathcal{Y}^{\mathcal{Z}}\right)$. In other words, we can think of f as a pair of functions $f = (f_0, f_1)$ where $f_0 : \mathcal{Z} \to \mathcal{X}$ and $f_1 : \mathcal{Z} \to \mathcal{Y}$.

Given $f \in \mathcal{Y}^{\mathcal{X}}$ and $g \in \mathcal{Z}^{\mathcal{Y}}$, let $g \circ f$ be the composition of f and g. That is, $g \circ f(x) = g(f(x))$. If $\mathcal{F} \subseteq \mathcal{Y}^{\mathcal{X}}$, let $g \circ \mathcal{F}$ be the set of functions $g \circ f$ for $f \in \mathcal{F}$. Similarly, if $\mathcal{G} \subseteq \mathcal{Z}^{\mathcal{Y}}$, $\mathcal{G} \circ f$ is the set of functions $f \circ g$ where $g \in \mathcal{G}$. Define $\mathcal{G} \circ \mathcal{F}$ accordingly.

We define a weight assignment on a set $\mathcal{X}$ as a function $D : \mathcal{X} \to \mathbb{R}$ such that $\sum_{x \in \mathcal{X}} D(x) = 1$. For some event `event`, we write $\Pr_{x \leftarrow D}[\texttt{event}]$ to represent the sum of the weights of all x consistent with that event. A distribution on $\mathcal{X}$ is a weight-assignment D such that $D(x) \geq 0$ for all $x \in \mathcal{X}$. If D is a distribution, we say that x occurs with probability $D(x)$. We will sometimes abuse notation and write $\mathcal{X}$ to denote the uniform distribution on $\mathcal{X}$.

Given a weight assignment D on $\mathcal{Y}^{\mathcal{X}}$ and a function $g \in \mathcal{Z}^{\mathcal{Y}}$, define the weight assignment $g \circ D$ over $\mathcal{Z}^{\mathcal{X}}$ where the weight of a function h is the sum of $D(f)$ for all f where $h = g \circ f$. Given $f \in \mathcal{Y}^{\mathcal{X}}$ and a weight assignment E over $\mathcal{Z}^{\mathcal{X}}$, define $E \circ f$ and $E \circ D$ accordingly.

Given a weight assignment D on $\mathcal{Y}^{\mathcal{X}}$, and a subset $\mathcal{W}$ of $\mathcal{X}$, we define the marginal weight assignment $D_{\mathcal{W}}$ on $\mathcal{Y}^{\mathcal{W}}$ where the weight of each function $f : \mathcal{W} \to \mathcal{Y}$ is the sum of all weights of functions $f' : \mathcal{X} \to \mathcal{Y}$ consistent with f on $\mathcal{W}$. In other words,

$$D_{\mathcal{W}}(f) = \Pr_{f' \leftarrow D}[f(w) = f'(w) \forall w \in \mathcal{W}]$$

We say that D is k-wise equivalent to another distribution D' if, for all subsets $\mathcal{W}$ of size k, $D_{\mathcal{W}} = D'_{\mathcal{W}}$. We write this as $D \stackrel{k}{=} D'$. We say that D is k-wise independent if it is k-wise equivalent to the uniform distribution.

Given a weight assignment D on a set $\mathcal{Y}$, and another set $\mathcal{X}$, define $D^{\mathcal{X}}$ as the weight assignment on $\mathcal{Y}^{\mathcal{X}}$ where $D^{\mathcal{X}}(f) = \prod_{x \in \mathcal{X}} D(f(x))$. When D is a distribution, this corresponds to the distribution over $\mathcal{Y}^{\mathcal{X}}$ where the output for each input is chosen independently according to D.

The distance between two weight assignments D_1 and D_2 over a set $\mathcal{X}$ is

$$|D_1 - D_2| = \sum_{x \in \mathcal{X}} |D_1(x) - D_2(x)|$$

If $|D_1 - D_2| \leq \epsilon$, we say D_1 and D_2 are ϵ-close. If $|D_1 - D_2| \geq \epsilon$, we say they are ϵ-far.

2.2 Quantum Computation

Here we state some basic facts about quantum computation needed for the paper, and refer the reader to Nielsen and Chuang [NC00] for a more in depth discussion.

Fact 1. *Any classical efficiently computable function f can be implemented efficiently by a quantum computer. Moreover, f can be implemented as an oracle which can be queried on quantum superpositions.*

The following is a result from Zhandry [Zha12]:

Fact 2. *For any sets $\mathcal{X}$ and $\mathcal{Y}$, we can efficiently "construct" a random oracle from $\mathcal{X}$ to $\mathcal{Y}$ capable of handling q quantum queries, where q is a polynomial. More specifically, the behavior of any quantum algorithm making at most q queries to a $2q$-wise independent function is identical to its behavior when the queries are made to a random function.*

Given an efficiently sampleable distribution D over a set $\mathcal{Y}$, we can also "construct" a random function drawn from $D^{\mathcal{X}}$ as follows: Let $\mathcal{Z}$ be the set of randomness used to sample from D, and let $f(r)$ be the element $y \in \mathcal{Y}$ obtained using randomness $r \in \mathcal{Z}$. Then $D^{\mathcal{X}} = f \circ \mathcal{Z}^{\mathcal{X}}$, so we first construct a random function $O' \in \mathcal{Z}^{\mathcal{X}}$, and let $O(x) = f(O'(x))$.

2.3 Cryptographic Primitives

In this paper, we always assume the adversary is a quantum computer. However, for any particular primitive, there may be multiple definitions of security, based on how the adversary is allowed to interact with the primitive. Here we define pseudorandom functions and two security notions. The definitions of pseudorandom generators and synthesizers appears in the relevant sections.

Definition 2.1 (PRF). *A pseudorandom function is a function $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, where $\mathcal{K}$ is the key-space, and $\mathcal{X}$ and $\mathcal{Y}$ are the domain and range. $\mathcal{K}$, $\mathcal{X}$, and $\mathcal{Y}$ are implicitly functions of the security parameter n. We write $y = \mathsf{PRF}_k(x)$.*

Definition 2.2 (Standard-Security)**.** *A pseudorandom function* PRF *is standard-secure if no efficient quantum adversary* A *making* classical *queries can distinguish between* PRF_k *for a random* k *from a truly random function. That is, for every such* A*, there exists a negligible function* $\epsilon = \epsilon(n)$ *such that*

$$\left| \Pr_{k \leftarrow \mathcal{K}}[A^{\mathsf{PRF}_k}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}}[A^{O}() = 1] \right| < \epsilon$$

Definition 2.3 (Quantum-Security)**.** *A pseudorandom function* PRF *is quantum-secure if no efficient quantum adversary* A *making* quantum *queries can distinguish* PRF_k *from a random* k *from truly a random function.*

We call such quantum-secure pseudorandom functions Quantum Random Functions, or QPRFs.

3 Separation Result

In this section, we show our separation result:

Theorem 3.1. *If secure PRFs exist, then there are standard-secure PRFs that are not QPRFs.*

Proof. Let PRF be a standard-secure pseudorandom function with key-space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$. Interpret $\mathcal{X}$ as $[N]$, where N is the number of elements in $\mathcal{X}$. We can assume without loss of generality that $\mathcal{Y}$ contains at least N^2 elements (if not, we can construct a new pseudorandom function with smaller domain but larger range in a standard way).

We now construct a new pseudorandom function $\mathsf{PRF}'_{(k,a)}(x) = \mathsf{PRF}_k(x \mod a)$ where:

- The key space of PRF' is $\mathcal{K}' = \mathcal{K} \times \mathcal{A}$ where $\mathcal{A} = \mathbb{Z} \cap (N/2, N]$. That is, a key for PRF' is a pair (k, a) where k is a key for PRF, and a is an integer in the range $(N/2, N]$.
- The domain is $\mathcal{X}' = [N']$ where N' is the smallest power of 2 greater than $4N^2$.

The following two claims are proved in Appendix A:

Claim 1. *If* PRF *is standard-secure, then so is* PRF'

Sketch of Proof. The basic idea is that an algorithm making a polynomial number of classical queries to PRF' has negligible probability of querying on two points x and x' such that $x \equiv x' \mod a$. As long as no such pair of points are queried, PRF' will still look like a random function. □

Claim 2. *If* PRF *is quantum-secure, then* PRF' *is not.*

Sketch of Proof. If we allow quantum queries to PRF', we can use the period finding algorithm of Boneh and Lipton [BL95] to find a. With a, it is easy to distinguish PRF' from a random oracle. □

Thus one of PRF and PRF' is standard-secure but not quantum-secure, as desired. □

We have shown that for pseudorandom functions, security against classical queries does not imply security against quantum queries. In the next sections, we will show, however, that several of the standard constructions in the literature are nevertheless quantum-secure.

4 Distinguishing Oracle Distributions

In this section, we describe some tools for arguing that a quantum algorithm cannot distinguish between two oracle distributions. Let $\mathcal{X}$ and $\mathcal{Y}$ be sets. We start by recalling some definitions and theorems from Zhandry [Zha12]:

Definition 4.1. *Let D_1 and D_2 be weight assignments for $\mathcal{Y}^{\mathcal{X}}$. Define the pseudometric $|D_1 - D_2|_{(k)}$ as the minimum, over weight assignments D_1' that are k-wise equivalent to D_1, of the quantity $|D_1' - D_2|$. That is,*

$$|D_1 - D_2|_{(k)} = \min_{D_1' \overset{k}{=} D_1} |D_1' - D_2|$$

Zhandry shows that this is in fact a pseudometric, and establishes the following two theorems:

Theorem 4.2. *Let A be a quantum algorithm making q quantum queries to an oracle drawn from D_1 or D_2. Then the distributions of outputs of A under D_1 and D_2 are $|D_1 - D_2|_{(2q)}$-close.*

Theorem 4.3. *Fix k. Suppose we have a family of distributions D_λ over $\mathcal{Y}^{\mathcal{X}}$ parametrized by $\lambda \in [0, 1]$. Let $\mathcal{P}$ be the collection of probabilities for the marginal distributions over all sets of k inputs. Suppose there are integers d and Δ such that for each $p \in \mathcal{P}$:*

- *p is represented by a polynomial in λ of degree at most d.*
- *The λ^j coefficient of p is 0 for each $j \in \{1, ..., \Delta\}$.*

Then $|D_\lambda - D_0|_{(k)} < 4\zeta(2\Delta + 2)\lambda^{\Delta+1}(d - \Delta)^{2\Delta+2}$ where ζ denotes the Riemann Zeta function.

We now show a similar result:

Theorem 4.4. *Fix k. Suppose we have a family of distributions E_r over $\mathcal{Y}^{\mathcal{X}}$ parametrized by $r \in \mathbb{Z}^+ \bigcup \{\infty\}$. Let $\mathcal{P}$ be the collection of probabilities for the marginal distributions over all sets of k inputs. Suppose there are integers d and Δ such that for each $p \in \mathcal{P}$:*

- *p is represented by a polynomial in $1/r$ of degree at most d.*
- *The $(1/r)^j$ coefficient of p is 0 for each $j \in \{1, ..., \Delta\}$.*

Then $|E_r - E_\infty|_{(k)} < 2^{1-\Delta}\zeta(2\Delta + 2)(1/r)^{\Delta+1}(d - \Delta)^{3\Delta+3}$

Sketch of Proof. Let $D_\lambda = E_{1/\lambda}$. We see that the conditions of Theorems 4.4 and 4.3 are identical, with the following exception: Theorem 4.3 requires D_λ to be a distribution for all $\lambda \in [0, 1]$, while Theorem 4.4 only requires D_λ to be a distribution when $1/\lambda$ is an integer (or $\lambda = 0$). The proof is thus similar in flavor to that of Theorem 4.3, except that we need to cope with the relaxed assumptions. The proof is in Appendix B. $\square$

In the next section, we apply Theorem 4.4 to a new class of distributions.

4.1 Small-Range Distributions

We now apply Theorem 4.4 to a new distribution on oracles, which we call small-range distributions. Given a distribution D on $\mathcal{Y}$, define SR_r^D as follows:

- For each $i \in [r]$, chose a random value $y_i \in \mathcal{Y}$ according to the distribution D.
- For each $x \in \mathcal{X}$, pick a random $i \in [r]$ and set $O(x) = y_i$.

The following is proved in Appendix C:

Lemma 4.5. *Fix k. The probabilities in each of the marginal distributions of SR_r^D over k inputs are polynomials in $1/r$ of degree k.*

An alternate view of this function is to choose a function $g \leftarrow D^{[r]}$ and a function $f \leftarrow [r]^{\mathcal{X}}$, and the output function is the composition $g \circ f$. That is, $\mathsf{SR}_r^D = D^{[r]} \circ [r]^{\mathcal{X}}$. Notice that, as r goes to infinity, f will be injective with probability 1, and hence for each x, $g(f(x))$ will be distributed independently according to D. That is, $\mathsf{SR}_\infty^D = D^{\mathcal{X}}$. We can then use Theorem 4.4 with $\Delta = 0$ (and that $\zeta(2) = \pi^2/6$) to bound the ability of any quantum algorithm to distinguish SR_r^D from SR_∞^D:

Corollary 4.6. *The output distributions of a quantum algorithm making q quantum queries to an oracle either drawn from SR_r^D or $D^{\mathcal{X}}$ are $\ell(q)/r$-close, where $\ell(q) = 2(\pi^2/6)(2q)^3 < 27q^3$.*

We observe that this bound is tight: in Appendix D we show that the quantum collision finding algorithm of Brassard, Høyer, and Tapp [BHT97] can be used to distinguish SR_r^D from $D^{\mathcal{X}}$ with optimal probability. This shows that Theorem 4.4 is tight for $\Delta = 0$.

4.2 Oracle-Indistinguishability

We now use the above techniques to prove our main tool for the rest of the paper. Let D_1 and D_2 be two distributions over a set $\mathcal{Y}$. Recall that D_1 and D_2 are computationally indistinguishable if, for all efficient (quantum) algorithms A,

$$\left| \Pr_{y \leftarrow D_1}[A(y) = 1] - \Pr_{y \leftarrow D_2}[A(y) = 1] \right| < \epsilon$$

where ϵ is negligible. We now consider a new notion, which we call oracle-indistinguishability:

Definition 4.7 (Oracle-Indistinguishable)**.** *Two distributions D_1 and D_2 over a set $\mathcal{Y}$ are oracle-indistinguishable if, for all sets $\mathcal{X}$, no efficient quantum algorithm A can distinguish $D_1^{\mathcal{X}}$ from $D_2^{\mathcal{X}}$. That is, for all such A and $\mathcal{X}$, there is a negligible function ϵ such that*

$$\left| \Pr_{O \leftarrow D_1^{\mathcal{X}}}[B^O() = 1] - \Pr_{O \leftarrow D_2^{\mathcal{X}}}[B^O() = 1] \right| < \epsilon$$

We now explore the relationship between standard- and oracle-indistinguishability. Clearly, oracle- implies standard-indistinguishability: if A breaks the standard-indistinguishability of D_1, and D_2, then $B^O() = A(O(x))$ for any $x \in \mathcal{X}$ breaks the oracle-indistinguishability. In the other direction, in the classical world, we can do a simple hybrid argument across the q oracle inputs any adversary B makes queries to, resulting in an algorithm that breaks the standard indistinuishability. However, in the quantum world, each query might be over a superposition of exponentially many inputs. Therefore hybrid will be over exponentially many inputs, so the proof fails.

In the statistical setting, this question has been answered by Boneh et al. [BDF+11]. They show that if a (potentially unbounded) quantum adversary making q queries distinguishes $D_1^{\mathcal{X}}$ from $D_2^{\mathcal{X}}$ with probability ϵ, then D_1 and D_2 must $\Omega(\epsilon^2/q^4)$-far.

Now, we extend their result to the computational setting:

Theorem 4.8. *Let D_1 and D_2 be efficiently sampleable distributions over a set $\mathcal{Y}$. Then D_1 and D_2 are quantum indistinguishable if and only if they are also quantum oracle-indistinguishable.*

Proof. Let B be a quantum adversary that distinguishes $D_1^{\mathcal{X}}$ from $D_2^{\mathcal{X}}$ with probability ϵ, for distributions D_1 and D_2 over $\mathcal{Y}$. That is, there is some set $\mathcal{X}$ such that

$$\left|\Pr_{O\leftarrow D_1^{\mathcal{X}}}[B^O()=1]-\Pr_{O\leftarrow D_2^{\mathcal{X}}}[B^O()=1]\right|=\epsilon$$

Choose r so that $\ell(q)/r=\epsilon/4$, where $\ell(q)$ is the polynomial from Corollary 4.6. That is, $r=4\ell(q)/\epsilon$. No quantum algorithm can distinguish $\mathsf{SR}_r^{D_i}$ from $D_i^{\mathcal{X}}$ with probability greater than $\ell(q)/r=\epsilon/4$. Thus, it must be that

$$\left|\Pr_{O\leftarrow \mathsf{SR}_r^{D_1}}[B^O()=1]-\Pr_{O\leftarrow \mathsf{SR}_r^{D_2}}[B^O()=1]\right|\geq\epsilon/2$$

We now define $r+1$ hybrids H_i as follows: For $j=0,...,i-1$, draw y_j from D_1. For $j=i,...,r-1$, draw y_j from D_2. Then give B the oracle O where for each x, $O(x)$ is a randomly selected y_i. H_r is the case where $O\leftarrow\mathsf{SR}_r^{D_1}$, and H_0 is the case where $O\leftarrow\mathsf{SR}_r^{D_2}$. Hence H_0 and H_r are distinguished with probability at least $\epsilon/2$. Let

$$\epsilon_i=\Pr_{O\leftarrow H_{i+1}}[B^O()=1]-\Pr_{O\leftarrow H_i}[B^O()=1]$$

be the probability that B distinguishes H_{i+1} from H_i. Then $|\sum_{i=1}^r\epsilon_i|\geq\epsilon/2$.

We construct an algorithm A that distinguishes between D_1 and D_2 with probability $\epsilon/2r$. A, on inputs y, does the following:

- Choose a random $i\in[r]$.
- Construct a random oracle $O_0\leftarrow[r]^{\mathcal{X}}$.
- Construct random oracles $O_1\leftarrow D_1^{\{0,...,i-1\}}$ and $O_2\leftarrow D_2^{\{i+1,...,r-1\}}$.
- construct the oracle O where $O(x)$ is defined as follows:
  - Compute $j=O_0(x)$.
  - If $j=i$, output y.
  - Otherwise, if $j<i$, output $O_1(j)$ and if $j>i$, output $O_2(j)$.
- Simulate B with the oracle O, and output the output of B.

If $y\leftarrow D_1$, B sees hybrid H_{i+1}. If $y\leftarrow D_2$, B sees H_i. Therefore, we have that for fixed i, $\Pr_{y\leftarrow D_1}[A(y)=1]-\Pr_{y\leftarrow D_2}[A(y)=1]=\epsilon_i$. Averaging over all i,

$$\left|\Pr_{y\leftarrow D_1}[A(y)=1]-\Pr_{y\leftarrow D_2}[A(y)=1]\right|=\left|\frac{1}{r}\sum_{i=1}^r\epsilon_i\right|\geq\frac{\epsilon}{2r}=\frac{\epsilon^2}{8\ell(q)}$$

Thus, A takes about the same time as B, and distinguishes D_1 from D_2 with a probability polynomial in the probability B distinguishes $D_1^{\mathcal{X}}$ from $D_2^{\mathcal{X}}$. If B breaks the oracle-indistinguishability of $D_1^{\mathcal{X}}$ and $D_2^{\mathcal{X}}$, then B is efficient and distinguishes the two with non-negligible probability. Hence, A is also efficient and distinguishes D_1 from D_2 with non-negligible probability. $\square$

Notice that this proof works in the statistical setting as well, so that if any quantum algorithm making q quantum queries distinguishes $D_1^{\mathcal{X}}$ from $D_2^{\mathcal{X}}$ with probability ϵ, then D_1 and D_2 must be $\Omega(\epsilon^2/\ell(q)) = \Omega(\epsilon^2/q^3)$-far, improving the result of Boneh et al. by a factor of q.

5 Pseudorandom Functions from Pseudorandom Generators

We give the construction of pseudorandom functions from pseudorandom generators due to Goldreich, Goldwasser, and Micali [GGM86]. Using the techniques of the previous section, we prove its security in a new way that makes sense in the quantum setting. First, we define pseudorandom generators:

Definition 5.1 (PRG)**.** *A pseudorandom generator (PRG) is a function $G : \mathcal{X} \to \mathcal{Y}$. $\mathcal{X}$ and $\mathcal{Y}$ are implicitly indexed by the security parameter n.*

Definition 5.2 (Standard-Security)**.** *A pseudorandom function G is standard-secure if the distributions $G \circ \mathcal{X}$ and $\mathcal{Y}$ are indistinguishable.*

Construction 1 (GGM-PRF)**.** *Let $G : \mathcal{K} \to \mathcal{K}^2$ be a length-doubling pseudorandom generator. Write $G(x) = (G_0(x), G_1(x))$ where G_0, G_1 are functions from $\mathcal{K}$ to $\mathcal{K}$. Then we define the GGM pseudorandom function $\mathsf{PRF} : \mathcal{K} \times [2]^n \to \mathcal{K}$ where*

$$\mathsf{PRF}_k(x) = G_{x_1}(...G_{x_{n-1}}(G_{x_n}(k))...)$$

As described in the introduction, the standard proof of security fails to prove quantum-security. With the techniques from Section 4, we show how to work around this problem. We first define a stronger notion of security for pseudorandom generators, which we call oracle-security:

Definition 5.3 (Oracle-Security)**.** *A pseudorandom generator $G : \mathcal{X} \to \mathcal{Y}$ is oracle-secure if the distributions $G \circ \mathcal{X}$ and $\mathcal{Y}$ are oracle-indistinguishable.*

Since both $G \circ \mathcal{X}$ and $\mathcal{Y}$ are efficiently sampleable, Theorem 4.8 gives us:

Corollary 5.4. *If G is a secure PRG, then it is also oracle-secure.*

We now can prove the security of Construction 1.

Theorem 5.5. *If G is a standard-secure PRG, then* PRF *from Construction 1 is a QPRF.*

Proof. We adapt the security proof of Goldreich et al. to convert any adversary for PRF into an adversary for the oracle-security of G. Then Corollary 5.4 shows that this adversary is impossible under the assumption that G is standard-secure.

Suppose a quantum adversary A distinguishes PRF from a random oracle with probability ϵ. Define hybrids H_i as follows: Pick a random function $P \leftarrow \mathcal{K}^{[2]^{n-i}}$ and give A the oracle

$$O_i(x) = G_{x_1}(...G_{x_i}(P(x_{[i+1,n]}))...).$$

H_0 is the case where A's oracle is random. When $i = n$, $P \leftarrow \mathcal{K}^{[2]^{n-i}}$ is a random function from the set containing only the empty string to $\mathcal{K}$, and hence is associated with the image of the empty string, a random element in $\mathcal{K}$. Thus H_n is the case where A's oracle is PRF. A simple hybrid argument shows there for some i, A distinguishes O_i from O_{i+1} with probability at least ϵ/n.

We now construct a quantum algorithm B breaking the oracle-security of G. B has quantum access to an oracle $P : [2]^{n-i-1} \to \mathcal{K}^2$, and distinguishes $P \leftarrow G \circ \mathcal{K}^{[2]^{n-i-1}}$ from $P \leftarrow (\mathcal{K}^2)^{[2]^{n-i-1}}$:

- Interpret P as (P_0, P_1) where $P_b \in [2]^{n-i-1} \to \mathcal{K}$.
- Construct the oracle $O : [2]^n \to \mathcal{K}$ where $O(x) = G_{x_1}(...G_{x_i}(P_{x_{i+1}}(x_{[i+2,n]}))...)$.
- Simulate A with oracle O, and output the output of A.

In the case where P is truly random, so are P_0 and P_1, and thus $O = O_i$, the oracle in hybrid H_i. When O is drawn from $G \circ \mathcal{K}^{[2]^{n-i-1}}$, then $P_b \leftarrow G_b \circ \mathcal{K}^{[2]^{n-i-1}}$, and we thus get that $O = O_{i+1}$, the oracle in hybrid H_{i+1}. Since A distinguishes these with probability at least ϵ/n, B breaks the oracle-security of G with the same probability. $\square$

6 Pseudorandom Functions from Synthesizers

In this section, we show that the construction of pseudorandom functions from pseudorandom synthesizers due to Naor and Reingold [NR95] is quantum-secure.

Definition 6.1 (Synthesizer)**.** *A pseudorandom synthesizer is a function $S : \mathcal{X}^2 \to \mathcal{Y}$. $\mathcal{X}$ and $\mathcal{Y}$ are implicitly indexed by the security parameter n.*

Definition 6.2 (Standard-Security)**.** *A pseudoreandom synthesizer $S : \mathcal{X}^2 \to \mathcal{Y}$ is standard-secure if, for any set $\mathcal{Z}$, no efficient quantum algorithm A making classical queries can distinguish $O(z_1, z_2) = S(O_1(z_1), O_2(z_2))$ where $O_b \leftarrow \mathcal{X}^{\mathcal{Z}}$ from $\mathcal{Y}^{\mathcal{Z}\times\mathcal{Z}}$. That is, for any such A and $\mathcal{Z}$, there exists a negligible function ϵ such that*

$$\left| \Pr_{O_1 \leftarrow \mathcal{X}^{\mathcal{Z}}, O_2 \leftarrow \mathcal{X}^{\mathcal{Z}}}[A^{S(O_1,O_2)}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{Z}\times\mathcal{Z}}}[A^{O}() = 1] \right| < \epsilon$$

Where $S(O_1, O_2)$ means the oracle that maps (z_1, z_2) into $S(O_1(z_1), O_2(z_2))$.

Construction 2 (NR-PRF)**.** *Given a pseduorandom synthesizer $S : \mathcal{X}^2 \to \mathcal{X}$, let ℓ be an integer and $n = 2^\ell$. We let $\mathsf{PRF}_k(x) = \mathsf{PRF}_k^{(\ell)}(x)$ where $\mathsf{PRF}^{(i)} : \left(\mathcal{X}^{2\times 2^i}\right) \times [2]^{2^i} \to \mathcal{X}$ is defined as*

$$\begin{aligned}
\mathsf{PRF}^{(0)}_{a_{1,0},a_{1,1}}(x) &= a_{1,x} \\
\mathsf{PRF}^{(i)}_{a_{1,0},a_{1,1},a_{2,0},a_{2,1},\ldots,a_{2^i,0},a_{2^i,1}}(x) &= S(\mathsf{PRF}^{(i-1)}_{a_{1,0},a_{1,1},\ldots,a_{2^{i-1},0},a_{2^{i-1},1}}(x_{[1,2^{i-1}]}), \\
&\qquad \mathsf{PRF}^{(i-1)}_{a_{2^{i-1}+1,0},a_{2^{i-1}+1,1},\ldots,a_{2^i,0},a_{2^i,1}}(x_{[2^{i-1}+1,2^i]}))
\end{aligned}$$

The following theorem is proved in Appendix E:

Theorem 6.3. *If S is a standard-secure synthesizer, then PRF from Construction 2 is a QPRF.*

Sketch of Proof. The proof is very similar to that of the security of the GGM construction: we define a new notion of security for synthesizers, called quantum-security, and prove that quantum-security implies that Construction 2 is quantum secure, following the techniques of Naor and Reingold. Then, we prove the equivalence of quantum-security and standard-security for synthesizers, completing the proof. $\square$

7 Direct Construction of Pseudorandom Functions

In this section, present the construction of pseudorandom functions from Banerjee, Peikert, and Rosen [BPR11]. We show that this construction is quantum-secure.

Let p, q be integers with $q > p$. Let $\lfloor x \rceil_p$ be the map from $\mathbb{Z}_q$ into $\mathbb{Z}_p$ defined by first rounding x to the nearest multiple of q/p, and then interpreting the result as an element of $\mathbb{Z}_p$. More precisely, $\lfloor x \rceil_p = \lfloor (p/q)x \rceil \mod p$ where the multiplication and division in $(p/q)x$ are computed in $\mathbb{R}$.

Construction 3. *Let p, q, m, ℓ be integers with $q > p$. Let $\mathcal{K} = \mathbb{Z}_q^{n\times m} \times (\mathbb{Z}^{n\times n})^\ell$. We define $\mathsf{PRF} : \mathcal{K} \times [2]^\ell \to \mathbb{Z}_p^{m\times n}$ as follows: For a key $k = (\mathbf{A}, \{\mathbf{S}_i\})$, let*

$$\mathsf{PRF}_k(x) = \left\lfloor \mathbf{A}^t \prod_{i=1}^{\ell} \mathbf{S}_i^{x_i} \right\rceil_p$$

Next is an informal statement of the security of PRF, whose proof appears in Appendix F:

Theorem 7.1. *let PRF be as in Construction 3. For an appropriate chose of integers p, q, m, ℓ and distribution χ on $\mathbb{Z}$, if we draw $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$ and $S_i \leftarrow \chi^{n\times n}$ and the Learning With Errors (LWE) problem is hard for modulus q and distribution χ, then PRF is a QPRF.*

Sketch of Proof. We follow the ideas from the previous sections and define a new notion of hardness for LWE, called oracle-hard, and show its equivalence to standard hardness. We then show that if LWE is oracle-hard, PRF from Construction 3 is quantum-secure. This part is very similar to the proof of Banerjee et al., with some modifications to get it to work in the quantum setting. □

8 Conclusion

We have shown that not all pseudorandom functions secure against classical queries are also secure against quantum queries. Nevertheless, we demonstrate the security of several constructions of pseudorandom functions against quantum queries. Specifically, we show that the construction from pseudorandom generators [GGM86], the construction from pseudorandom synthesizers [NR95], and the direct construction based on the Learning With Errors problem [BPR11] are all secure against quantum algorithms making quantum queries. We accomplish these results by providing more tools for bounding the ability of a quantum algorithm to distinguish between two oracle distributions. We leave as an open problem proving the quantum security of some classical uses of pseudorandom functions. We have two specific instances in mind:

- Pseudorandom permutations (Block Ciphers) secure against quantum queries. We know how to build pseudorandom permutations from pseudorandom functions in the classical setting ([LR88, NR99]). Classically, the first step to prove security is to replace the pseudorandom functions with truly random functions, which no efficient algorithm can detect. The second step is to prove that no algorithm can distinguish this case from a truly random permutation. For this construction to be secure against quantum queries, a quantum-secure pseudorandom function is clearly needed. However, it is not clear how to transform the second step of the proof to handle quantum queries.

- Message Authentication Codes (MACs) secure against quantum queries. MACs can be built from pseudorandom functions and proven existentially unforgeable against a classical adaptive chosen message attack. If we allow the adversary to ask for an authentication on a superposition of messages, a new notion of security is required. One possible definition of security is that, after q queries, no adversary can produce $q+1$ classical valid message/tag pairs. Given a pseudorandom function secure against quantum queries, proving this form of security reduces to proving the impossibility of the following: After q quantum queries to a random oracle O, output $q+1$ input/output pairs of O with non-negligible probability.

A Proof of the Separation Result

Here we finish the counter-example from Section 3 and the proof of Theorem 3.1 by proving Claims 1 and 2. Recall that we start with a pseudorandom function PRF with key-space $\mathcal{K}$, domain $[N]$, and range $\mathcal{Y}$ where $|\mathcal{Y}| \geq N^2$. We then construct a new pseudorandom function PRF' whose keys are pairs (k,a) where $k \in \mathcal{K}$ and $a \in \mathcal{A}$ where $\mathcal{A}$ is the set of integers in $(N/2, N]$. We let N' be the smallest power of 2 greater than $4N^2$, and for $x \in [N']$, define $\mathsf{PRF}'_{(k,a)}(x) = \mathsf{PRF}_k(x \mod a)$.

Proof of Claim 1. We prove that if PRF is standard-secure, so is PRF'. Suppose we have a quantum adversary A making classical queries that distinguishes PRF' from a random function with non-negligible probability ϵ. That is,

$$\left| \Pr_{k \leftarrow \mathcal{K}, a \leftarrow \mathcal{A}}[A^{\mathsf{PRF}'_{(k,a)}}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}}[A^{O}() = 1] \right| = \epsilon$$

This is equivalent to

$$\left| \Pr_{k \leftarrow \mathcal{K}, a \leftarrow \mathcal{A}}[A^{\mathsf{PRF}_k(\cdot \mod a)}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}}[A^{O}() = 1] \right| = \epsilon$$

Consider the quantity

$$\left| \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}, a \leftarrow \mathcal{A}}[A^{O(\cdot \mod a)}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}}[A^{O}() = 1] \right|$$

The left hand side is the case where O is a random function in $\mathcal{Y}^{\mathcal{X}}$, a is a random integer in $(N/2, N]$, and we give A the oracle $O'(x) = O(x \mod a)$. As long as A never queries its oracle on two points x and x' such that $x \equiv x' \mod a$, this oracle will look random. If A makes q queries, there are $\binom{q}{2}$ possible differences between query points. Each difference is at most $8N^2$, so for large N it can only be divisible by at most 2 different moduli a. Notice that $|A| \geq (N-1)/2$. Each difference thus has a probability at most $2/|\mathcal{A}| \leq 4/(N-1)$ of being divisible by a, so the total probability of querying x and x' such that $x \equiv x' \mod a$ is at most $2q^2/(N-1)$. Thus this probability, and hence the ability of A to distinguish O' from a random oracle, is negligible.

A simple hybrid argument then shows that

$$\left| \Pr_{k \leftarrow \mathcal{K}, a \leftarrow \mathcal{A}}[A^{\mathsf{PRF}_k(\cdot \mod a)}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}, a \leftarrow \mathcal{A}}[A^{O(\cdot \mod a)}() = 1] \right| \geq \epsilon - 2q^2/(N-1)$$

Define a quantum algorithm B which distinguishes PRF from a random oracle. B has an oracle O, chooses a random integer $a \in (N/2, N]$, and simulates A with the oracle $O'(x) = O(x \mod a)$. When $O = \mathsf{PRF}_k$, we get the left side, and when O is random, we get the right side. Thus,

$$\left| \Pr_{k \leftarrow \mathcal{K}}[B^{\mathsf{PRF}_k}() = 1] - \Pr_{O \leftarrow \mathcal{Y}^{\mathcal{X}}}[B^O() = 1] \right| \geq \epsilon - 2q^2/(N-1)$$

Since N is exponential, B breaks the standard-security of PRF. □

Proof of Claim 2. We now show that PRF and PRF' cannot both be quantum-secure. Suppose PRF is quantum secure. We first consider the case where PRF' is built from a truly random function $O : [N] \to \mathcal{Y}$. That is, $\mathsf{PRF}'_a(x) = O(x \mod a)$.

Since $|\mathcal{Y}| \geq N^2$, the probability that there is a collision (x, x') where $O(x) = O(x')$ is less than 1/2. In this case, we then notice that PRF' is periodic with period a, and we can use the results of Boneh and Lipton [BL95] to find this period in polynomial time by making quantum queries to the oracle. Thus, we get a distinguisher that works as follows, given access to an oracle O':

- Use the period-finding algorithm of Boneh and Lipton to find the period a of O'.
- If $a \in (N/2, N]$, pick a random $x \in [N' - a]$, and verify that $O'(x) = O'(x + a)$. If so, output 1. Otherwise, output 0.

If $O' = O(x \mod a)$, then with probability at least 1/2, O will have no collisions, meaning we will find a with probability $1 - o(1)$. $O'(x) = O'(x + a)$ will always be true in this case, so we output 1. If O' is random, then for any x, the probability that there is any $x' \in x + (N/2, N]$ with $O'(x') = O'(x)$ is negligible, so the random oracle will fail the test with all but negligible probability. Therefore, we distinguish PRF' from random with probability at least $1/2 - o(1)$.

We now switch to the true definition of PRF'. That is, we replace the random oracle O with PRF. Since PRF is quantum-secure, this only affects the behavior of our distinguisher negligibly. Therefore, our distinguisher still distinguishes PRF' from random with probability at least $1/2 - o(1)$. □

B Proof of Theorem 4.4

Here we prove Theorem 4.4. Recall that we have a family of distributions E_r over $\mathcal{Y}^{\mathcal{X}}$ parametrized by $r \in \mathbb{Z}^+ \bigcup \{\infty\}$. Let $\mathcal{P}$ be the collection of probabilities for the marginal distributions over all sets of k inputs, for some fixed k. Suppose there are integers d and Δ such that for each $p \in \mathcal{P}$:

- p is represented by a polynomial in $1/r$ of degree at most d.
- The $(1/r)^j$ coefficient of p is 0 for each $j \in \{1, ..., \Delta\}$.

Then we need to show that $|E_r - E_\infty|_{(k)} < 2^{1-\Delta}\zeta(2\Delta+2)(1/r)^{\Delta+1}(d-\Delta)^{3\Delta+3}$

We let $D_\lambda = E_{1/\lambda}$. We follow the same proof technique as in Zhandry [Zha12]. Zhandry shows that if we have distinct λ_i for $i \in \{1, ..., d-\Delta\}$ such that each of the D_{λ_i} are distributions, then,

$$|D_\lambda - D_0|_{(k)} \leq 2 \sum_{i=1}^{d-\Delta} |a_i(\lambda)|$$

Where

$$a_i(\lambda) = \left(\frac{\lambda}{\lambda_i}\right)^{\Delta+1} \prod_{j=1, j\neq i}^{d-\Delta} \left(\frac{\lambda - \lambda_j}{\lambda_i - \lambda_j}\right)$$

Zhandry uses $\lambda_i = \left(\frac{i}{d-\Delta}\right)^2$, but these are not the reciprocals of integers, as needed for E_{1/λ_i} to be distributions. Rounding is not enough, because many of the λ_i will round to the same values. We instead use the following λ_i:

$$\lambda_i = \frac{1}{\left\lfloor \frac{(d-\Delta)^3}{2i^2} \right\rfloor}$$

Claim 3. *If* $\lambda_i = \frac{1}{\left\lfloor \frac{(d-\Delta)^3}{2i^2} \right\rfloor}$ *and* $\lambda \leq \lambda_i$ *for all* i*, then* $\sum_i |a_i(\lambda)| < 2^{-\Delta}\zeta(2\Delta+2)(d-\Delta)^{3\Delta+3}$

Before proving this claim, we note that it proves the theorem when $\lambda \leq \lambda_i$ for all i (equivalently, $\lambda \leq \lambda_1$). If $\lambda > \lambda_1$, then the bound we are trying to prove is at least

$$2^{1-\Delta}\zeta(2\Delta+2)\left(\frac{(d-\Delta)^3}{\lfloor (d-\Delta)^3/2 \rfloor}\right)^{\Delta+1} > 4\zeta(2\Delta+2) > 2$$

Thus, since any two distributions have a distance at most 2, this bound is trivially satisfied. It now remains to prove the claim:

Proof of Claim 3. First, notice that

$$\left|\frac{a_i(\lambda)}{\lambda^{\Delta+1}}\right| = \left(\frac{1}{\lambda_i}\right)^{\Delta+1} \prod_{j=1, j\neq i}^{d-\Delta} \left(\frac{|\lambda - \lambda_j|}{|\lambda_i - \lambda_j|}\right) \leq \left(\frac{1}{\lambda_i}\right)^{\Delta+1} \prod_{j=1, j\neq i}^{d-\Delta} \left(\frac{\lambda_j}{|\lambda_i - \lambda_j|}\right)$$

Now, observe that $\lambda_i \geq \frac{2i^2}{(d-\Delta)^3}$ and that

$$|\lambda_i - \lambda_j| = \lambda_i\lambda_j \left| \left\lfloor \frac{(d-\Delta)^3}{2i^2} \right\rfloor - \left\lfloor \frac{(d-\Delta)}{2j^2} \right\rfloor \right| \geq \frac{2i^2}{(d-\Delta)^3}\lambda_j \left(\left| \frac{(d-\Delta)^3}{2i^2} - \frac{(d-\Delta)^3}{2j^2} \right| - 1 \right)$$

Which can be simplified to

$$|\lambda_i - \lambda_j| \geq \lambda_j \frac{|i^2 - j^2| - \frac{2i^2j^2}{(d-\Delta)^3}}{j^2}$$

We notice that the numerator is minimized by making i and j as large as possible, which is when they are $d-\Delta$ and $d-\Delta-1$. In this case, the quantity becomes $\lambda_j\left(3 - \frac{2}{d-\Delta}\right)/j^2$, which is greater than 0 as long as $d-\Delta \geq 1$ (if $d-\Delta = 0$, then D_λ is the same distribution for all λ, so the theorem is trivial).

Thus

$$\left|\frac{a_i(\lambda)}{\lambda^{\Delta+1}}\right| \leq \left(\frac{1}{\lambda_i}\right)^{\Delta+1} \prod_{j=1, j\neq i}^{d-\Delta} \left(\frac{j^2}{|i^2 - j^2| - \frac{2i^2j^2}{(d-\Delta)^3}}\right)$$

The $(1/\lambda_i)^{\Delta+1}$ term is bounded by $\left(\frac{(d-\Delta)^3}{2i^2}\right)^{\Delta+1}$. We now bound the other term:

Claim 4. *For all integers D and i such that $i \le D$, $\alpha_{D,i} \le 2$ where*

$$\alpha_{D,i} = \prod_{j=1, j\neq i}^{D} \left(\frac{j^2}{|i^2 - j^2| - \frac{2i^2 j^2}{D^3}} \right)$$

Proof. First, rewrite $\alpha_{D,i}$ as

$$\alpha_{D,i} = \prod_{j=1,j\neq i}^{D} \frac{j^2}{|i^2 - j^2|} \prod_{j=1,j\neq i}^{D} \frac{1}{1 - \frac{2i^2j^2}{|i^2-j^2|D^3}}$$

The first term, as demonstrated by Zhandry, equals $2(D!)^2/(D-i)!(D+i)!$. For the second term, we will decompose $2i^2j^2/(i^2-j^2)$ as $i^2\left(\frac{i}{i-j} + \frac{i}{i+j} - 2\right)$. Then

$$\begin{aligned}
\prod_{\substack{j=1\\ j\neq i}}^{D} \frac{1}{1 - \frac{2i^2j^2}{|i^2-j^2|D^3}} &< \prod_{j=1,j\neq i}^{D} 1 + \frac{2i^2j^2}{|i^2-j^2|D^3} < 1 + \frac{1}{D^3} \sum_{j=1,j\neq i}^{D} \frac{2i^2j^2}{|i^2-j^2|} \\
&= 1 + \frac{i^2}{D^3}\left(\sum_{j=1}^{i-1} \left(\frac{i}{i-j} + \frac{i}{i+j} - 2 \right) + \sum_{j=i+1}^{D} \left(\frac{i}{j-i} - \frac{i}{j+i} + 2 \right) \right) \\
&= 1 + \frac{i^2}{D^3}\left(\sum_{j_1=1}^{i-1} \frac{i}{j_1} + \sum_{j_2=i+1}^{2i-1} \frac{i}{j_2} - 2(i-1) + \sum_{j_3=1}^{D-i} \frac{i}{j_3} - \sum_{j_4=2i+1}^{D+i} \frac{i}{j_4} + 2(D-i) \right) \\
&< 1 + \frac{i^2}{D^3}\left(2(D-2i+1) + i\left(\sum_{j=1}^{2i-1} \frac{1}{j} + \sum_{j=1}^{2i} \frac{1}{j} \right) \right) \\
&< 1 + \frac{i^2}{D^3}\left(2D + 2 + 2i\left(\sum_{j=1}^{2i} \frac{1}{j} - 1 \right) \right) < 1 + \frac{2i^2}{D^3}(D + 1 + i\log 2i)
\end{aligned}$$

Let $\beta_{D,i} = 2\frac{D!^2}{(D-i)!(D+i)!}\left(1 + \frac{2i^2}{D^3}(D+1+i\log 2i)\right)$. Then $\alpha_{D,i} < \beta_{D,i}$. For fixed i, the quantity $\left(1 + \frac{2i^2}{D^3}(D+1+i\log 2i)\right)$ is positive and converges to 1 for large D. Also for fixed i, $2(D!)^2/(D-i)!(D+1)!$ converges to 2. Thus, $\beta_{D,i}$ converges to 2 for for large D. We now show that $\beta_{D,i} < 2$ by showing that it is monotonically increasing in D. Indeed,

$$\begin{aligned}
\frac{\beta_{D+1,i}}{\beta_{D,i}} &= \frac{\frac{(D+1)!^2}{(D+i+1)!(D-i+1)!}\left(1 + \frac{2i^2}{(D+1)^3}(D+2+i\log 2i)\right)}{\frac{D!^2}{(D+i)!(D-i)!}\left(1 + \frac{2i^2}{(D)^3}(D+1+i\log 2i)\right)} \\
&= \frac{D^3}{(D+i+1)(D-i+1)(D+1)} \frac{(D+1)^3 + 2i^2(D+2+i\log 2i)}{D^3 + 2i^2(D+1+i\log 2i)}
\end{aligned}$$

Using some algebraic manipulation, we get

$$\frac{\beta_{D+1,i}}{\beta_{D,i}} = 1 + i^2 \frac{D^4 - 3D^3 + c_1 D^2 + c_2 D + c_3}{(D+1)(D+1+i)(D+1-i)(D^3 + 2i^2(D+1+i\log 2i))}$$

Where

$$\begin{aligned} c_1 &= 2(i^2 - 6 - 3i\log 2i) \\ c_2 &= 2(2i^2 - 4 - 3i\log 2i + i^3\log 2i) \\ c_3 &= 2(i^2-1)(1+i\log 2i) \end{aligned}$$

Now, each of the c_k is positive and increasing for large i. For each c_k, we can find the minimum with respect to i, assuming $i \geq 1$.

- $c_1'(i) = 2(2i - 3\log 2i - 3)$, which is positive when $i = 5$. $c_1''(i) = 2(2 - 3/i)$, which is positive for $i \geq 2$. Therefore, $c_1(i)$ is strictly increasing for $i \geq 5$. Testing $i = 1, 2, 3, 4, 5$ shows that the minimum occurs at $i = 5$, and $c_1(5) \geq -32$
- $c_2'(i) = 2((4i-3) + i^2 + 3(i^2-1)\log 2i)$, which is positive for $i \geq 1$. Therefore, the minimum is when $i = 1$, and $c_2(1) \geq 7$.
- c_3 is trivially non-negative for all $i \geq 1$.

The results are thus $c_1 > -32$, $c_2 > -7$, and $c_3 > 0$. Thus,

$$\frac{\beta_{D+1,i}}{\beta_{D,i}} > 1 + i^2 \frac{D^4 - 3D^3 - 32D^2 - 7D}{(D+1)(D+1+i)(D+1-i)(D^3 + 2i^2(D+1+i\log 2i))}$$

The denominator is positive (since $i \leq D$), so $\frac{\beta_{D+1,i}}{\beta_{D,i}} > 1$ if $D^4 - 3D^3 - 32D^2 - 7D > 0$. At $D = 8$, this polynomial is positive, as are the first four derivatives (the rest being 0). This means the polynomial is positive for all $D \geq 8$. Thus, we have shown that $\beta_{D,i}$ approaches 2 for large D, and for $D \geq 8$, $\beta_{D,i}$ is strictly increasing in D. Therefore, for $D \geq 8$, we have that $\alpha_{D,i} \leq \beta_{D,i} < 2$. For the case where $D < 8$, we have 28 (D, i) pairs to check. Below, we have calculated $\alpha_{D,i}$ for $1 \leq i \leq D < 8$:

	$D = 1$	2	3	4	5	6	7
$i = 1$	1.00	2.00	1.82	1.79	1.79	1.81	1.82
2	-	0.50	1.43	1.29	1.30	1.33	1.37
3	-	-	0.23	0.86	0.79	0.82	0.87
4	-	-	-	0.10	0.46	0.43	0.47
5	-	-	-	-	0.04	0.22	0.22
6	-	-	-	-	-	0.01	0.10
7	-	-	-	-	-	-	0.01

All of these are at most 2, completing the proof of the claim. □

With this proved, we can now complete the proof of Claim 3.

$$\left|\frac{a_i(\lambda)}{\lambda^{\Delta+1}}\right| \leq \left(\frac{1}{\lambda_i}\right)^{\Delta+1} \prod_{j=1, j\neq i}^{d-\Delta} \left(\frac{j^2}{|i^2 - j^2| - \frac{2i^2j^2}{(d-\Delta)^3}}\right) \leq \left(\frac{(d-\Delta)^3}{2i^2}\right)^{\Delta+1} \times 2$$

This gives

$$|a_i(\lambda)| \leq \lambda^{\Delta+1}(d-\Delta)^{3\Delta+3} 2^{-\Delta} \frac{1}{i^{2\Delta+2}}$$

Summing over all i from 1 to $d - \Delta$ gives

$$\sum_{i=1}^{d-\Delta} |a_i(\lambda)| \leq \lambda^{\Delta+1}(d-\Delta)^{3\Delta+3}2^{-\Delta}\sum_{i=1}^{d-\Delta}\frac{1}{i^{2\Delta+2}}$$

The sum on the right hand side is the truncated p series for $p = 2\Delta + 2$. This series sums to $\zeta(p)$, so the truncation is strictly less than this value. Therefore,

$$\sum_{i=1}^{d-\Delta} |a_i(\lambda)| < \lambda^{\Delta+1}(d-\Delta)^{3\Delta+3}2^{-\Delta}\zeta(2\Delta+2)$$

□

C Proof of Lemma 4.5

In this section, we prove Lemma 4.5.

Proof of Lemma 4.5. Our goal is to show that, for each of the marginal distributions over k inputs to SR_r^D, each probability is a polynomial in $1/r$ of degree at most k.

Fix some x_i and y_i for $i \in [k]$. We consider the probability that $O(x_i) = y_i$ for all $i \in [k]$. We can assume without loss of generality that the x_i are distinct. Otherwise, there are i, j such that $x_i = x_j$. If $y_i \neq y_j$, then the probability is 0 (O is not a function in this case). If $y_i = y_j$, the $O(x_j) = y_j$ condition is redundant and can be removed, reducing this to the $k-1$ case. By induction on k, the resulting probability is a polynomial of degree at most $k - 1 < k$.

Recall that $\mathsf{SR}_r^D = D^{[r]} \circ [r]^{\mathcal{X}}$ and D is a distribution on $\mathcal{Y}$. Let $O_1 \leftarrow [r]^{\mathcal{X}}$ and $O_2 \leftarrow D^{[r]}$. Let O_1' be the restriction of O_1 to $\{x_0, ..., x_{k-1}\}$. Each O_1' then occurs with probability $1/r^k$. Now,

$$\begin{aligned}
\Pr_{O\leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] &= \Pr_{O_1\leftarrow[r]^{\mathcal{X}}, O_2\leftarrow D^{[r]}}[O_2(O_1(x_i)) = y_i \forall i \in [k]] \\
&= \Pr_{O_1'\leftarrow[r]^{\{x_0,\ldots,x_{k-1}\}}, O_2\leftarrow D^{[r]}} \Pr[O_2(O_1'(x_i)) = y_i \forall i \in [k]] \\
&= \frac{1}{r^k}\sum_{O_1'}\Pr_{O_2\leftarrow D^{[r]}}[O_2(O_1'(x_i)) = y_i \forall i \in [k]]
\end{aligned}$$

We now associate with each O_1' a partition P of $[k]$ into r disjoint subsets P_j for $j \in [r]$. The elements of P_j are the indicies i such that $O_1'(x_i) = j$. Thus:

$$\Pr_{O\leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] = \frac{1}{r^k}\sum_{P=(P_j)}\Pr[O_2(j) = y_i \forall j \in [r], \forall i \in P_j]$$

Since $O_2 \leftarrow D^{[r]}$, the distribution of outputs of O_2 for each j are independent. Thus the probabilities $\Pr[O_2(j) = y_i \forall i \in P_j]$ are also independently distributed. Thus,

$$\Pr_{O\leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] = \frac{1}{r^k}\sum_{P=(P_j)}\prod_{j\in[r]}\Pr[O_2(j) = y_i \forall i \in P_j]$$

Since there are only k elements, at most k of the P_js are non-empty. Thus, we can associate to each partition P another partition Q of $[k]$ into $k_Q \leq k$ non-empty subsets, and a strictly increasing function from f_Q from $[k_Q] \to [r]$. The association is as follows: $Q_{j'} = P_{f_Q(j')}$ and $P_j = \emptyset$ if j has no pre-image under f_Q. This allows us to write:

$$\Pr_{O \leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] = \frac{1}{r^k} \sum_{Q=(Q_{j'})} \sum_{f_Q} \prod_{j' \in [k_Q]} \Pr[O_2(f_Q(j')) = y_i \forall i \in Q_{j'}]$$

We now notice that, for fixed j', if the y_i are all equal for $i \in Q_{j'}$, then since $O_2 \leftarrow D^{[r]}$, $\Pr[O_2(f_Q(j')) = y_i \forall i \in Q_{j'}] = D(y_i)$ where i is any index in $Q_{j'}$. Otherwise, $\Pr[O_2(j) = y_i \forall i \in Q_{j'}] = 0$ since O_2 needs to be a function. Thus we can write $\Pr[O_2(j) = y_i \forall i \in Q_{j'}] = D(y_i)\sigma(Q_{j'})$ where $\sigma(S)$ is 1 if y_i are all equal for $i \in S$, and 0 otherwise. Thus,

$$\Pr_{O \leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] = \frac{1}{r^k} \sum_{Q=(Q_{j'})} \sum_{f_Q} \prod_{j' \in [k_Q]} D(y_i)\sigma(Q_{j'})$$

The summand does not depend on f_Q, so let c_Q be the number of f_Q. Then we can write

$$\Pr_{O \leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]] = \frac{1}{r^k} \sum_{Q=(Q_{j'})} c_Q \prod_{j' \in [k_Q]} \Pr[O_2(j') = y_i \forall i \in Q_{j'}]$$

The Q we are summing over are independent of r, as is the product in the above expression. c_Q is equal to the number of ways of picking k_Q distinct elements of $[r]$, which is $\binom{r}{k_Q}$, and is thus polynomial of degree k_Q in r (and hence a polynomial of degree at most k). Therefore, performing the sum, $\Pr_{O \leftarrow \mathsf{SR}_r^D}[O(x_i) = y_i \forall i \in [k]]$ is a polynomial of degree at most k in r, divided by r^k. The result is a polynomial of degree at most k in $1/r$.

□

D A Quantum Distinguisher for Small-Range Distributions

In this section, we give a quantum distinguisher that distinguishes $\mathsf{SR}_r^{\mathcal{Y}}$ from a random function with probability (asymptotically) matching the bound of Corollary 4.6. Our algorithm is basically the collision finding algorithm of Brassard, Høer, and Tapp [BHT97], with a check at the end to verify that a collision is found. The algorithm has oracle access to a function O from $\mathcal{X}$ to $\mathcal{Y}$, which is either $\mathsf{SR}_r^{\mathcal{Y}}$ or a random function. It is given as input the integer r, the number of queries q, and operates as follows:

- Let $p = (q-1)/2$. Pick a set S of p points in $\mathcal{X}$ at random, and check that there is no collision on S by making p classical queries to O. Sort the elements of S, and store the pairs $(s, O(s))$ as a table for efficient lookup.
- Construct the oracle $O'(x) = \begin{cases} 1 & \text{if } x \notin S \text{ and } O(x) = O(s) \text{ for some } s \in S \\ 0 & \text{otherwise} \end{cases}$
- Run Grover's algorithm [Gro96] on O' for p iterations to look for a point x such that $O'(x) = 1$.
- Check that there is an $s \in S$ such that $O(x) = O(s)$ by making one more classical query to O.

Before analyzing this construction, we explain what Grover's algorithm does. It takes as input an oracle O' mapping some space $\mathcal{X}$ into $[2]$, and tries to find an x such that $O'(x) = 1$. Specifically, if N points map to 1, then after q queries to O', Grover's algorithm will output an x such that $O'(x) = 1$ with probability $\Theta(q^2 N/|\mathcal{X}|)$

We now analyze this construction. The first step takes p queries to O. If we find a collision, we are done. Otherwise, we have p points that map to p different values. Call this set of values T. The oracle O' outlined in the second step makes exactly one query to O for each query to O'. The number of points in x such that $O'(x) = 1$ is the number of points x in $\mathcal{X} \setminus S$ (which is $|X| - p$) such that $O(x) \in T$. In the random oracle case, the probability that $O(x)$ is one of p random values is $p/|\mathcal{Y}|$, so the expected number of such x is $(|\mathcal{X}| - p)p/|\mathcal{Y}|$. Thus, after p iterations, Grover's algorithm will output such an x with probability $\Theta(p^3(|\mathcal{X}| - p)/|\mathcal{X}||\mathcal{Y}|))$. In the $\mathsf{SR}_r^{\mathcal{Y}}$ case, since there are only r possible outputs, the probability that x maps to T is p/r, so the expected number of such x is $p(|\mathcal{X}| - p)/r$. Thus, Grover's algorithm will output such an x with probability $\Theta(p^3(|\mathcal{X}| - p)/r|\mathcal{X}|)$.

The difference in these probabilities is $\Theta(p^3(1/r - 1/|\mathcal{Y}|)(|\mathcal{X}| - p)/|\mathcal{X}|$. If we let $|\mathcal{Y}|$ be at least $2r$ and $|\mathcal{X}|$ at least $2p + 1 = q$, we see that we distinguish $\mathsf{SR}_r^{\mathcal{Y}}$ from random with probability $\Omega(p^3/r) = \Omega(q^3/r)$, thus matching the bound of Corollary 4.6. This shows that the corollary is optimal, and hence Theorem 4.4 is optimal for the case $\Delta = 0$.

E Security Proof for the Synthesizer-Based PRF

Here, we prove Theorem 6.3 by showing that PRF from Construction 2 is quantum secure if the underlying synthesizer S is standard-secure.

Recall the definition of standard-security for a synthesizer $S : \mathcal{X}^2 \to \mathcal{Y}$ from Definition 6.2: for all sets $\mathcal{Z}$, no efficient quantum algorithm A making classical queries to an oracle O from $\mathcal{Z}^2 \to \mathcal{Y}$ can tell if $O(z_1, z_2) = S(O_1(z_1), O_2(z_2))$ for random oracles $O_i \leftarrow \mathcal{X}^{\mathcal{Z}}$ or if O is truly random.

Since all queries are classical, and only a polynomial number of queries are possible, a simple argument shows that Definition 6.2 is equivalent to the case where $|\mathcal{Z}| \in n^{O(1)}$. Further, if $\mathcal{Z}$ is polynomial in size, we can query the entire set classically, so there is no advantage in having quantum queries. Therefore, Definition 6.2 is equivalent to the following:

Definition E.1 (Standard-Security)**.** *A pseudoreandom synthesizer $S : \mathcal{X}^2 \to \mathcal{Y}$ is standard-secure if, for any set $\mathcal{Z}$ where $|\mathcal{Z}| \in n^{O(1)}$, no efficient quantum algorithm A making quantum queries can distinguish $O(z_1, z_2) = S(O_1(z_1), O_2(z_2))$ where $O_b \leftarrow \mathcal{X}^{\mathcal{Z}}$ from $O \leftarrow \mathcal{Y}^{\mathcal{Z} \times \mathcal{Z}}$.*

Before proving security, we define the quantum-security of a pseudorandom synthesizer. The definition is similar to Definition E.1, except that there is no bound on the size of $\mathcal{Z}$:

Definition E.2 (Quantum-Security)**.** *A pseudoreandom synthesizer $S : \mathcal{X}^2 \to \mathcal{Y}$ is quantum-secure if, for any set $\mathcal{Z}$, no efficient quantum algorithm A making quantum queries can distinguish $O(z_1, z_2) = S(O_1(z_1), O_2(z_2))$ where $O_b \leftarrow \mathcal{X}^{\mathcal{Z}}$ from $O \leftarrow \mathcal{Y}^{\mathcal{Z} \times \mathcal{Z}}$*

We now show that the two definitions are equivalent:

Lemma E.3. *If S is standard-secure, then it is also quantum-secure.*

Proof. Let's define a new oracle distribution, which we will denote $\mathtt{AR}_s$, which stands for *almost random*. $\mathtt{AR}_s$ is defined as follows:

- Pick random oracles P_1 and P_2 from $[s]^{\mathcal{Z}}$.
- Pick a random oracle Q from $\mathcal{Y}^{[s]^2}$.
- Output the oracle $O(z_1, z_2) = Q(P_1(z_1), P_2(z_2))$.

Notice that as s goes to ∞, O_1 and O_2 become injective with probability approaching 1, and thus $\mathtt{AR}_\infty$ is the uniform distribution.

Now, let B be an adversary breaking the oracle-security of S with non-negligible probability ϵ. Define $\epsilon(s)$ as the following quantity:

$$\epsilon(s) = \left| \Pr_{O_1 \leftarrow \mathcal{X}^{\mathcal{Z}}, O_2 \leftarrow \mathcal{X}^{\mathcal{Z}}}[B^{S(O_1,O_2)}() = 1] - \Pr_{O \leftarrow \mathtt{AR}_s}[B^O() = 1] \right|$$

Then $\epsilon = \lim_{s\to\infty} \epsilon(s)$. Let r be an integer such that $\ell(q)/r = \epsilon/8$ where q in the number of queries made by B. We now replace O_i with $\mathsf{SR}_r^{\mathcal{X}}$, and the P_i (as a part of $\mathtt{AR}_s$) with $\mathsf{SR}_r^{[s]}$. Each of these changes will only change the behavior of A by $\epsilon/8$. Thus, a simple argument shows that

$$\left| \Pr_{O_i \leftarrow \mathsf{SR}_r^{\mathcal{X}}}[B^{S(O_1,O_2)}() = 1] - \Pr_{P_i \leftarrow \mathsf{SR}_r^{[s]}, Q \leftarrow \mathcal{Y}^{[s]^2}}[B^{Q(P_1,P_2)}() = 1] \right| \geq \epsilon(s) - \epsilon/2$$

Notice that we can think of the oracle $Q(P_1, P_2)$ as the oracle

$$O'(z_1, z_2) = Q(S_1 \circ R_1(z_1), S_2 \circ R_2(z_2)) = O(R_1(z_1), R_2(z_2))$$

Where $S_i \leftarrow [s]^{[r]}$, $R_i \leftarrow [r]^{\mathcal{Z}}$, and $O(w_1, w_2) = Q(S_1(w_1), S_2(w_2))$. As s goes to ∞, S_i become injective with probability converging to one, so O approaches a random function from $[r]^2 \to \mathcal{Y}$.

We now describe a new algorithm A which tries to break the standard-security of S according to Definition E.1. A takes as input a quantum-accessible oracle O from $[r]^2$ to $\mathcal{Y}$. A constructs two random oracles $R_1 \leftarrow [r]^{\mathcal{Z}}$ and $R_2 \leftarrow [r]^{\mathcal{Z}}$, gives B the oracle $O'(z_1, z_2) = O(R_1(z_1), R_2(z_2))$, and simulates B. If $O = S(T_1, T_2)$ for random oracles $T_i \leftarrow \mathcal{X}^{[r]}$, then the oracle seen by A is $O'(z_1, z_2) = S(O_1(z_1), O_2(z_2))$, where O_1 and O_2 are drawn from $\mathsf{SR}_r^{\mathcal{X}}$. If O is a random oracle, then the oracle seen by A is $O'(z_1, z_2) = O(R_1(z_1), R_2(z_2))$, where $R_i \leftarrow [r]^{\mathcal{Z}}$. This corresponds to the case where $s = \infty$, and thus the advantage of A in distinguishing these two cases is $\epsilon(\infty) - \epsilon/2 = \epsilon/2$. If ϵ is non-negligible, then there is a polynomial bounding r infinitely often, and in these cases, A breaks the standard-security of S.

□

We are now ready to prove that Construction 2 is quantum-secure:

Proof of Theorem 6.3. Let A be a quantum adversary breaking the quantum-security of PRF with probability ϵ. That is,

$$\left| \Pr_{k \leftarrow \mathcal{X}^{2n}}[A^{\mathsf{PRF}_k}() = 1] - \Pr_{O \leftarrow \mathcal{X}^{[2]^n}}[A^O() = 1] \right| = \epsilon$$

Let hybrid H_i be the game where the oracle seen by A is PRF, except that each instance of $\mathsf{PRF}^{(i)}$ is replaced with a truly random function from $[2]^{2^i}$ into $\mathcal{X}$. Since $\mathsf{PRF}^{(0)}$ is already a random function, H_0 is equivalent to the case where the oracle is PRF. Similarly, H_ℓ is by definition the case where the oracle is truly random. Thus a simple hybrid argument shows that there is an i such that A can distinguish H_i from H_{i-1} with probability at least ϵ/ℓ.

We now describe an algorithm B which breaks the quantum-security of S. B is given an oracle from P from $\left(\mathcal{X} \times [2^{\ell-i}]\right)^2$ into $\mathcal{X}$, which is either $S(Q_1, Q_2)$ for random oracles $Q_b \leftarrow \mathcal{X}^{\mathcal{X}\times[2^{\ell-i}]}$ or a truly random oracle. It then constructs oracles

$$P_y(x_1, x_2) = P((x_1, y), (x_2, y))$$

Notice that there are $2^{\ell-i}$ possible y values, and that for fixed y, P_y is either a random oracle from $\mathcal{X}^2$ into $\mathcal{X}$, or it is $S(Q_{y,1}, Q_{y,2})$ for random oracles $Q_{y,b}$ from $\mathcal{X}$ to $\mathcal{X}$. We then construct the oracle O which is PRF, except that we stop the recursive construction at $\mathsf{PRF}^{(i)}$. There are $2^{\ell-i}$ different instances of $\mathsf{PRF}^{(i)}$, so we use the $2^{\ell-i}$ P_y oracles in their place. If P is $S(Q_1, Q_2)$, this corresponds to hybrid H_{i-1}, whereas if P is a random oracle, this corresponds to H_i. Thus, B distinguishes the two cases with probability ϵ/ℓ.

However, under the assumption that S is standard-secure, Lemma E.3 shows that it is quantum-secure, meaning the algorithm B is impossible. Therefore, PRF is quantum-secure. □

F Security Proof for the Direct Construction

Here we give a precise statement and proof for Theorem 7.1, which state's that PRF from Construction 3 is quantum secure for the right parameters. First, we define the Learning With Errors (LWE) problem:

Definition F.1 (Learning With Errors)**.** *Let $q \geq 2$ an integer, n a security parameter, and $m = poly(n)$ and $w = poly(n)$ be integers. For a distribution χ over $\mathbb{Z}$ and a secret matrix $\mathbf{S} \in \mathbb{Z}_q^{n\times w}$, the LWE distribution $\mathsf{LWE}_{\mathbf{S},\chi}$ is the distribution over $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times w}$ defined as follows:*

- *Choose a random matrix $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$.*
- *Choose a random error matrix $\mathbf{E} \leftarrow \chi^{m\times w}$*
- *Output $(\mathbf{A}^t, \mathbf{B}^t = \mathbf{A}^t\mathbf{S} + \mathbf{E} \mod q)$*

The LWE problem is then to distinguish between a polynomial number of samples from $\mathsf{LWE}_{\mathbf{S},\chi}$ for a fixed $\mathbf{S} \leftarrow \chi^{n\times w} \mod q$ from the same number of samples from the uniform distribution. The LWE problem is hard if, for all efficient quantum adversaries A, the probability A distinguishes these two cases is negligible in n.

We now define the oracle-LWE problem:

Definition F.2 (Oracle-LWE)**.** *The oracle-LWE problem is to distinguish an oracle O whose outputs are generated by $\mathsf{LWE}_{\mathbf{S},\chi}$ (for a fixed $\mathbf{S} \leftarrow \chi^{n\times w} \mod q$) from a truly random oracle O. We say that LWE is oracle-hard if, for all efficient adversaries A making quantum queries, A cannot distinguish these two distributions with more than negligible probability.*

Lemma F.3. *If LWE is hard, it is also oracle-hard.*

Proof. The proof is very similar to that of Theorem 4.8. Let A be an adversary breaking the oracle-hardness of LWE using q quantum queries with probability ϵ. Let r be an integer such that $\ell(q)/r \approx \epsilon/4$. We then construct an algorithm B, which takes as input r pairs $(\mathbf{A}_i^t, \mathbf{B}_i^t)$, and distinguishes when the pairs come from $\mathsf{LWE}_{\mathbf{S},\chi}$ for some fixed $\mathbf{S} \leftarrow \chi^{n\times w}$ from when the pairs are random. B works as follows:

- Construct the oracle O where $O(x)$ is selected at random from $(\mathbf{A}_i^t, \mathbf{B}_i^t)$
- Simulate A with oracle O, and output the output of A.

Using the same analysis as in the proof of Theorem 4.8, we get that B distinguishes the two cases with probability $\epsilon/2$. If ϵ is non-negligible, then there is a polynomial that bounds r infinitely often, and in these cases, the number of samples received by B is a polynomial, and hence B breaks the hardness of LWE. □

Next, we need to define the discrete Gaussian distribution:

Definition F.4 (Discrete Gaussian)**.** *Let $D_{\mathbb{Z},r}$ denote the discrete Gaussian distribution over $\mathbb{Z}$, where the probability of x is proportional to $e^{-\pi x^2/r^2}$.*

We are now ready to state and prove Theorem 7.1:

Theorem 7.1. Let $\chi = D_{\mathbb{Z},r}$, and $q \geq p\cdot \ell(Cr\sqrt{n+\ell})^\ell n^{\omega(1)}$ for some suitable universal constant C. Let PRF be as in Construction 3, and suppose each $\mathbf{S}_i$ is drawn from $\chi^{n\times n}$. If the LWE problem is hard for modulus q and distribution χ, then PRF from Construction 3 is a QPRF.

Proof. The proof is very similar to that of Banerjee et al. Notice that our theorem requires $q \geq p\cdot \ell(Cr\sqrt{n+\ell})^\ell n^{\omega(1)}$ whereas the original only requires $q \geq p\cdot \ell(Cr\sqrt{n})^\ell n^{\omega(1)}$. We will explain why this is later. We first define a class of functions $G : \mathcal{K}\times[2]^k \to \mathbb{Z}_q^{m\times n}$ to be PRF without rounding. That is,

$$G_k(x) = \mathbf{A}^t \prod_{i=1}^{\ell} \mathbf{S}_i^{x_i}$$

Then $\mathsf{PRF}_k(x) = \lfloor G_k(x) \rceil_p$. We also define a related class of functions $\tilde{G}$ where $\tilde{G} = \tilde{G}^{(\ell)}$ and

- $\tilde{G}^{(0)}$ is a function from $[2]^0$ into $\mathbb{Z}_q^{m\times n}$ defined as follows: pick a random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, and set $\tilde{G}^{(0)}(\epsilon) = \mathbf{A}^t$.
- $\tilde{G}^{(i)}$ is a function from $[2]^i$ into $\mathbb{Z}_q^{m\times n}$ defined as follows: pick a random $\tilde{G}^{(i-1)}$, pick $\mathbf{S}_i \leftarrow \chi^{n\times n}$ and for each $x' \in [2]^{i-1}$, pick $\mathbf{E}_{x'} \leftarrow \chi^{m\times n}$. Then

$$\tilde{G}^{(i)}(x = x'x_i) = \tilde{G}^{(i-1)}(x')\cdot \mathbf{S}_i^{x_i} + x_i\cdot \mathbf{E}_{x'} \mod q$$

Let A be an adversary that distinguishes PRF from a random function with probability ϵ.

First, consider the case where A sees a truly random function $U : [2]^k \to \mathbb{Z}_p^{m\times n}$. Replace U with $\lfloor U' \rceil_p$ where U' is a truly random function from $[2]^k \to \mathbb{Z}_q^{m\times n}$. For each input, the bias introduced by this rounding is negligible because $q \geq pn^{\omega(1)}$. Thus, by Theorem 4.8, the ability of A to distinguish these two cases is negligible.

Now, let $B = \ell(Cr\sqrt{n+k})^\ell$. Let $\mathsf{BAD}(y)$ be the event that

$$\lfloor y + [-B, B]^{m\times n}\rceil_p \neq \{\lfloor y\rceil_p\}$$

That is, $\mathsf{BAD}(y)$ is the event that y is very close to another element in $\mathbb{Z}_q$ that rounds to a different value in $\mathbb{Z}_p$. Banerjee et al. show that for each x, the probability that $\mathsf{BAD}(U'(x))$ occurs is negligible. Therefore, according to Theorem 4.8, $\mathsf{BAD}(U'(x))$ as an oracle with outputs in $\{\mathsf{True}, \mathsf{False}\}$ is indistinguishable from the oracle that always outputs False. Hence, it is impossible for an algorithm making quantum queries to U' to find an x such that $\mathsf{BAD}(U'(x))$ occurs, except with negligible probability.

The next step is to prove that U' and $\tilde{G}$ are oracle-indistinguishable. Once we have accomplished this, we replace U' with $\tilde{G}$. Then the probability that A detects this change is negligible. Additionally, it is also impossible to find an x such that $\mathsf{BAD}(\tilde{G}(x))$ occurs, except with negligible probability.

Lastly, we replace $\tilde{G}$ with G. Banerjee et al. show that as long as $\mathsf{BAD}(\tilde{G}(x))$ does not occur, $\left\lfloor \tilde{G}(x)\right\rceil_p = \lfloor G_k(x)\rceil_p = \mathsf{PRF}_k(x)$ with all but negligible probability. Our modification to the parameters of the theorem (replacing $\sqrt{n}$ with $\sqrt{n+k}$) allows us to choose C so that this probability is actually $2^{-\ell}\sigma$ for some negligible σ. Summing over all 2^ℓ different x, we get that, except with negligible overall probability, $\mathsf{PRF}_k(x) = \left\lfloor \tilde{G}(x)\right\rceil_p$ whenever $\mathsf{BAD}(\tilde{G}(x))$ does not occur.

Thus, if A distinguishes $\mathsf{PRF}_k(x)$ from $\left\lfloor \tilde{G}(x)\right\rceil_p$ with non-negligible probability, it must be that the sum over all queries made by A of the sum of the query magnitudes of all the x such that $\mathsf{BAD}(\tilde{G}(x))$ occurs is non-negligible (Theorems 3.1 and 3.3 of [BBBV97]). But this means we can find an x such that $\mathsf{BAD}(\tilde{G}(x))$ occurs with non-negligible probability (simply run A, and at a randomly chosen query, halt and sample the query). But, as we have already shown, this is impossible.

Hence, we have shown that PRF is indistinguishable from a random function.

It remains to show that U' and $\tilde{G}$ are oracle-indistinguishable. We show that this is true given that the LWE problem is oracle-hard. Using Lemma F.3, we reach the same conclusion assuming LWE is hard, thus completing the theorem.

Let B be an adversary that distinguishes U' from $\tilde{G}$ with probability ϵ. Define hybrid H_i as the case where B is given the oracle O_i where $O_i = \tilde{G}$, except that, in the recursive definition of $\tilde{G}$, $\tilde{G}^{(i)}$ is replaced with a truly random function. H_0 corresponds to the correct definition of $\tilde{G}$, and H_k corresponds to U'. Thus, there exists an i such that B distinguishes H_i from H_{i-1} with probability ϵ/ℓ.

Construct an adversary C with access to an oracle $P : [2]^{i-1} \to \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times n}$. P is either a random function or each output is chosen according to the LWE distribution. In other words, $P(x) = (\mathbf{A}^t, \mathbf{B}^t)$, where either $\mathbf{A}(x)$ and $\mathbf{B}(x)$ are chosen at random for all x, or there is a secret $\mathbf{S} \leftarrow \chi^{n\times n}$ and $\mathbf{B}(x)^t = \mathbf{A}(x)^t\mathbf{S} + \mathbf{E}(x) \mod q$ where $\mathbf{E}(x) \leftarrow \chi^{m\times n}$.

For each $j > i$, C constructs random oracles $Q_j : [2]^{j-1} \to \mathbb{Z}^{m\times n}$ where $Q_j(x) \leftarrow \chi^{m\times n}$. C also generates $\mathbf{S}_j \leftarrow \chi^{n\times n}$ for $j > i$. Then C works as follows:

- Let $\tilde{G}^{(i)}(x = x'x_i) = \begin{cases} \mathbf{A}(x')^t & \text{if } x_i = 0 \\ \mathbf{B}(x')^t & \text{if } x_i = 1 \end{cases}$
- Let $\tilde{G}^{(j)}(x = x'x_j) = \tilde{G}^{(j-1)}(x') \cdot \mathbf{S}_j^{x_j} + x_j \cdot Q_j(x') \mod q$ for $j > i$.
- Let $O(x) = \tilde{G}^{(k)}(x)$

- Run B with oracle O.

When P is a random oracle, this corresponds to H_i. When P is the LWE oracle, this corresponds to H_{i-1}. Thus, C distinguishes these two cases with probability at least ϵ/ℓ. Under the assumption that LWE is oracle hard, this quantity, and hence ϵ, are negligible. We then use Lemma F.3 to complete the theorem.

□

References

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26:1510–1523, 1997.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In *Advances in Cryptology - ASIACRYPT 2011*, 2011.

[BHT97] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology Column)*, 28:14–19, 1997.

[BL95] Dan Boneh and Richard J. Lipton. Quantum cryptanalysis of hidden linear functions. *Advances in Cryptology - CRYPTO 1995*, 1995.

[BMR10] Dan Boneh, Hart Montgomery, and Ananth Raghunathan. Algebraic Pseudorandom Functions with Improved Efficiency from the Augmented Cascade. *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, pages 1–23, 2010.

[BPR11] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. *Advances in Cryptology - EUROCRYPT 2012*, pages 1–26, 2011.

[DY05] Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. *Public Key Cryptography*, 2005.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to Construct Random Functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of the 28th Annual ACM Symposium on Theory of Computing STOC)*, pages 212–219, 1996.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput.*, 17(2):373–386, April 1988.

[LW09] Allison B. Lewko and Brent Water. Efficient Pseudorandom Functions From the Decisional Linear Assumption and Weaker Variants. *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, pages 1–17, 2009.

[NC00] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. *American Journal of Physics*, 70(5):558, 2000.

[NR95] Moni Naor and Omer Reingold. Synthesizers and their application to the parallel construction of pseudo-random functions. *Proceedings of the 36th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1995.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS)*, 51(2):231–262, 1997.

[NR99] Moni Naor and Omer Reingold. On the Construction of Pseudorandom Permutations : Luby-Rackoff Revisited. *Journal of Cryptology*, (356):29–66, 1999.

[NRR00] Moni Naor, Omer Reingold, and Alon Rosen. Pseudo-random functions and factoring. *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 1–23, 2000.

[Zha12] Mark Zhandry. Identity-based Encryption is Secure in the Quantum Random Oracle Model. *Cryptology ePrint Archive*, 2012.