A Combinatorial Approach to Quantum Random Functions

Nico Döttling[1], Giulio Malavolta[2], and Sihang Pu[1]

[1] CISPA Helmholtz Center for Information Security,
{doettling, sihang.pu}@cispa.saarland,
[2] Max Planck Institute for Security and Privacy.
giulio.malavolta@hotmail.it

Abstract. Quantum pseudorandom functions (QPRFs) extend the classical security of a PRF by allowing the adversary to issue queries on input superpositions. Zhandry [Zhandry, FOCS 2012] showed a separation between the two notions and proved that common construction paradigms are also quantum secure, albeit with a new ad-hoc analysis. In this work we revisit the question of constructing QPRFs and propose a new method starting from small-domain (classical) PRFs: At the heart of our approach is a new domain-extension technique based on bipartite expanders. Interestingly, our analysis is almost entirely classical.

As a corollary of our main theorem, we obtain the first (approximate) key-homomorphic quantum PRF based on the quantum intractability of the learning with errors problem.

1 Introduction

Pseudorandom functions (PRFs) are one of the fundamental building blocks of modern cryptography. PRFs were introduced in the seminal work of Goldreich, Goldwasser and Micali [13] answering the question of how to build a function that is indistinguishable from a random function. Loosely speaking, a PRF guarantees that no efficient algorithm, with oracle access to such a function, can distinguish it from a truly random function. PRFs have been shown to be an invaluable tool in the design of cryptographic primitives (such as block ciphers and message authentication codes) and are by now a well-understood object: After the tree-based construction of [13], PRFs have been build from pseudorandom synthesizers [19] and directly from many hard problems [20,21,22,11,18,7,2].

However, when considering the more delicate quantum settings, the study of the hardness of PRFs is still at its infancy. Before delving into the details of this primitive, some clarification is needed as one can define the quantum security of a PRFs in two ways:

1. The PRF is secure against a quantum machine that can only issue classical queries to the function (although the internal state of the adversary is quantum).
2. The PRF is secure against a quantum machine that is allowed to query it on input superposition states and is given as a response the superposition of the corresponding outputs, i.e., it can issue *quantum queries*. This setting is the focus of our work and we refer to it as *quantum security*.

The first setting is commonly referred to as *post-quantum security* and it involves the use of hard problems that are conjectured to be intractable even for quantum computers, but this aspect typically does not further affect the analysis of known construction paradigms. On the other hand, the latter setting has been shown to require a fundamentally different approach: In his pioneer work, Zhandry [27] gave a separation between the two models, i.e., he constructed a PRF that is post-quantum secure but provably not quantum secure. On the positive side, he showed that the generic constructions of [13] and [19] are also quantum secure, albeit with a completely different analysis. He also provided a quantum analysis of the PRFs of [2], which assumes the post-quantum hardness of the learning with errors problem [24].

Beyond the theoretical interest, quantum security gives a more conservative model to analyze the hardness of PRFs in a world with quantum machines. As an example, if PRF is used as a message authentication code (MAC) by some quantum computer, then it is reasonable to assume that an adversary might be able to obtain the function output when evaluated on some input superposition. In this case, MACs based on post-quantum secure PRFs might not be secure anymore. Boneh and Zhandry in their work[8] studied this problem and constructed the first message authentication codes against quantum chosen message attack. They also showed that a quantum secure PRF is sufficient for constructing a quantum secure MAC. Unfortunately, the current landscape of quantum PRFs is rather unsatisfactory: Current techniques to analyze hardness of PRFs in the quantum settings are geared towards specific constructions. As a result, only a handful of quantum-secure schemes are known.

1.1 What Makes QPRFs Challenging?

At the heart of Zhandry's separation result [27] is the observation that quantum algorithms can detect *hidden linear structures*. This problem is also present when we extend the domain of truly random functions. Assume that $f : \{0,1\}^\lambda \to \{0,1\}^\lambda$ is a uniformly random function and $H : \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ is a random linear function and therefore a universal hash function [9]. The function $x \mapsto f(H(x))$ can easily be shown to be *statistically indistinguishable* from a truly random function for any classical distinguisher with oracle access to this function. However, using the algorithm of Boneh and Lipton [6] one can efficiently find elements in the kernel of H via superposition queries to $f(H(\cdot))$. Given an element z in the kernel of H, $f(H(x))$ can be distinguished from a truly random function $g : \{0,1\}^{2\lambda} \to \{0,1\}^\lambda$ by two classical queries, as it holds for any $x \in \{0,1\}^{2\lambda}$ that $f(H(x+z)) = f(H(x))$. Such a collision, however, happens only with exponentially small probability for a random g.

What this shows is that the advantage of superposition adversaries over classical adversaries goes far beyond their computational advantage. Superposition adversaries can learn strictly more about the structure of a function it is given oracle access to than a classical (even unbounded) adversary ever could.

1.2 Our Results

In this work we explore a different route and we propose a new approach to construct QPRFs. Our construction is based on the framework of Döttling and Schröder [12], which in turn builds on earlier ideas of PRF domain extension [16,4] and constructions of adaptively secure PRFs from non-adaptively secure ones [3].

At the heart of our approach is a domain extension technique based on bipartite expander graphs, which crucially allows us to reduce the quantum hardness of our PRF to the classical (post-quantum) hardness of a small-domain PRF. Specifically, we will prove the following theorem.

Theorem 1 (Informal). *For any q let $\mathsf{PRF}_q : \mathcal{K} \times \mathcal{Y} \to \mathcal{Z}$ be a (post-quantum) classically secure PRF with (small) domain $\mathcal{Y}_q$ and let $\Gamma(x, j)$ be a suitable expander mapping from a vertex x to its j-th degree neighbor, where the expander Γ has degree D_i. Then*

$$F(K, x) = \bigoplus_{i=1}^{\omega(\log \lambda)} \bigoplus_{j \in [D_i]} \mathsf{PRF}_{2^i}(K_{2^i}, \Gamma(x, j)),$$

where $K = (K_{2^1}, \ldots, K_{2^i}, \ldots, K_{2^{\omega(\log \lambda)}})$[3], is a quantum PRF.

This gives an alternative and (arguably) conceptually simpler approach to constructing QPRFs. An interesting aspect of our result is that our analysis concerns almost exclusively the classical settings and quantum security is achieved by a simple observation: The crux of our analysis will consist in reducing the classical hardness of the PRF to that of a small domain PRF, which is also trivially quantum secure since the attacker can query the full domain. This result can be seen as a compiler which converts any post-quantum secure PRF into a QPRF at a moderate overhead and without having to go through the (expensive) GGM construction of [27].

As an additional result, we obtain a new implication: Assuming the quantum-intractability of the learning with errors problem, then there exists a quantum (almost) key-homomorphic PRF.

Quantum Key-Homomorphic PRF. Key-homomorphic PRFs were introduced by Boneh et al. [5] and have applications in the context of proxy-re-encryption and related key security. In a nutshell, for key-homomorphic PRFs the key-space is a group and it holds for all x that $PRF(K_1 + K_2, x) = PRF(K_1, x) + PRF(K_2, x)$. Key-homomorphic PRFs give rise to a very natural protocol for a distributed PRF. Boneh et al. showed that the function

$$\mathsf{PRF}_{\mathsf{KH}}(\mathbf{k}, x) = \left\lfloor \prod_{i=1}^{\ell} \mathbf{A}_{x_i} \cdot \mathbf{k} \right\rceil_p ,$$

where $\mathbf{A}_0$ and $\mathbf{A}_1$ are two random public matrices in $\mathbb{Z}_q^{m \times m}$, is additively key-homomorphic (ignoring a small error) over the vector space $\mathbb{Z}_q^m$. The function is pseudorandom under the learning with errors assumption, which is conjectured to be

[3] Note that we could XOR them from $\log \lambda$ to $\omega(\log \lambda)$, but for simplicity, we still use the range from 1 to $\omega(\log \lambda)$.

intractable also for quantum computers. Then a simple application of our compiler shows us that

$$F(K,x) = \sum_{i=1}^{\omega(\log \lambda)} \sum_{j \in [D_i]} \mathsf{PRF}_{\mathsf{KH}}(K_{2^i}, \mathit{\Gamma}(x,j)) \bmod p$$

is a quantum key-homomorphic PRF.

1.3 Technical Overview

We start by providing a technical outline of our results. As mentioned above, we use the framework of [12] to construct our QPRFs. This framework has two steps, a domain extension step and a combiner step. The domain extension step takes a *small domain* PRF with domain size $\mathsf{poly}(q)$ and constructs from it a q-bounded PRF on a large domain, e.g. $\{0,1\}^\lambda$. A PRF is called q-bounded if security is only guaranteed for adversaries which make at most q queries. An important aspect about this step is that the small domain PRF can be evaluated in time (essentially) independent of q.

The second step, or combiner step, combines a small number of bounded PRFs which have the same domain. The key idea here is to set the bounds in an exponentially increasing way. More specifically, if $\mathsf{PRF}_q(K_q, x)$ are q-bounded PRFs, we combine them into a function F via

$$F(K,x) = \bigoplus_{i=1}^{t} \mathsf{PRF}_{2^i}(K_{2^i}, x)$$

where $K = (K_1, \ldots, K_{2^t})$. We will choose the parameter t to be slightly super-logarithmic in the security parameter λ. We claim that if $\mathsf{PRF}_q(K_q, x)$ is a q-bounded QPRF as long as q is polynomial, then $F(K,x)$ is an (unbounded) QPRF. We will briefly argue how this can be established. Fix a BQP distinguisher $\mathcal{A}$ against the QPRF security of F. Since this distinguisher is efficient, there is a polynomial upper bound q on the number of superposition queries $\mathcal{A}$ will make. Given such a distinguisher we will, choose $i^* = \lceil \log(q) \rceil \leqslant t$ and construct a BQP distinguisher $\mathcal{A}'$ against the 2^{i^*}-bounded security of $\mathsf{PRF}_{2^{i^*}}$. Notice that since $2^{i^*} \leqslant 2q$ and q is polynomial it holds that 2^{i^*} is also polynomial. The distinguisher $\mathcal{A}'$ gets q-bounded superposition access to an oracle $\mathcal{O}$ which computes either $\mathsf{PRF}_{2^{i^*}}$ or a uniformly random function f. Given a superposition query $\sum |x\rangle$ by $\mathcal{A}$, $\mathcal{A}'$ submits this query to its oracle $\mathcal{O}$ obtaining a superposition state $\sum |x\rangle|\mathcal{O}(x)\rangle$. Now, $\mathcal{A}'$ can convert this state into

$$\sum |x\rangle|\mathcal{O}(x) \oplus \bigoplus_{i \neq i^*} \mathsf{PRF}_{2^i}(K_{2^i}, x)\rangle$$

via a local quantum computation and forwards this state to $\mathcal{A}$. In the end, $\mathcal{A}'$ outputs whatever $\mathcal{A}$ outputs. Now notice that if $\mathcal{O}(\cdot)$ computes $\mathsf{PRF}_{2^{i^*}}(K_{2^{i^*}}, \cdot)$, then $\mathcal{A}'$ perfectly simulated superposition access to $F(K,\cdot)$ to $\mathcal{A}$. On the other hand, if $\mathcal{O}(\cdot)$ computes a truly random function, then $\mathcal{O}(x) \oplus \bigoplus_{i \neq i^*} \mathsf{PRF}_{2^i}(K_{2^i}, x)$ is also a truly random function. Consequently, $\mathcal{A}'$ distinguishes $\mathsf{PRF}_{2^{i^*}}$ from uniform with the same advantage that $\mathcal{A}$ distinguishes F from uniform.

The more challenging aspect of our approach is the construction of a q-bounded QPRF from a small domain PRF. As outlined in Section 1.1, even domain extension techniques that are statistically secure against classical adversaries might be completely insecure against a superposition adversary. We circumnavigate this problem by adopting a *perfectly secure* domain extension technique. We can then use a Lemma by Zhandry [27] which states that any classical $2q$-uniform function is identically distributed to a uniform function from the view of a q-bounded superposition adversary.

It turns out that we can realize perfectly secure domain extension using *highly unbalanced expander graphs* via constructions that have previously been used to construct space-efficient k-independent functions [10]. In a nutshell, a highly unbalanced expander is a bipartite graph Γ where the set of left vertices $[N]$ can be made super-polynomially large, the set of right vertices $[L]$ is only polynomially large, and the degree D is polylogarithmic. Moreover, such graphs have a unique neighbor expansion property in the sense that it holds for any subset $S \subset [N]$ of left-vertices not larger than a (polynomial) bound Q that there exists a vertex v in $\Gamma(S) \subset [L]$ (the neighborhood of S) which has a unique neighbor in S. A construction of such graphs was provided by Guruswami, Umans and Vadhan [14].

Equipped with such a graph Γ, we can now extend a random function f defined on *the small domain* [L] to a Q-bounded random function g defined on the large domain $[N]$ as via a simple *tabulation function*. For a left vertex $x \in [N]$ and an index $j \in [D]$, let $\Gamma(x, j) \in [L]$ be the j-th neighbor of x. Define the function g by

$$g(x) = \bigoplus_{j \in [D]} f(\Gamma(x, j)).$$

We claim that if f is a uniformly random function, then g is a Q-uniform function, i.e. it holds for any pairwise distinct $x_1, \ldots, x_Q \in [N]$ that $g(x_1), \ldots, g(x_Q)$ are independent and uniformly random. To see this note that by the unique neighbor expansion property of Γ, as the set $S = \{x_1, \ldots, x_Q\}$ is of size Q there exists a vertex $v \in \Gamma(S)$ which has a unique neighbor x_{i^*} in S. In other words, there is an index $j^* \in [D]$ such that the term $f(\Gamma(x_{i^*}, j^*))$ only appears in

$$g(x_{i^*}) = \bigoplus_{j \in [D]} f(\Gamma(x_{i^*}, j)),$$

but not in any other $g(x_i)$ for $i \neq i^*$. Since $f(\Gamma(x_{i^*}, j^*))$ is uniformly random and independent of all the $g(x_i)$, it follows that $g(x_{i^*})$ is uniformly random and independent of all the $g(x_i)$. We can repeat this argument recursively arguing that the $g(x_1), \ldots, g(x_Q)$ are uniformly random and independent. Now assume that $Q = 2q$. We claim that if PRF is a post-quantum PRF with (polynomially-sized) domain $[L]$, then it holds that

$$F(K, x) = \bigoplus_{j \in [D]} \mathsf{PRF}(K, \Gamma(x, j))$$

is a q-bounded QPRF on the large domain $[N]$. To argue security, assume that $\mathcal{A}$ is a q-bounded BQP distinguisher which distinguishes F from a truly random function. We will first replace PRF with a truly random function f and argue security via the post-quantum security of PRF. Specifically, if $\mathcal{A}$ could distinguish these two cases we can construct a post-quantum distinguisher $\mathcal{A}'$ against the PRF security of PRF. $\mathcal{A}'$ is given access to an oracle $\mathcal{O}$ and proceeds as follows. It first queries $\mathcal{O}$ on every possible input obtaining the entire function table of $\mathcal{O}$. This can be performed efficiently as the domain of $\mathcal{O}$ is of size L, which is polynomial. Now, $\mathcal{A}'$ can give $\mathcal{A}$ superposition access to the function $\mathcal{O}'(x) = \bigoplus_{j \in [D]} \mathcal{O}(\Gamma(x, j))$ via a local quantum computation, since it knows the entire function table of $\mathcal{O}$. Consequently, if $\mathcal{A}$ distinguishes $F(K, x)$ from a function $F'(x) = \bigoplus_{j \in [D]} f(\Gamma(x, j))$ where f is a truly random function, then $\mathcal{A}'$ distiguishes PRF from a truly random function. Finally, as $F'(x) = \bigoplus_{j \in [D]} f(\Gamma(x, j))$ is a $2q$-uniform function, we can argue that since $\mathcal{A}$ is q-bounded it is identically distributed to a uniformly random function from the view of $\mathcal{A}$ via a Lemma by Zhandry [27]. This concludes the overview.

From a conceptual perspective, the main reason why our proof is simpler than, e.g., Zhandry's proof for QPRF security of the GGM construction [27], stems from the fact that the above reduction $\mathcal{A}'$ can query the entire function table of the small domain PRF PRF and simulate a quantum oracle for $\mathcal{A}$ locally.

2 Applications

In this section we discuss the possible applications of quantum secure PRFs.

2.1 Quantum secure MACs

Classically, any pseudorandom function can be used to implement message authentication codes (MAC). Moreover, for quantum adversaries, we can use post-quantum secure PRFs to protect classical messages. However, what if the quantum adversary has the ability to query superpositions of messages? In this situation, the entire chosen message game would be held in the quantum environment which needs stronger version of security. For instance, considering a random oracle H, if the adversary can only issue classical queries, after learning q queries she does not learn any additional information at other inputs; but if she can issue quantum queries, then she might get information on all inputs simultaneously, even with just a single query.

Boneh and Zhandry [8] defined a quantum chosen message attack game to model the security of any MAC scheme in the quantum setting. First, quantum queries need to be explicitly modeled as the adversary could be entangled with the queries. We denote the adversary's state just prior to issuing a signing query by $\Sigma_{m,x,y} \psi_{m,x,y} |m, x, y\rangle$ and the signing oracle performs the following transformation,

$$\Sigma_{m,x,y} \psi_{m,x,y} |m, x, y\rangle \to \Sigma_{m,x,y} \psi_{m,x,y} |m, x \oplus S(k, m; r), y\rangle,$$

where r is a random string and $S(k, m; r)$ is the signing algorithm of a MAC scheme. Then we say that the adversary wins this game if she can generate $q + 1$ valid classical message-tag pairs after issuing q quantum chosen message queries. The formal definition of quantum secure MACs is given as follows [4].

Definition 1. *A MAC system is existentially unforgeable under a quantum chosen message attack (EUF-qCMA) if no adversary can win the quantum MAC game with non-negligible advantage in λ.*

Boneh and Zhandry [8] also showed that a quantum secure pseudorandom function gives rise to the quantum-secure MAC, namely $S(k,m) = \mathsf{PRF}(k,m)$.

Theorem 2 ([8]). *If $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a quantum-secure pseudorandom function and $1/|\mathcal{Y}|$ is negligible, then $S(k,m) = \mathsf{PRF}(k,m)$ is a EUF-qCMA-secure MAC.*

Therefore, the theorem 2 implies that a quantum secure PRF is sufficient to give us a quantum secure MAC.

2.2 Pseudorandom Quantum States

Pseudorandom states (or pseudorandom quantum states, denoted as PRS), are a set of random states $\{|\phi_k\rangle\}$ that is indistinguishable from Haar random quantum states. In [17], Ji et al. generalizes the definition of pseudorandomness in the classical case to the quantum setting:

Definition 2 (Pseudorandom states). *Let κ be the security parameter. Let $\mathcal{H}$ be a Hilbert space and $\mathcal{K}$ the key space, both parameterized by κ. A keyed family of quantum states $\{|\phi_k\rangle \in S(\mathcal{H}_{k\in\mathcal{K}})\}$ is pseudorandom, if the following two conditions hold:*

1. *Efficient generation. There is a polynomial-time quantum algorithm G that generates state $|\phi_k\rangle$ on input k. That is, for all $k \in \mathcal{K}, G(k) = |\phi_k\rangle$.*
2. *Pseudorandomness. Any polynomially many copies of $|\phi_k\rangle$ with the same random $k \in \mathcal{K}$ is computationally indistinguishable from the same number of copies of a Haar random state. More precisely, for any efficient quantum algorithm $\mathcal{A}$ and any $m \in \mathsf{poly}(\kappa)$,*

$$|\Pr_{k\leftarrow\mathcal{K}}[\mathcal{A}(|\phi_k\rangle^{\otimes m}) = 1] - \Pr_{|\psi\rangle\leftarrow\mu}[\mathcal{A}(|\psi\rangle^{\otimes m}) = 1]| = \mathsf{negl}(\kappa)$$

   where μ is the Haar measure on $S(\mathcal{H})$.

Moreover, they also show that any quantum secure PRF could be used to construct PRS as follows.

Theorem 3 ([17]). *For any QPRF $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$, the family of states $\{|\phi_k\rangle\}_{k\in\mathcal{K}}$,*

$$|\phi_k\rangle = \frac{1}{\sqrt{N}} \sum_{x\in\mathcal{X}} \omega_N^{\mathsf{PRF}_k(x)} |x\rangle,$$

is a PRS.

Finally, PRS can be immediately used to construct a private-key quantum money scheme [17].

[4] Recently, blind-unforgeable, a stronger security notion for qMAC is defined in [1]. It implies EUF-qCMA notion and can also be satisfied by quantum secure PRF.

3 Preliminaries

We denote by $\lambda \in \mathbb{N}$ the security parameter, by $\mathsf{poly}(\lambda)$ any function that is bounded by a polynomial in λ, and by $\mathsf{negl}(\lambda)$ any function that is negligible in the security parameter. We abbreviate computational indistinguishability of two distributions by $\approx_c$. The set of N elements is always written as $[N]$. We also denote as $\mathcal{D}^{\mathcal{O}}$ a distinguisher $\mathcal{D}$ access to an oracle $\mathcal{O}$ via classical queries and $\mathcal{A}^{|\mathcal{O}\rangle}$ via quantum queries.

3.1 Quantum Computing

We recall some basic facts about quantum computing.

Fact 1 ([23]) *Any classical efficiently computable function f can be implemented efficiently by a quantum computer. Furthermore, any function that has an efficient classical algorithm computing it can be implemented efficiently as a quantum-accessible oracle.*

Fact 2 ([28]) *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.*

A more formal statement of Fact 2 is given in the following.

Theorem 4 ([28]). *Let A be a quantum algorithm making q quantum queries to an oracle $H : \mathcal{X} \to \mathcal{Y}$. If we draw H from some weight assignment D[5], then for every z, the quantity $\Pr_{H \leftarrow_\$ D}[A^H(\cdot) = z]$ is a linear combination of the quantities $\Pr_{H \leftarrow_\$ D}[H(x_i) = r_i, \forall i \in \{1, \ldots, 2q\}]$ for all possible settings of the x_i and r_i.*

This is proved in [28] and immediately implies that, if two weight assignments on oracles, D_1 and D_2, are $2q$-wise equivalent, then any q query quantum algorithm behaves the same under both weight assignments, since for all $2q$ pairs (x_i, r_i) it holds that

$$\Pr_{H \leftarrow_\$ D_1}[H(x_i) = r_i, \forall i \in \{1, \ldots, 2q\}] = \Pr_{H \leftarrow_\$ D_2}[H(x_i) = r_i, \forall i \in \{1, \ldots, 2q\}].$$

3.2 Pseudorandom Functions

We recall definition of classical pseudorandom functions [13].

Definition 3 (Pseudorandom Functions). *Let $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ be two finite sets depending on λ. We say that an efficiently computable keyed function $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ with key-space $\mathcal{K}_\lambda$ is a pseudorandom function (PRF), if it holds for every PPT oracle adversary $\mathcal{A}$ that*

$$|\Pr[\mathcal{A}^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}^{R}(1^\lambda) = 1]| \leqslant \mathsf{negl}(\lambda),$$

where $K \leftarrow_\$ \mathcal{K}_\lambda$ and $R : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is a randomly chosen function. Moreover, if $|\mathcal{X}| \leqslant \mathsf{poly}(\lambda)$, then we say that PRF is a small-domain PRF, otherwise we call PRF a large-domain PRF.

[5] A weight assignment on a set X is a function $D : X \to R$ such that $\sum_x D(x) = 1$. As an example, and the way we use it in our work, it could model a probability distribution.

If $\mathcal{A}$ is a quantum machine, then we say that the PRF is *post-quantum secure*. Note that $\mathcal{A}$ is restricted to issue only classical queries, but its computation can be quantum. We now recall the notion of q-bounded PRF [12]. The difference between q-bounded PRF and PRF is just the former can only send at most q distinct queries. As in [12], our only restriction is that the runtime of the function depends polynomially on λ and $\log(q)$.

Definition 4 (Bounded Pseudorandom Functions). *Let $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ be finite sets. A keyed function $F_q : \mathcal{K}_q \times X_\lambda \to \mathcal{Y}_\lambda$ parameterized by a parameter q is a q-bounded pseudorandom function (bPRF), if F_q is computable in time $\mathsf{poly}(\lambda, \log(q))$ and if it holds for all efficiently computable $q^* = q(\lambda) \leqslant \mathsf{poly}(\lambda)$ and all q^*-query distinguishers $\mathcal{D}$ (i.e. send at most q^* distinct queries) that*

$$|\Pr[\mathcal{D}^{F_q(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{D}^{R}(1^\lambda) = 1]| \leqslant \mathsf{negl}(\lambda),$$

where $K \leftarrow_{\$} \mathcal{K}_q$ and $R : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is a randomly chosen function.

Quantum Pseudorandom Functions. We define quantum PRFs in the following. Roughly speaking, we say a pseudorandom function PRF is quantum-secure if no efficient quantum adversary $\mathcal{A}$ making quantum queries can distinguish between a random function R and the function PRF. By quantum query we mean that the adversary $\mathcal{A}$ can send a quantum superposition to the oracle and receive a the corresponding quantum superposition of the function evaluation in return.

Definition 5 (Quantum-secure Pseudorandom Functions). *A pseudorandom function $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is quantum-secure if no efficient quantum adversary $\mathcal{A}$ making quantum queries can distinguish between a truly random function R and the function $\mathsf{PRF}(K, \cdot)$ for a random $K \leftarrow_{\$} \mathcal{K}_\lambda$. Specifically, for keyed function $\mathsf{PRF} : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ with key-space $\mathcal{K}_\lambda$, we say it is a quantum-secure pseudorandom function (QPRF) if it holds for every efficient quantum adversary $\mathcal{A}$ that*

$$|\Pr[\mathcal{A}^{|\mathsf{PRF}(K,\cdot)\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|R\rangle}(1^\lambda) = 1]| \leqslant \mathsf{negl}(\lambda),$$

where $K \leftarrow_{\$} \mathcal{K}_\lambda$ and $R : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is a randomly chosen function.

We also define the notion of q-bounded quantum PRFs in a similar spirit as above.

Definition 6 (Bounded Quantum-secure Pseudorandom Functions). *Let $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ be finite sets. A keyed function $F_q : \mathcal{K}_q \times X_\lambda \to \mathcal{Y}_\lambda$ parameterized by a parameter q is a q-bounded quantum-secure pseudorandom function (bQPRF), if F_q is computable in time $\mathsf{poly}(\lambda, \log(q))$ and if it holds for all efficiently computable $q^* = q(\lambda) \leqslant \mathsf{poly}(\lambda)$ and all q^*-query quantum adversary $\mathcal{A}$ (i.e. send at most q^* distinct quantum queries) that*

$$|\Pr[\mathcal{A}^{|F_q(K,\cdot)\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|R\rangle}(1^\lambda) = 1]| \leqslant \mathsf{negl}(\lambda),$$

where $K \leftarrow_{\$} K_q$ and $R : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$ is a randomly chosen function.

4 Bipartite Expanders

Expanders are highly connected sparse graphs, which are significantly useful in computer science, and there is a rich body of work on constructions and properties of expanders (see, e.g., [15] and references therein). We recall the definitions of bipartite graphs and expanders in the following.

Definition 7 (Bipartite Graph). *A bipartite graph with N left-vertices, L right-vertices, and D left-degrees is specified by a function $\mathit{\Gamma} : [N] \times [D] \to [L]$, where $\mathit{\Gamma}(x, j)$ denotes the j-th neighbor of x. For a set $S \subseteq [N]$, we denote as $\mathit{\Gamma}(S)$ its set of neighbors $\{\mathit{\Gamma}(x, j) : x \in S, j \in [D]\}$.*

Definition 8 (Bipartite Expander). *A bipartite graph $\mathit{\Gamma} : [N] \times [D] \to [L]$ is a $(\leqslant Q, A)$ expander if for all $S \subseteq [N]$ with $|S| \leqslant Q$, it has: $|\mathit{\Gamma}(S)| \geqslant A \cdot |S|$, where A is expansion factor.*

We are only interested in highly unbalanced expanders with $N \gg L$. An explicit construction (i.e., where $\mathit{\Gamma}(\cdot, \cdot)$ is computable in polynomial time) of such an expander has been shown in [14]. We recall here the theorem.

Theorem 5 ([14]). *For all constants $\alpha > 0$: for every $N \in \mathbb{N}, Q \leqslant N$, and $\xi > 0$, there is an explicit $(\leqslant Q, (1 - \xi)D)$ expander $\mathit{\Gamma} : [N] \times [D] \to [L]$ with degree $D = O\left(((\log N)(\log Q)/\xi)^{1+1/\alpha}\right)$ and $L \leqslant D^2 \cdot Q^{1+\alpha}$. Moreover, D and L are powers of 2.*

4.1 Q-unique Expanders

In our construction, we need a $(\leqslant Q, (1 - \xi)D)$ expander to be *Q-unique*, which means in every subset of left-vertices with size not greater than Q, there must exist a vertex with a unique neighbor (i.e., this unique neighbor is connected to only one vertex). This property is defined in [10] as constructing functions where every subset S of inputs of size at most Q contains an input that has many unique neighbors. It is formalized as:

Definition 9 (*Q-unique* Expander). *A $(\leqslant Q, (1-\xi)D)$ expander $\mathit{\Gamma} : [N] \times [D] \to [L]$ is Q-unique if for all $S \subseteq [N], |S| \leqslant Q$, there exists a $x \in S$ such that $|\mathit{\Gamma}(\{x\}) \backslash \mathit{\Gamma}(S \backslash \{x\})| > l \geqslant 0$ holds.*

Note l in Definition 9 is a way to measure *uniqueness* of a expander: The greater the l is, the more unique neighbors an input can have. In our construction, we only need $l = 0$ which means (at least) one unique neighbor would be sufficient for us. Moreover, there is also a concept of *Q-wise-independence*:

Definition 10 (*Q-wise-independence*). *Let Q be a positive integer and let $\mathcal{F}$ be a family of functions from $\mathcal{Y}$ to $\mathcal{Z}$. We say that $\mathcal{F}$ is a Q-wise-independent family of functions if, for every choice of $l \leqslant Q$ distinct keys $y_1, \ldots, y_l$ and arbitrary values $z_1, \ldots, z_l$, then, for f selected uniformly at random from $\mathcal{F}$ we have that*

$$\Pr[f(y_1) = z_1, \ldots, f(y_l) = z_l] = |\mathcal{Z}|^{-l}.$$

The existence of such *Q-unique* expanders is showed as follows.

Theorem 6. *Given any $(\leqslant Q, (1-\xi)D)$ expander $\varGamma : [N] \times [D] \to [L]$ from Definition 8, if $\xi < 1/2$, then expander $\varGamma$ is Q-unique for $l = 0$.*

Proof. First we want to show that there must exist a vertex in $\varGamma(S)$ with degree at most one, when $\xi < 1/2$. Assume towards contradiction that every vertex in $\varGamma(S)$ has degree at least 2 when $\xi < 1/2$. Then the number of edges between S and $\varGamma(S)$ is at least two times as $|\varGamma(S)|$. By Definition 7, we have that

$$D \cdot |S| \geqslant 2 \cdot |\varGamma(S)|.$$

Next, by Definition 8, we have $|\varGamma(S)| \geqslant (1-\xi)D \cdot |S|$ (in which $(1-\xi)D$ is expansion factor), therefore

$$\begin{aligned} D \cdot |S| &\geqslant 2(1-\xi)D \cdot |S| \\ 1 &\geqslant 2(1-\xi) \\ \xi &\geqslant 1/2, \end{aligned}$$

which is a contradiction since there is $\xi < 1/2$. It follows that if $\xi < 1/2$, then there exists one vertex in $\varGamma(S)$ with degree less than or equal to 1. However, we already know that the degree cannot be zero since it's in the neighbors set. Therefore it must be 1. This completes the proof. □

Now we will state a useful lemma here. In his seminal paper [25], Siegel showed how a *Q-unique* expander can be combined with a small domain random function to obtain a *Q-wise-independent* function. We use a light variation of Siegel's technique here. It is also used by works [26] and [10].

Lemma 1. *Let $\varGamma : [N] \times [D] \to [L]$ be a Q-unique expander, let $f : [L] \to \{0,1\}^\lambda$ be a uniformly random function and let $h : [N] \to \{0,1\}^\lambda$ be defined by*

$$h(x) = \bigoplus_{j \in [D]} f(\varGamma(x, j)).$$

Then h is a Q-wise-independent function.

4.2 Parameters

Typically, goals in constructing an unbalanced bipartite expander are to maximize the expansion factor A, minimize the degree D, and minimize the size L of the right-hand side ($L \leqslant N$). Although we do not care about the concrete expansion factor A in this work, we still expect a small L (to highly extend domains of PRFs) and small D (to reduce computational overheads). By Theorem 5 we can fix a domain size $N = 2^\lambda$ and a bound $Q = \mathsf{poly}(\lambda)$ and get an explicit expander $\varGamma : [N] \times [D] \to [L]$ where $D = \mathsf{poly}(\log(N), \log(Q))$ and $L = \mathsf{poly}(D, Q)$. Consequently, the degree D is essentially independent of Q and L is of size at most polynomial in Q.

5 Our Quantum Pseudorandom Function

In this section, we present our constrction for a quantum PRF from bipartite expanders. First we show how to construct a perfectly secure (or *loseless*) domain extender which takes as input a small-domain classical (post-quantum) PRF and outputs a q-bounded quantum PRF with large domain. Second we show a combiner that turns a family of q-bounded quantum PRFs into a standard quantum PRF. Note that our proof of security is tight: If a quantum adversary $\mathcal{A}$ can distinguish a quantum PRF F from truly random function R with advantage ϵ, then there exists an adversary $\mathcal{A}'$ which can distinguish a small-domain PRF from a truly random function (issuing only classical queries) with the same advantage ϵ.

5.1 Domain Extension

In the following we present a new domain extension technique based on bipartite expanders. Our compiler is shown below.

Construction 1 *Let* $\mathsf{PRF} : \mathcal{K}_q \times \mathcal{Y} \to \mathcal{Z}$ *be a keyed function with key space* $\mathcal{K}_q$*. Let* $\Gamma : \{0,1\}^\lambda \times [D] \to \{0,1\}^l$ *be a* $(\leqslant 2q, (1-\xi)D)$ *expander with* $\xi \in (0, 1/2)$*. We define the keyed function* $F_q : \mathcal{K}_q \times \mathcal{X} \to \mathcal{Z}$ *with key space* $\mathcal{K}_q$ *by*

$$F_q(K,x) = \bigoplus_{j \in [D]} \mathsf{PRF}(K, \Gamma(x,j)),$$

where $K \leftarrow_{\$} \mathcal{K}_q$, $D = \mathsf{poly}(\lambda), l = \mathcal{O}(\log(\lambda))$*, and* $\mathcal{X} : \{0,1\}^\lambda, \mathcal{Y} : \{0,1\}^l, \mathcal{Z} : \{0,1\}^m$.

The following theorem states that the function F_q is a q-bounded quantum PRF.

Theorem 7. *Let* PRF *and* F_q *be as in Construction 1. If* PRF *is a post-quantum (classically secure) PRF, then* F_q *is a q-bounded quantum PRF. More specifically, if there exists a* $q^* \leqslant \mathsf{poly}(\lambda)$ *and a* q^**-query quantum adversary* $\mathcal{A}$ *that distinguishes* F_{q^*} *from a truly random function* $R : \mathcal{X} \to \mathcal{Z}$ *with advantage* ϵ*, then there exists an efficient quantum adversary* A' *with essentially the same runtime as* $\mathcal{A}$ *that distinguish* PRF *from a truly random function* $R' : \mathcal{Y} \to \mathcal{Z}$ *with advantage at least* ϵ.

Before delving into the proof of the main theorem, we state the following useful lemma. Loosely speaking, we show that if a small-domain PRF is post-quantum secure[6](where the adversary is allowed to issue only classical queries), then such a PRF is also quantum secure. Intuitively, this holds because an adversary can query classically the full domain of the PRF in polynomial time. We stress that the counterexample of [27] does not apply in these settings, since we consider only PRFs with small (poly-sized) domain.

[6] Any small domain PRF built from symmetric primitives is post-quantum secure as long as underling symmetric assumptions are post-quantum secure.

Lemma 2. *Let* PRF *be a small-domain and post-quantum secure PRF as defined in Definition 3, then* PRF *is also quantum-secure as defined in Definition 5. Specifically, if there exists an efficient quantum adversary $\mathcal{A}$ which can make quantum queries to distinguish* $\mathsf{PRF} : \mathcal{K} \times \mathcal{Y} \rightarrow \mathcal{Z}$ *from truly random function* $R : \mathcal{Y} \rightarrow \mathcal{Z}$ *with advantage ϵ,where $|\mathcal{Y}| \leqslant \mathsf{poly}(\lambda)$, then there exists an efficient quantum adversary $\mathcal{A}'$ (with essentially the same runtime as $\mathcal{A}$) that can only make classical queries to distinguish from* PRF *and $\mathcal{R}$ with advantage ϵ.*

Proof. Assume that there exists an efficient (i.e., running in polynomial time) quantum adversary $\mathcal{A}$ who is able to distinguish PRF from a random function $R : \mathcal{Y} \rightarrow \mathcal{Z}$ with advantage ϵ (given quantum oracle access to PRF). We can construct a quantum adversary $\mathcal{A}'$ who only sends classical queries and breaks the security of PRF with the same advantage. From Fact 1, any classical efficiently computable function f can be efficiently implemented by quantum computer, thus we are able to efficiently implement a quantum circuit which computes transformation U_f on quantum computers. Specifically, given input states $|x, y\rangle$, where x corresponds to 'data' register and y corresponds to 'target' register, the quantum circuit corresponding to U_f would transform it into $|x, y \oplus f(x)\rangle$, i.e., $U_f|x, y\rangle = |x, y \oplus f(x)\rangle$. For notational convenience, we also use $U_f|x\rangle$ to denote the state of "target" register after passing through the circuit corresponding to U_f.

```
Quantum Adversary $\mathcal{A}'(1^\lambda)$:
    Obtain function table of $T(x)$ by querying $\mathcal{O}'$ classically;
    Construct the quantum circuit corresponding to $U_T$;
    $b' \leftarrow \mathcal{A}^{|\mathcal{O}(|y\rangle)\rangle}$;
    Output $b'$.
Classical Oracle $\mathcal{O}'(x)$:
    Return $T(x)$.
Quantum Oracle $\mathcal{O}(|y\rangle)$:
    Return $U_T|y\rangle$.
```

Recall that given the description of T, then U_T is efficiently computable. Furhtermore, $\mathcal{A}'$ issues only polynomially-many queries, since PRF has a small domain. We can conclude that $\mathcal{A}'$ is efficient. Consider the case where $T(x) = \mathsf{PRF}(K, x)$, for uniformly chosen $K \leftarrow_{\$} \mathcal{K}$, then O is identically distributed to PRF. On the other hand, if $T(x) = R(x)$ then O is identically distributed to a truly random function. Thus it holds that

$$|\Pr[\mathcal{A}'^{\mathsf{PRF}(K,\cdot)}(1^\lambda) = 1] - \Pr[\mathcal{A}'^{R}(1^\lambda) = 1]| = |\Pr[\mathcal{A}^{|\mathsf{PRF}(K,\cdot)\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|R\rangle}(1^\lambda) = 1]| \\ = \epsilon,$$

which completes the proof. □

We are now in the position of proving the main theorem of this section.

Proof (of Theorem 7). Let $\mathcal{A}$ be a q-query quantum adversary with advantage ϵ against F_q. We are going to construct an adversary $\mathcal{A}'$ with the same advantage against PRF. Consider the following sequence of hybrids.

- **Hybrid$_\mathbf{0}$**: This is defined exactly as the real experiment where $\mathcal{A}$ has oracle access to a function

$$F_0(x) = \bigoplus_{j \in [D]} \mathsf{PRF}(K, \varGamma(x,j)),$$

where K is uniformly sampled from $\mathcal{K}, x \in \mathcal{X}$ and $\varGamma : \mathcal{X} \times [D] \to \mathcal{Y}$ is a $(\leqslant 2q, (1-\xi)D)$ expander as in Construction 1. $\mathcal{A}$ can send at most q distinct quantum queries.
- **Hybrid$_\mathbf{1}$**: This experiment is defined as Hybrid_0 except that we replace PRF with a truly random function $R : \mathcal{Y} \to \mathcal{Z}$. That is, the adversary $\mathcal{A}$ has oracle access to a function

$$F_1(x) = \bigoplus_{j \in [D]} R(\varGamma(x,j)),$$

where R_j is a function uniformly sampled from $\mathcal{Y}$ to $\mathcal{Z}$.
- **Hybrid$_\mathbf{2}$**: This is the ideal experiment where $\mathcal{A}$ has oracle access to a truly random function

$$F_2(x) = R(x)$$

which is uniformly sampled from $\mathcal{X}$ to $\mathcal{Z}$.

Since F_0 and F_2 are in real and ideal experiment, respectively, it holds that

$$|\Pr[\mathcal{A}^{|F_0\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_2\rangle}(1^\lambda) = 1]| = \epsilon.$$

Similarly, we can define two other advantages as:

$$|\Pr[\mathcal{A}^{|F_0\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda) = 1]| = \epsilon_0,$$
$$|\Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_2\rangle}(1^\lambda) = 1]| = \epsilon_1.$$

We first show that $\epsilon_1 = 0$. By Construction 1 we have $\xi \in (0, 1/2)$, thus we know the expander $\varGamma : \{0,1\}^\lambda \times [D] \to \{0,1\}^l$ is *2q-unique* by Theorem 6. By Lemma 1, we know that for all distinct $(x_1, \ldots, x_{2q}) \in \mathcal{X}^{2q}$, the outputs $F_1(x_1), \ldots, F_1(x_{2q})$ are distributed independently and uniformly at random, that is, $Pr[F_1(x_1) = r_1, \ldots, F_1(x_{2q}) = r_{2q}]$ equals to 2^{-2qm}.

Then by Theorem 4, we have

$$\Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda) = 1] = \Pr[\mathcal{A}^{|F_2\rangle}(1^\lambda) = 1],$$

since for all $2q$ pairs (x_i, r_i), it holds that

$$\begin{aligned}\Pr[F_1(x_i) = r_i, \forall i \in \{1, \ldots, 2q\}] &= 2^{-2qm} \\ &= \Pr[F_2(x_i) = r_i, \forall i \in \{1, \ldots, 2q\}].\end{aligned}$$

This means that $\epsilon_1 = 0$. By triangle inequality we have

$$\begin{aligned}\epsilon_0 &= |\Pr[\mathcal{A}^{|F_0\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda) = 1]| \\ &\geqslant |\Pr[\mathcal{A}^{|F_0\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_2\rangle}(1^\lambda) = 1]| - |\Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|F_2\rangle}(1^\lambda) = 1]| \\ &= \epsilon - \epsilon_1 \\ &= \epsilon.\end{aligned}$$

We are left with constructing an adversary that can distinguish a small-domain PRF from a truly random function with advantage ϵ_0. First we allow such an adversary to issue quantum oracle queries. Since we could use Toffoli gates to simulate any classical circuits in quantum settings, without losing generality, let $U_\oplus$ be a quantum circuit to compute $U_\oplus|x_1,\ldots,x_D,y\rangle \to |x_1,\ldots,x_D,y+(x_1\oplus\cdots\oplus x_D)\rangle$ and U_{Γ_j} be another one to compute $U_{\Gamma_j}|x,y\rangle \to |x,y+\Gamma(x,j)\rangle$. The adversary $\mathcal{A}''$ is defined in the following.

```
**Quantum Adversary** $\mathcal{A}''(1^\lambda)$:
For each $j \in [D]$ construct the circuit $U_{\Gamma_j}$;
$b' \leftarrow \mathcal{A}^{|\mathcal{O}(|x\rangle)\rangle}$;
Output $b'$.
**Quantum Oracle** $\mathcal{O}''(|x\rangle)$:
Return $U_T|x\rangle$.
**Quantum Oracle** $\mathcal{O}(|x\rangle)$:
Return $U_\oplus|\mathcal{O}''(U_{\Gamma_1}|x\rangle),\ldots,\mathcal{O}''(U_{\Gamma_D}|x\rangle)\rangle$.
```

Note that $\mathcal{A}$ makes at most q distinct quantum queries ($q \leqslant \mathsf{poly}(\lambda)$), thus $\mathcal{A}''$ is an efficient quantum adversary running in polynomial time. First assume that $T(x) = \mathsf{PRF}(K,x)$ for $K \leftarrow_\$ \mathcal{K}_q$, then the oracle $\mathcal{O}$ in $\mathcal{A}''$'s simulation is identically distributed to $F_0(x)$. On the other hand, if $T(x) = R(x)$ is a uniformly random function, $\mathcal{O}$ computes $\bigoplus_{j=1}^{D} R(\Gamma(x,j))$ which is F_1. Therefore we have

$$\begin{aligned}|\Pr[\mathcal{A}''^{|\mathsf{PRF}(K,\cdot)\rangle}(1^\lambda)=1]-\Pr[\mathcal{A}''^{|R(\cdot)\rangle}(1^\lambda)=1]| &= |\Pr[\mathcal{A}^{|F_0\rangle}(1^\lambda)=1]-\Pr[\mathcal{A}^{|F_1\rangle}(1^\lambda)=1]| \\ &= \epsilon_0 \\ &\geqslant \epsilon.\end{aligned}$$

By Lemma 2 we know that there exists an adversary $\mathcal{A}'$ with the same advantage ϵ, which issues only classical queries, since the PRF has a small (poly-sized) domain. This completes the proof. □

5.2 Unbounded Queries

Finally, we show that the combiner of [12] allows us to remove the restriction on the query bound of our quantum PRF. The innovation of our paper is that we lift the analysis to the quantum settings.

Construction 2 *Let $\omega(\log(\lambda))$ be a slightly super-logarithmic upperbound. For a given parameter q, let $F_q : \mathcal{K}_q \times \mathcal{X} \to \mathcal{Z}$ be a keyed function with corresponding key space $\mathcal{K}_q$. Define the function $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Z}$ with key space $\mathcal{K} = \prod_{i=1}^{\omega(\log(\lambda))} \mathcal{K}_{2^i}$ by*

$$F(K,x) = \bigoplus_{i=1}^{\omega(\log(\lambda))} F_{2^i}(K_{2^i},x),$$

where $K_{2^i} \leftarrow_\$ \mathcal{K}_{2^i}$ for $i = 1,\ldots,\omega(\log(\lambda))$ and $K = (K_{2^i})_{i=1,\ldots,\omega(\log(\lambda))}$.

Theorem 8. *Let F_q and F be as in Construction 2. If F_q is a q-bounded quantum PRF, then F is a quantum PRF. Specifically, if $\mathcal{A}$ is an efficient quantum adversary against F with advantage ϵ that makes at most $q' = \mathsf{poly}(\lambda)$ distinct quantum queries, then there exists an q^*-query quantum adversary $\mathcal{A}'$ (with essentially the same runtime as $\mathcal{A}$) with advantage ϵ against F_{q^*}, where $q^* = 2^{\lceil \log(q') \rceil} \leqslant 2q' = \mathsf{poly}(\lambda)$.*

Proof. Let $\mathcal{A}$ be an efficient quantum adversary which can send quantum superpositions to distinguish F from a truly random function R with advantage ϵ, then we can construct an efficient q^*-query quantum adversary $\mathcal{A}'$ to distinguish F_q from R for some q. Since $q' = \mathsf{poly}(\lambda)$, we have $\log(q') \leqslant \omega(\log(\lambda))$ thus $2^1 \leqslant q^* = 2^{\lceil \log(q') \rceil} \leqslant \mathsf{poly}(\lambda) < 2^{\omega(\log(\lambda))}$ for sufficient large λ.

```
**Bounded Quantum Adversary** $\mathcal{A}'(1^\lambda)$:
Set $i^*$ as $\lceil \log(q') \rceil$;
Generate $K_{2^i}$ for $i \in \{1,\ldots,\omega(\log(\lambda))\}\backslash i^*$;
$b' \leftarrow \mathcal{A}^{|\mathcal{O}(|x\rangle)\rangle}$;
Output $b'$.
**Quantum oracle** $\mathcal{O}'(|x\rangle)$:
Return $U_T|x\rangle$.
**Quantum oracle** $\mathcal{O}(|x\rangle)$:
Return $U_\oplus|F_{2^1},\ldots\mathcal{O}'(|x\rangle),\ldots F_{2^{\omega(\log(\lambda))}}\rangle$.
```

Where $\mathcal{O}'(|x\rangle)$ is the i^*-th element in $\{1,\ldots,\omega(\log(\lambda))\}$ and we write $F_{2^i}(K_i, x)$ as F_{2^i} to simplify the notation. Observe that $q' \leqslant 2^{\lceil \log(q') \rceil} = q^* \leqslant \mathsf{poly}(\lambda)$ thus $\mathcal{A}'$ is able to run $\mathcal{A}$ as a black box and q' queries can be handled by $\mathcal{A}'$. Then we consider distributions of different $T(x)$. If $T(x) = F_{2^{i^*}}(K, x)$ for uniformly randomized $K \leftarrow_\$ \mathcal{K}_{2^{i^*}}$, then oracle $\mathcal{O}$ is identically distributed to $F(K, x)$ for $K \leftarrow_\$ \mathcal{K}$. Otherwise, if $T(x) = R(x)$, then the distribution of oracle $\mathcal{O}$ should be uniform since O' and other F_{2^i} are independent, thus it will be identically distributed to $R(x)$. Therefore it holds that

$$|\Pr[\mathcal{A}'^{|F_{2^{i^*}}\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}'^{|R\rangle}(1^\lambda) = 1]| = |\Pr[\mathcal{A}^{|F\rangle}(1^\lambda) = 1] - \Pr[\mathcal{A}^{|R\rangle}(1^\lambda) = 1]| \\ = \epsilon.$$

which completes the proof. □

References

1. Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 788–817. Springer, Heidelberg, Germany, Zagreb, Croatia (May 10–14, 2020). https://doi.org/10.1007/978-3-030-45727-3_27
2. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 719–737. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_42

3. Berman, I., Haitner, I.: From non-adaptive to adaptive pseudorandom functions. In: Cramer, R. (ed.) TCC 2012: 9th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 7194, pp. 357–368. Springer, Heidelberg, Germany, Taormina, Sicily, Italy (Mar 19–21, 2012). https://doi.org/10.1007/978-3-642-28914-9_20
4. Berman, I., Haitner, I., Komargodski, I., Naor, M.: Hardness preserving reductions via Cuckoo hashing. In: Sahai, A. (ed.) TCC 2013: 10th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 7785, pp. 40–59. Springer, Heidelberg, Germany, Tokyo, Japan (Mar 3–6, 2013). https://doi.org/10.1007/978-3-642-36594-2_3
5. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology – CRYPTO 2013, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 410–428. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2013). https://doi.org/10.1007/978-3-642-40041-4_23
6. Boneh, D., Lipton, R.J.: Quantum cryptanalysis of hidden linear functions (extended abstract). In: Coppersmith, D. (ed.) Advances in Cryptology – CRYPTO'95. Lecture Notes in Computer Science, vol. 963, pp. 424–437. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 27–31, 1995). https://doi.org/10.1007/3-540-44750-4_34
7. Boneh, D., Montgomery, H.W., Raghunathan, A.: Algebraic pseudorandom functions with improved efficiency from the augmented cascade. In: Al-Shaer, E., Keromytis, A.D., Shmatikov, V. (eds.) ACM CCS 2010: 17th Conference on Computer and Communications Security. pp. 131–140. ACM Press, Chicago, Illinois, USA (Oct 4–8, 2010). https://doi.org/10.1145/1866307.1866323
8. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 592–608. Springer, Heidelberg, Germany, Athens, Greece (May 26–30, 2013). https://doi.org/10.1007/978-3-642-38348-9_35
9. Carter, L., Wegman, M.N.: Universal classes of hash functions (extended abstract). In: STOC. pp. 106–112. ACM (1977)
10. Christiani, T., Pagh, R., Thorup, M.: From independence to expansion and back again. In: Servedio, R.A., Rubinfeld, R. (eds.) 47th Annual ACM Symposium on Theory of Computing. pp. 813–820. ACM Press, Portland, OR, USA (Jun 14–17, 2015). https://doi.org/10.1145/2746539.2746620
11. Dodis, Y., Yampolskiy, A.: A verifiable random function with short proofs and keys. In: Vaudenay, S. (ed.) PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 3386, pp. 416–431. Springer, Heidelberg, Germany, Les Diablerets, Switzerland (Jan 23–26, 2005). https://doi.org/10.1007/978-3-540-30580-4_28
12. Döttling, N., Schröder, D.: Efficient pseudorandom functions via on-the-fly adaptation. In: Gennaro, R., Robshaw, M.J.B. (eds.) Advances in Cryptology – CRYPTO 2015, Part I. Lecture Notes in Computer Science, vol. 9215, pp. 329–350. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 2015). https://doi.org/10.1007/978-3-662-47989-6_16
13. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th Annual Symposium on Foundations of Computer Science. pp. 464–479. IEEE Computer Society Press, Singer Island, Florida (Oct 24–26, 1984). https://doi.org/10.1109/SFCS.1984.715949
14. Guruswami, V., Umans, C., Vadhan, S.: Unbalanced expanders and randomness extractors from parvaresh–vardy codes. Journal of the ACM (JACM) **56**(4), 20 (2009)
15. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bulletin of the American Mathematical Society **43**(4), 439–561 (2006)

16. Jain, A., Pietrzak, K., Tentes, A.: Hardness preserving constructions of pseudorandom functions. In: Cramer, R. (ed.) TCC 2012: 9th Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 7194, pp. 369–382. Springer, Heidelberg, Germany, Taormina, Sicily, Italy (Mar 19–21, 2012). https://doi.org/10.1007/978-3-642-28914-9_21
17. Ji, Z., Liu, Y.K., Song, F.: Pseudorandom quantum states. In: Shacham, H., Boldyreva, A. (eds.) Advances in Cryptology – CRYPTO 2018, Part III. Lecture Notes in Computer Science, vol. 10993, pp. 126–152. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2018). https://doi.org/10.1007/978-3-319-96878-0_5
18. Lewko, A.B., Waters, B.: Efficient pseudorandom functions from the decisional linear assumption and weaker variants. In: Al-Shaer, E., Jha, S., Keromytis, A.D. (eds.) ACM CCS 2009: 16th Conference on Computer and Communications Security. pp. 112–120. ACM Press, Chicago, Illinois, USA (Nov 9–13, 2009). https://doi.org/10.1145/1653662.1653677
19. Naor, M., Reingold, O.: Synthesizers and their application to the parallel construction of pseudo-random functions. In: 36th Annual Symposium on Foundations of Computer Science. pp. 170–181. IEEE Computer Society Press, Milwaukee, Wisconsin (Oct 23–25, 1995). https://doi.org/10.1109/SFCS.1995.492474
20. Naor, M., Reingold, O.: Number-theoretic constructions of efficient pseudo-random functions. In: 38th Annual Symposium on Foundations of Computer Science. pp. 458–467. IEEE Computer Society Press, Miami Beach, Florida (Oct 19–22, 1997). https://doi.org/10.1109/SFCS.1997.646134
21. Naor, M., Reingold, O.: On the construction of pseudo-random permutations: Luby-Rackoff revisited (extended abstract). In: 29th Annual ACM Symposium on Theory of Computing. pp. 189–199. ACM Press, El Paso, TX, USA (May 4–6, 1997). https://doi.org/10.1145/258533.258581
22. Naor, M., Reingold, O., Rosen, A.: Pseudo-random functions and factoring (extended abstract). In: 32nd Annual ACM Symposium on Theory of Computing. pp. 11–20. ACM Press, Portland, OR, USA (May 21–23, 2000). https://doi.org/10.1145/335305.335307
23. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th Annual ACM Symposium on Theory of Computing. pp. 84–93. ACM Press, Baltimore, MA, USA (May 22–24, 2005). https://doi.org/10.1145/1060590.1060603
25. Siegel, A.: On universal classes of extremely random constant-time hash functions. SIAM Journal on Computing **33**(3), 505–543 (2004)
26. Thorup, M.: Simple tabulation, fast expanders, double tabulation, and high independence. In: 54th Annual Symposium on Foundations of Computer Science. pp. 90–99. IEEE Computer Society Press, Berkeley, CA, USA (Oct 26–29, 2013). https://doi.org/10.1109/FOCS.2013.18
27. Zhandry, M.: How to construct quantum random functions. In: 53rd Annual Symposium on Foundations of Computer Science. pp. 679–687. IEEE Computer Society Press, New Brunswick, NJ, USA (Oct 20–23, 2012). https://doi.org/10.1109/FOCS.2012.37
28. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology – CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012). https://doi.org/10.1007/978-3-642-32009-5_44