Quantum Security of NMAC and Related Constructions

PRF Domain Extension Against Quantum attacks

Fang Song[1(✉)] and Aaram Yun[2(✉)]

[1] Portland State University, Portland, USA
fang.song@pdx.edu
[2] Ulsan National Institute of Science and Technology (UNIST), Ulsan, Korea
aaramyun@unist.ac.kr

Abstract. We prove the security of NMAC, HMAC, AMAC, and the cascade construction with fixed input-length as *quantum-secure* pseudorandom functions (PRFs). Namely, they are indistinguishable from a random oracle against any polynomial-time quantum adversary that can make quantum superposition queries. In contrast, many blockcipher-based PRFs including CBC-MAC were recently broken by quantum superposition attacks.

Classical proof strategies for these constructions do not generalize to the quantum setting, and we observe that they sometimes even fail completely (e.g., the universal-hash then PRF paradigm for proving security of NMAC). Instead, we propose a direct hybrid argument as a new proof strategy (both classically and quantumly). We first show that a quantum-secure PRF is secure against key-recovery attacks, and remains secure under random leakage of the key. Next, as a key technical tool, we extend the oracle indistinguishability framework of Zhandry in two directions: we consider distributions on *functions* rather than strings, and we also consider a relative setting, where an additional oracle, possibly correlated with the distributions, is given to the adversary as well. This enables a hybrid argument to prove the security of NMAC. Security proofs for other constructions follow similarly.

Keywords: Cascade construction · NMAC · HMAC · Augmented cascade · AMAC · PRF domain extension · Quantum query · Quantum security · Post-quantum cryptography

1 Introduction

After Shor proposed his celebrated quantum algorithm for solving integer factorization and discrete logarithms efficiently, it became apparent that once practical quantum computers become reality, a large part of public-key cryptography, including elliptic curve cryptography and RSA, will be completely broken. Therefore, research in *post-quantum cryptography* has been emerging: new cryptographic algorithms are designed which can still run on conventional classical computers, but their security holds against potential quantum attacks.

J. Katz and H. Shacham (Eds.): CRYPTO 2017, Part II, LNCS 10402, pp. 283–309, 2017.
DOI: 10.1007/978-3-319-63715-0_10

There are two possible approaches for modeling quantum attacks in post-quantum cryptography. One is to assume a quantum attacker who has only quantum computational capabilities. In other words, a classical attacker who has a quantum computer in its garage. Such an attacker can run quantum algorithms, but its interaction with the environment remains classical. In such an adversarial model, while some important classical proof techniques do not carry over such as rewinding [16,19], there are also many examples of existing security proofs that go through relatively easily as long as we switch to hardness assumptions which are not broken by quantum computers [14].

On the other hand, we can be more conservative, and design cryptographic schemes secure against quantum attackers who have not only quantum computational capabilities, but are also capable of interacting quantumly with the environment. In other words, such an attacker can access the cryptographic primitive under attack in quantum *superposition*. Such a scheme would be secure not only now, but also in the far future when quantum computing and quantum networking technologies become prevalent and ubiquitous, and could be also used as a subprotocol in larger quantum computing protocols. We take this adversarial model in this work and refer to this security notion as *quantum security* [20].

Proving quantum security is notoriously challenging. Classically, when an adversary has access to an oracle, each query examines only one point in the domain of the oracle, and that fact is often used crucially in classical security proofs. On the other hand, when an adversary can make superposed queries, each query can potentially probe all points in the input domain in superposition. Therefore, for example, one cannot perform lazy sampling when simulating such an oracle. In fact, there are schemes which are secure classically but fail to be quantum-secure. For example, Kuwakado and Morii showed that three-round Luby-Rackoff cipher [10] and Even-Mansour cipher [11] do not have quantum security, even though they are secure classically.

Later in a series of works [5,6,20], the quantum security of several basic primitives, such as PRFs, MACs and signatures, was proved. However one important question was still largely unclear, as Boneh and Zhandry noted [6]:

> Can we construct a quantum-secure PRF for a large domain from a quantum-secure PRF for a small domain? In particular, do the CBC-MAC or NMAC constructions give quantum-secure PRFs?

Unfortunately, in Crypto 2016, Kaplan et al. showed that many popular MACs and authenticated encryption schemes are not quantum-secure [9]. For example, CBC-MAC is shown to be insecure when the adversary is allowed to make quantum queries, even when the underlying blockcipher is quantum-secure, and the number of blocks are fixed. Since it is known that a quantum-secure PRF is also quantum-secure as a MAC [5], this shows that CBC-MAC is not a quantum-secure PRF, and the same is true for many other blockcipher-based MACs attacked in the paper. Similar results were independently discovered by Santoli and Schaffner in [13]. This brings us to the basic question:

Is domain extension for PRFs possible in the quantum setting?

1.1 Our Contributions

In this paper, we give a positive answer to this question. Our discovery is that NMAC and related schemes like HMAC, AMAC, and the (fixed-length) cascade construction are quantum-secure as PRFs. Together with results in [9], our work provides almost a complete picture on the PRF domain extension problem in the quantum world. We highlight some of our main proof ideas and contributions, followed by a gentle technical overview.

- **A general framework for oracle-indistinguishability of function distributions.** All constructions consist of iterated evaluations of the basic PRF, and the output from previous round is used as the *key* to determine the PRF in the next round. This is essentially giving multiple PRF oracles $F(k_i, \cdot)$ with independent keys k_i to the adversary. Luckily since the number of oracles is polynomially bounded classically (i.e., number of adversary's queries), this does not give the adversary more power by a simple hybrid argument relating to the standard PRF indistinguishability. However, when we allow quantum-accessible oracles, in effect, the adversary can query in quantum superposition exponentially many PRF oracles each with an independently random key. Our first technical contribution shows that, the standard notion of quantum-secure PRF implies this seemingly stronger notion, which enables us to prove security of the cascade construction (for fixed-length inputs) already. More generally we view this as *oracle-indistinguishability* of distributions over *functions*. Therefore we extend Zhandry's work to this setting and show equivalence between ordinary and oracle indistinguishability. We further generalize it, for applications in NMAC for example, to the setting that some additional oracle possibly dependent on the two distributions under consideration is also given to the adversary (we call this relative oracle-indistinguishability).
- **Direct hybrid argument for NMAC and variants.** NMAC and other variants can be viewed as "encrypted" version of the cascade construction by evaluating the output from cascade by another function (e.g., PRF under an independent key). Classical security proofs usually proceed by reducing to some property of its inner cascade. For example the famous "hash-then-PRF" paradigm states that the composition of a *(computationally) almost universal hash function* with a *PRF* gives a secure PRF with larger domain. Bellare [1] shows that the cascade construction is indeed computationally almost universal, and the composition theorem implies that NMAC is a secure PRF immediately. However, it is easy to see that this would not work in the quantum world; there are many universal hash functions with nontrivial periods, and if we start with such a periodic universal hash function, any hash-then-PRF construction inherits that period, which can be detected efficiently by quantum Fourier sampling. Therefore, one cannot prove the quantum security of hash-then-PRF constructions by relying solely on the (computationally almost) universality and the PRF security. Another approach by Gaži, Pietrzak, and Rybár [7] proves the security of NMAC by reducing it to the security of the cascade construction against *prefix-free* queries. However the notion of

prefix-free does not have a natural counterpart in the regime of quantum superposition queries. Instead, we prove the security of NMAC by a direct hybrid argument based on our relative oracle-indistinguishability framework for function distributions. We stress that this also provides an alternative (and cleaner in our opinion) proof for *classical security* as well.

– **Further properties of quantum-secure PRFs.** In proving the security of these constructions, we also give further characterizations and strengthened properties of PRFs. Specifically, we show that a quantum-secure PRF is also secure against key-recovery attacks, and in addition a PRF remains indistinguishable from a random oracle even if the PRF key is leaked in some restricted way. While the corresponding classical results are more or less straightforward, they face considerable difficulties to carry through quantumly. We hence demonstrate more examples and tools of quantum proof techniques where classical security can be "lifted" to quantum security.

Technical Overview. NMAC is a construction producing a variable-input-length PRF $\mathsf{NMAC}[f]$, given a secure PRF $f : \{0,1\}^c \times \{0,1\}^b \to \{0,1\}^c$ (with $b \geq c$)[1]. Here, the first input argument is the key $k \in \{0,1\}^c$, the second input argument is the message block $x \in \{0,1\}^b$, and the output $f(k,x) = y \in \{0,1\}^c$ has the same bit length as the key. NMAC turns this f into a PRF with the key length of $2c$, the output length of c, and the unbounded input length by

$$\mathsf{NMAC}[f]((k_1,k_2),x_1 \dots x_l) := f(k_2, \mathsf{Casc}[f](k_1,x_1 \dots x_l) \| 0^{b-c}),$$

where $\mathsf{Casc}[f]$ is the *cascade construction* given as

$$\mathsf{Casc}[f](k,x_1 \dots x_l) = f(\dots f(f(k,x_1),x_2),\dots,x_l).$$

To explain our methods, first let us discuss the cascade construction. It is well-known that the cascade construction would not be a secure PRF if messages of variable lengths are allowed. For example, an adversary may query $y = \mathsf{Casc}[f](k,x_1) = f(k,x_1)$, and compute $f(y,x_2) = \mathsf{Casc}[f](k,x_1x_2)$, then query $\mathsf{Casc}[f](k,x_1x_2)$ to check if the queried oracle is $\mathsf{Casc}[f]$ or a true random function. To prevent such an *extension attack*, one obvious way is to fix the number of blocks. More generally, one can prove security against prefix-free adversaries, who never make queries m and m' where m is a proper prefix of m'. In fact, the cascade construction is proved to be secure in this sense in [4]. To achieve full security, one would process the output of the cascade construction further, and this would give us schemes like NMAC/HMAC or AMAC.

Quantum security of fixed-length cascade. For quantum security, there seems no natural analogue of prefix-freeness in presence of quantum superposed

[1] To be precise, the definition of NMAC given here is a simplified version which is not exactly the same as the original definition given in [3], which for example can handle messages whose lengths are not divisible by the block length b. However, the differences do not affect the security, so previous works on NMAC, like [1,7], also analyzed this simplified version.

queries. Instead, we consider fixed-input-length cascade $\mathsf{Casc}_l[f]$, processing messages of total block length l, for arbitrary but fixed l.

It is easy to observe that, when $b = 1$, the l-fold cascade $\mathsf{Casc}_l[f]$ is the same as the Goldreich-Goldwasser-Micali construction [8] of a PRF out of a secure PRG. Zhandry in [20] proved that if the underlying PRG is secure against polynomial-time quantum adversaries, then the GGM construction remains quantum-secure. In fact, a PRG is equivalent to a PRF with a polynomial-size domain, therefore Zhandry's proof almost immediately applies to $\mathsf{Casc}_l[f]$ with such a small-sized PRF f. But, to remove the small-domain restriction, we need more work.

To get a sense of the general difficulty of proving quantum security, we briefly review the classical GGM proof. Roughly speaking, two hybrid arguments are used to construct a distinguisher for the underlying PRG from a distinguisher for the GGM construction; one hybrid argument is over the bit-length of the message inputs of the GGM PRF, and the other is over the individual queries made by the adversary. When trying to adapt the classical proof to quantum security, the first hybrid is not at all problematic, but the second hybrid is not usable; since the adversary in general makes many superposed queries which examine all bitstrings of the given length, the fact that only polynomially-many bitstrings are examined by queries of the adversary is no longer true in the quantum setting.

Zhandry resolves this, by observing that the second hybrid is in fact not necessary, and instead the first hybrid can be carried out by relying on the *oracle security* of the underlying PRG. Suppose D is a distribution on a set $\mathcal{Y}$. Let us define $D^{\mathcal{X}}$ as a distribution of functions of form $\mathcal{X} \to \mathcal{Y}$ where for each $x \in \mathcal{X}$, a function value $y \in \mathcal{Y}$ is chosen independently according to D. Then, two distributions D_1 and D_2 are said to be *oracle-indistinguishable*, if $D_1^{\mathcal{X}}$ and $D_2^{\mathcal{X}}$ are indistinguishable for all $\mathcal{X}$. We also say that a PRG G is *oracle-secure*, if its output distribution is oracle-indistinguishable from the uniform random distribution. This notion expresses indistinguishability of possibly exponentially many independent samples from the PRG (indexed by each $x \in \mathcal{X}$) and possibly exponentially many uniform random numbers, and oracle indistinguishability together with the first hybrid argument gives the security proof of GGM, both classically and quantumly. In the classical case, the oracle indistinguishability can be proved via a hybrid argument over the total number of adversarial queries, since at most polynomially many of the samples will be examined by the adversary. On the other hand, in the quantum case, a completely different approach is needed, which is given by Zhandry's "small-range distributions".

Returning to the cascade construction, we need to work with PRFs instead of PRGs. We may follow the same outline of the proof for the GGM construction, except we need oracle security of PRFs. Hence, we adapt the notion of oracle indistinguishability to function distributions. When D is a distribution of functions of form $\mathcal{X} \to \mathcal{Y}$, then for any set $\mathcal{Z}$, we define $D^{\mathcal{Z}}$ as the distribution of functions of form $f : \mathcal{Z} \times \mathcal{X} \to \mathcal{Y}$, sampled by choosing $f(z) \leftarrow D$ independently for each $z \in \mathcal{Z}$. (Note that we are using the 'currying' isomorphism here, regarding f as $f : \mathcal{Z} \to \mathcal{Y}^{\mathcal{X}}$.) Then, the oracle indistinguishability of D_1 and D_2

can be defined as indistinguishability of $D_1^{\mathcal{Z}}$ and $D_2^{\mathcal{Z}}$ for every set $\mathcal{Z}$. We prove oracle security of secure PRFs also by the small-range distributions.

Quantum security of NMAC. We prove the security of NMAC by a direct hybrid argument, adapting the hybrid argument for the cascade construction, rather than reducing to some property of the inner cascade in the classical literature. We start by the standard procedure of swapping the outer instance of the PRF f with a random oracle H; now the modified scheme is $H(\mathsf{Casc}[f](k, x_1 \dots x_l)) = H(f(\dots f(f(k, x_1), x_2), \dots, x_l))$. Using a hybrid argument, we would like to repeatedly swap inner instances of the PRF f with true random functions, until only the true random function remains. However, we need a stronger security notion for the PRF f to do this: while the random oracle H prevents the fatal extension attack, still, queries of different block lengths would leak some information on the inner state of PRF instances. In particular, an adversary can make a single-block query x to obtain $H(f(k, x))$, and make a zero-block query to obtain $H(k)$. Here, the hash value $H(k)$ of the secret key k is leaked by the random oracle H, and this prevents using the indistinguishability of the PRF f. What we need is that $f(k, \cdot)$ should remain pseudorandom even when $H(k)$ is leaked and the random oracle H is accessible. We call this property the *security under random leakage*. Nonetheless, we prove that a quantum-secure PRF remains quantum-secure under random leakage, and therefore we do not need to impose this additional condition on a PRF.

To carry out the hybrid argument, however, we need another augmentation to the oracle indistinguishability: while our NMAC security proof itself is in the standard model, a random oracle H is introduced during the security proof, and the PRF security under random leakage is inherently a security notion in the (quantum) random oracle model. Hence we introduce and study oracle indistinguishability of function distributions, *relative to a random oracle* H. The function distributions may be in general dependent on the random oracle H, and an adversary always in addition has access to H to attack indistinguishability or oracle indistinguishability. The tools we introduced so far are enough to enable us to complete the hybrid argument and prove quantum security of NMAC finally.

Quantum security of augmented cascade and AMAC. In [2], Bellare, Bernstein, and Tessaro prove PRF security of AMAC. In fact, they analyze ACSC, which is the *augmented cascade*. We can say that ACSC is to AMAC as NMAC is to HMAC. In ACSC, the output of the usual cascade construction $\mathsf{Casc}[f]$ is further processed by a keyless output transform Out, which is typically truncation: $\mathsf{Out}(b_1 \dots b_c) = b_1 \dots b_r$ for some $r < c$. They show that the augmented cascade is a secure PRF, if f is secure under Out-leakage, that is, $f(k, \cdot)$ remains pseudorandom even when $\mathsf{Out}(k)$ is leaked. In this paper, using oracle indistinguishability of functions, we also prove that ACSC is quantum-secure if f is secure under Out-leakage.

Organization. Sect. 2 introduces basic notations and definitions. We develop our technical tool of oracle distribution for function distributions in Sect. 3.

Combined with the further properties of PRFs we establish in Sect. 4, we prove quantum security of NMAC and other constructions in Sect. 5.

2 Preliminaries

2.1 Notations and Conventions

In this paper, all constructions and security notions are *implicitly* asymptotic: many quantities and objects are parametrized by the main security parameter λ, but for simplicity, we will often omit writing the dependency on λ explicitly. Although it is in reality a family of sets $\{\mathcal{X}_\lambda\}_\lambda$, we write it simply as $\mathcal{X}_\lambda$, or even just $\mathcal{X}$. Similarly, a function $f : \mathcal{X} \to \mathcal{Y}$ in such a case is really a family $\{f_\lambda : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda\}_\lambda$ of functions. We also omit the size input 1^λ from arguments of polynomial-time computable functions.

A quantity $p = p(\lambda)$ is *polynomially bounded*, if $p(\lambda) = O(\lambda^d)$ for some $d > 0$. We denote this as $p(\lambda) = poly(\lambda)$, or even, $p = poly()$. Similarly, a quantity $\epsilon = \epsilon(\lambda)$ is *negligible*, if $\epsilon(\lambda) \leq 2^{-\omega(\log \lambda)}$. We denote this as $\epsilon(\lambda) = negl(\lambda)$, or even, $\epsilon = negl()$.

If D is a distribution, then $x \leftarrow D$ means x is sampled according to D. Also, if $\mathcal{X}$ is a set, then $x \leftarrow \mathcal{X}$ means that x is sampled from $\mathcal{X}$ uniform randomly.

For any $r \in \mathbb{N}$, we define $[r] := \{0, 1, \ldots, r-1\}$.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets. We denote by $\mathcal{Y}^\mathcal{X}$ the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We sometimes call it the *function space* from $\mathcal{X}$ to $\mathcal{Y}$.

In this paper, we are mostly interested in quantum security. Unless explicitly mentioned otherwise, by an *adversary*, we always mean a polynomial-time quantum algorithm which may have access to some oracles, to which it can make polynomially many quantum superposed queries. Similarly, when we mention 'security', unless it is in a context describing previous works and comparing them with ours, it means quantum security. On the other hand, by an 'algorithm', we always mean a classical algorithm, unless mentioned otherwise.

2.2 I.i.d Samples of Functions

Following Zhandry [20], we introduce the notation $D^\mathcal{X}$ as follows.

Definition 2.1 (Indexed family of i.i.d. samples). *Let D be a probability distribution over a set $\mathcal{Y}$, and let $\mathcal{X}$ be another set. Then, we denote by $D^\mathcal{X}$ the probability distribution over $\mathcal{Y}^\mathcal{X}$, defined such that, f is sampled according to $D^\mathcal{X}$ if and only if $f(x)$ is sampled according to D, independently for each $x \in \mathcal{X}$.*

In other words, if $f \leftarrow D^\mathcal{X}$, then $\{f(x)\}_{x \in \mathcal{X}}$ is an indexed family of i.i.d. samples, where each $f(x)$ is distributed according to D.

Suppose D is a distribution over $\mathcal{Y}^\mathcal{X}$. Since $\mathcal{Y}^\mathcal{X}$ itself is just a set, the previous definition is applicable. Let us clarify this as the following definition.

Definition 2.2 (Indexed family of i.i.d. samples of functions). *Let D be a probability distribution over $\mathcal{Y}^{\mathcal{X}}$, and let $\mathcal{Z}$ be another set. We define the distribution $D^{\mathcal{Z}}$ of functions $f \in (\mathcal{Y}^{\mathcal{X}})^{\mathcal{Z}}$ as in* Definition 2.1*; if f is sampled according to $D^{\mathcal{Z}}$, then $f(z) \in \mathcal{Y}^{\mathcal{X}}$ is sampled according to D, independently for each $z \in \mathcal{Z}$.*

Then, evaluating $f(z) \in \mathcal{Y}^{\mathcal{X}}$ on $x \in X$ will give a value $f(z)(x) = y \in \mathcal{Y}$. Considering the 'currying' isomorphism $(\mathcal{Y}^{\mathcal{X}})^{\mathcal{Z}} \cong \mathcal{Y}^{\mathcal{Z}\times\mathcal{X}}$, we may regard $D^{\mathcal{Z}}$ as a distribution over $\mathcal{Y}^{\mathcal{Z}\times\mathcal{X}}$, writing $f(z,x)$, instead of $f(z)(x)$. We will use the two perspectives interchangeably.

In this paper, although our results are *not* in the quantum random oracle model, during the security proofs, we mostly work in the quantum random oracle model. In other words, all players, including the adversary, are given oracle access to a uniform random function $H : \mathcal{A} \to \mathcal{B}$, and various constructions depend on H. Therefore, we need to consider the case when a distribution D over $\mathcal{Y}^{\mathcal{X}}$ depends on H, that is, D and the uniform distribution of H are both marginal distributions of a joint distribution. Therefore, we give a definition of $D^{\mathcal{Z}}$, relative to a random oracle H:

Definition 2.3 (Indexed family of relative i.i.d. samples of functions). *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, that is, a uniform random function in $\mathcal{B}^{\mathcal{A}}$. And let D be a probability distribution over $\mathcal{Y}^{\mathcal{X}}$ which depends on H, and let $\mathcal{Z}$ be another set. We define the distribution $D_H^{\mathcal{Z}}$* relative to *H as follows. To jointly sample f from $D_H^{\mathcal{Z}}$ and also a particular $h : \mathcal{A} \to \mathcal{B}$ as realization of the random variable H, first sample $h \leftarrow \mathcal{B}^{\mathcal{A}}$ uniform randomly, and form $D|h$, which is the conditional distribution of D conditioned on the event $H = h$. Finally, sample $f \leftarrow (D|h)^{\mathcal{Z}}$. When the dependence on the random oracle H is clear, we abuse the notation and simply write $D^{\mathcal{Z}}$, instead of $D_H^{\mathcal{Z}}$.*

In other words, when we are in the quantum random oracle model, at first a function h is sampled uniformly, as a realization of the random variable H. When a distribution D is dependent on H, then sampling $f \leftarrow D^{\mathcal{Z}}$ means that, $f(z)$ is independently sampled from $D|h$, for each $z \in \mathcal{Z}$.

2.3 Various Security Notions of PRFs

First, let us define the syntax of the pseudorandom function as follows:

Definition 2.4 (Pseudorandom function). *A pseudorandom function (PRF) is a polynomial-time computable function f of form $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. We call the sets $\mathcal{K}$, $\mathcal{X}$, $\mathcal{Y}$ as the key space, the domain, and the codomain of f, respectively.*

The domain of a PRF may be of fixed size or arbitrarily large. For a blockcipher, $\mathcal{X}$ would be $\{0,1\}^n$ for some n. On the other hand, for HMAC, the domain $\mathcal{X}$ is the set of all bitstrings, or bitstrings up to some large fixed length.

In this paper, we are concerned with polynomial-time quantum adversaries who can make quantum superposed queries to their oracles. Therefore, our standard definition of PRF security is as follows:

Definition 2.5 (Quantum security of PRF). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that f is* secure*, if for any adversary A, we have the following:*

$$\mathbf{Adv}_f^{\mathsf{prf}}(A) := \left|\mathbf{Pr}[A^{f(k,\cdot)}() = 1] - \mathbf{Pr}[A^{\rho}() = 1]\right| = negl(),$$

where $k \leftarrow \mathcal{K}$, $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$ are uniformly and independently random.

That is, sampling $k \leftarrow \mathcal{K}$ and letting F as $F(x) := f(k, x)$, any quantum adversary cannot distinguishing F from a true random function $\rho : \mathcal{X} \to \mathcal{Y}$.

Here, $\mathbf{Adv}_f^{\mathsf{prf}}(A)$ is the *advantage* of A in distinguishing $f(k, \cdot)$ from ρ, and if f is a secure PRF, then the advantage is negligible for any adversary A.

Sometimes, we may want less than the full PRF security against distinguishing attack, and only require the following:

Definition 2.6 (Security of PRF against key recovery). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that f is* secure against key recovery*, if for any adversary A, we have the following:*

$$\mathbf{Adv}_f^{\mathsf{prf\text{-}kr}}(A) := \mathbf{Pr}[A^{f(k,\cdot)}() = k] = negl(),$$

where $k \leftarrow \mathcal{K}$ is uniformly random.

Classically, it is well known, and indeed trivial to prove that a secure PRF is also secure against key recovery. However, in the quantum world, it is less trivial than classically. We discuss this more in Sect. 4.

Finally, let us present a stronger security notion for PRF, which will be crucial later when we prove the security of NMAC.

Definition 2.7 (Security of PRF under random leakage). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that f is* secure under random leakage*, if for any set $\mathcal{W}$ and any adversary A, we have the following:*

$$\mathbf{Adv}_f^{\mathsf{prf\text{-}rl}}(A) := \left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{\rho,H}(w) = 1]\right| = negl(),$$

where $k \leftarrow \mathcal{K}, w \leftarrow \mathcal{W}, H \leftarrow \mathcal{W}^{\mathcal{K}}, \rho \leftarrow \mathcal{Y}^{\mathcal{X}}$ are uniform, independent random.

The above notion is related to the leakage-resilient cryptography. Here, the PRF key k is leaked once, via the leakage function $H(\cdot)$. But, this leakage is very weak; the adversary does not choose H, which is just a random oracle.

2.4 NMAC and Related Constructions

In this subsection, we give definitions of NMAC and other hash-based PRFs which we study in this paper.

Cascade construction. Suppose that $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ is a PRF where the codomain is the same as the key space $\mathcal{K}$. We define the *l-fold cascade* of f,

denoted by $\mathsf{Casc}_l[f] : \mathcal{K} \times \mathcal{X}^l \to \mathcal{K}$, as follows: given $k \in \mathcal{K}$ and $x_1, \ldots, x_l \in \mathcal{X}$, we define a sequence of values $y_0, \ldots, y_l \in \mathcal{K}$, recursively.

$$\begin{aligned} y_0 &:= k, \\ y_i &= f(y_{i-1}, x_i), \qquad \text{for } i = 1, \ldots, l. \end{aligned}$$

Then, the cascade PRF is given as the last value y_l.

$$\mathsf{Casc}_l[f](k, x_1 \ldots x_l) := y_l.$$

In other words,

$$\mathsf{Casc}_l[f](k, x_1 \ldots x_l) = f(\ldots f(f(k_1, x_1), x_2), \ldots, x_l).$$

From the definition of $\mathsf{Casc}_l[f]$, we see $\mathsf{Casc}_0[f] : \mathcal{K} \times \mathcal{X}^0 \to \mathcal{K}$ is given as

$$\mathsf{Casc}_0[f](k, \epsilon) = k,$$

where $\epsilon \in \mathcal{X}^0$ is the empty string of length 0.

NMAC. Suppose that $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ is a PRF where the codomain is the same as the key space $\mathcal{K}$. Here, we assume that $|\mathcal{K}| \leq |\mathcal{X}|$. The *NMAC* of f, denoted by $\mathsf{NMAC}[f] : \mathcal{K}^2 \times \mathcal{X}^* \to \mathcal{K}$ is defined as

$$\mathsf{NMAC}[f]((k_1, k_2), x_1 \ldots x_m) := f(k_2, \mathsf{pad}(\mathsf{Casc}_m[f](k_1, x_1 \ldots x_m))),$$

where $\mathsf{pad} : \mathcal{K} \to \mathcal{X}$ is a simple injective 'padding function'. Typically, when $\mathcal{X} = \{0,1\}^b$ and $\mathcal{K} = \{0,1\}^c$, then $\mathsf{pad}(k) = k \| 0^{b-c}$, but the choice of pad does not affect the security of NMAC.

Augmented cascade. Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ be a PRF where the codomain is the same as the key space $\mathcal{K}$, and let $\mathsf{Out} : \mathcal{K} \to \mathcal{Y}$ be an unkeyed function. Then, the *augmented cascade* $\mathsf{ACSC}[f, \mathsf{Out}] : \mathcal{K} \times \mathcal{X}^* \to \mathcal{Y}$ is

$$\mathsf{ACSC}[f, \mathsf{Out}](k, x_1 \ldots x_m) := \mathsf{Out}(\mathsf{Casc}_m[f](k, x_1 \ldots x_m)).$$

2.5 Implementing Oracles

Here, we are going to discuss which function distributions can be 'efficiently implemented'. One possible answer is the following:

Definition 2.8. *Let D be a function distribution over $\mathcal{Y}^{\mathcal{X}}$. We say that D is* efficiently samplable*, if there exists a set $\mathcal{R}$ and a polynomial-time deterministic algorithm $D.eval : \mathcal{R} \times \mathcal{X} \to \mathcal{Y}$, such that sampling $f \in \mathcal{Y}^{\mathcal{X}}$ according to the distribution D can be done by sampling $r \leftarrow \mathcal{R}$ and defining f by $f(x) := D.eval(r, x)$. We also require that $\log |\mathcal{R}| = poly()$.*

In other words, we may sample a function $f \leftarrow D$, by sampling $r \leftarrow \mathcal{R}$.

One typical example of an efficiently samplable distribution is PRF_f over $\mathcal{Y}^{\mathcal{X}}$ of a PRF $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$. Here, $\mathcal{R} = \mathcal{K}$, and $\mathsf{PRF}_f.eval = f$.

Zhandry shows that in fact we can efficiently 'implement' function distributions which are not necessarily efficiently samplable. One such example is the uniform distribution over $\mathcal{Y}^{\mathcal{X}}$. While it is not efficiently samplable in the above sense, still, given any adversary A making at most q quantum superposed queries, it is possible to implement the uniform distribution for the adversary A perfectly. This is due to Theorem 3.1 of [21]. Here, we give a slightly extended version as follows, whose proof we defer to the full version of our paper [15], due to page limitation.

Theorem 2.9. *Let A be an adversary having oracle access to $O_1, \ldots, O_t$, and makes at most q_i quantum queries to $O_i \in \mathcal{Y}_i^{\mathcal{X}_i}$ for $i = 1, \ldots, t$. If we draw O_i from any joint distribution for $i = 1, \ldots, t$, then for every v, the quantity $\mathbf{Pr}[A^{O_1,\ldots,O_t}() = v]$ is a linear combination of the quantities*

$$\mathbf{Pr}[\forall i \in \{1, \ldots, t\}, \forall j \in \{1, \ldots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}]$$

for all possible settings of the values $x_j^{(i)} \in \mathcal{X}$ and $y_j^{(i)} \in \mathcal{Y}$.

Hence if D, D' are distributions over $\mathcal{Y}^{\mathcal{X}}$ which are $2q$-wise equivalent, i.e.,

$$\mathbf{Pr}_{O \leftarrow D}[\forall i \in \{1, ..., 2q\}, O(x_i) = y_i] = \mathbf{Pr}_{O \leftarrow D'}[\forall i \in \{1, ..., 2q\}, O(x_i) = y_i],$$

for any distinct $x_1, \ldots, x_{2q} \in \mathcal{X}$ and any $y_1, \ldots, y_{2q} \in \mathcal{Y}$, then when A makes at most q queries to its oracle, for any output value v of A, we have

$$\mathbf{Pr}_{O \leftarrow D}[A^O() = v] = \mathbf{Pr}_{O \leftarrow D'}[A^O() = v].$$

In particular, for any adversary making at most q quantum queries, the uniform random function $\mathsf{U} \in \mathcal{Y}^{\mathcal{X}}$ can be efficiently 'implemented' by any $2q$-wise independent function family. We use the following standard fact (for example, see p. 72 of [18]):

Proposition 2.10. *For every n, m, k, there exists a family of k-wise independent functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ such that, choosing a function h from $\mathcal{H}$ takes $k \cdot \max\{n, m\}$ random bits, and evaluating $h \in \mathcal{H}$ takes time $poly(n, m, k)$.*

Therefore, implementing a uniform distribution in $\mathcal{Y}^{\mathcal{X}}$ for any adversary making q quantum queries requires sampling $2q \cdot \max\{\log|\mathcal{X}|, \log|\mathcal{Y}|\}$ bits, and answering one query takes time $poly(\log|\mathcal{X}|, \log|\mathcal{Y}|, q)$.

Let us propose the following definition which captures both efficiently samplable distributions and uniform distributions.

Definition 2.11. *Let D be a function distribution over $\mathcal{Y}^{\mathcal{X}}$. We say that D is* bounded samplable*, if there exists a set $\mathcal{R}^{(q)}$ for each q and a polynomial-time deterministic algorithm $D.eval : 1^* \times \bigcup_q \mathcal{R}^{(q)} \times \mathcal{X} \to \mathcal{Y}$ such that, if we sample $f \in \mathcal{Y}^{\mathcal{X}}$ according to the distribution D, and sample $f' \in \mathcal{Y}^{\mathcal{X}}$ by sampling $r \leftarrow \mathcal{R}^{(q)}$ and defining f' by $f'(x) := D.eval(1^q, r, x)$, then two random functions f and f' are $2q$-wise equivalent. Also, we require that $\log|\mathcal{R}^{(q)}| = poly(\lambda, q)$.*

If D is bounded samplable, then a function f can be sampled according to D by sampling $r \leftarrow \mathcal{R}^{(q)}$, and it can be evaluated by $f(x) = D.eval(1^q, r, x)$. The resulting distribution may not be identical to D, but would be enough to 'fool' any adversary making at most q queries. The following lemma is obvious.

Lemma 2.12. *For any $\mathcal{X}, \mathcal{Y}$, the uniform distribution over $\mathcal{Y}^{\mathcal{X}}$ is bounded samplable.*

Moreover, we can see from Theorem 2.9 that, when A has access to several oracles $O_1, \ldots, O_t$ sampled according to $D_1, \ldots, D_t$, and if they are *independent*, then if the distributions D_i are all bounded samplable, then they can be 'implemented' separately: sampling $f_i \leftarrow D_i$ can be done by sampling $r_i \leftarrow \mathcal{R}_i^{(q)}$, and letting $f_i(x) = D_i.eval(1^{q_i}, r_i, x)$ for all $i = 1, \ldots, t$, since we have

$$\begin{aligned}&\mathbf{Pr}[\forall i \in \{1, \ldots, t\}, \forall j \in \{1, \ldots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}] \\ &= \prod_{i=1}^{t} \mathbf{Pr}[\forall j \in \{1, \ldots, 2q_i\}, O_i(x_j^{(i)}) = y_j^{(i)}]\end{aligned}$$

Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, that is, a uniform random function. For our purpose, we need to 'relativize' the efficient samplability and the bounded samplability, with respect to H. First, let us give the following definitions.

Definition 2.13. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D_i be a distribution over $\mathcal{Y}_i^{\mathcal{X}_i}$, for $i = 1, \ldots, t$. We say that $D_1, \ldots, D_t$ are* conditionally independent relative to H, *if for any $h \in \mathcal{B}^{\mathcal{A}}$, the distributions $D_1, \ldots, D_t$ are independent, conditioned on the event that $H = h$.*

Definition 2.14. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D, D' be distributions over $\mathcal{Y}^{\mathcal{X}}$. We say that D, D' are k-*wise equivalent relative to H, *if*

$$\begin{aligned}&\mathbf{Pr}_{O \leftarrow D}[\forall i \in \{1, ..., k\}, O(x_i) = y_i \mid H = h] \\ &\quad = \mathbf{Pr}_{O \leftarrow D'}[\forall i \in \{1, ..., k\}, O(x_i) = y_i \mid H = h],\end{aligned}$$

for any distinct $x_1, \ldots, x_k \in \mathcal{X}$, any $y_1, \ldots, y_k \in \mathcal{Y}$, and any $h \in \mathcal{B}^{\mathcal{A}}$.

Then, we are ready to define relative versions of efficient samplability and bounded samplability as follows.

Definition 2.15. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D be a distribution over $\mathcal{Y}^{\mathcal{X}}$. We say that D is* efficiently samplable relative to H, *if there exists a set $\mathcal{R}$ and a polynomial-time deterministic oracle algorithm $D.eval^H : \mathcal{R} \times \mathcal{X} \to \mathcal{Y}$ such that sampling $f \in \mathcal{Y}^{\mathcal{X}}$ according to D can be done by sampling $r \leftarrow \mathcal{R}$ and defining f by $f(x) := D.eval^H(r, x)$. We also require that $\log |\mathcal{R}| = poly()$.*

Definition 2.16. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D be a distribution over $\mathcal{Y}^{\mathcal{X}}$. We say that D is* bounded samplable relative to H, *if there exists a set $\mathcal{R}^{(q)}$ for each q, and a polynomial-time deterministic oracle algorithm $D.eval^H : 1^* \times \bigcup_q \mathcal{R}^{(q)} \times \mathcal{X} \to \mathcal{Y}$ such that, if we sample $f \in \mathcal{Y}^{\mathcal{X}}$*

according to D, and sample $f' \in \mathcal{Y}^{\mathcal{X}}$ by sampling $r \leftarrow \mathcal{R}^{(q)}$ and defining f' by $f'(x) := D.eval^H(1^q, r, x)$, then f and f' are $2q$-wise equivalent relative to H. We also require that $\log|\mathcal{R}^{(q)}| = poly(\lambda, q)$.

We have the following lemma about 'relative implementation' of an oracle.

Lemma 2.17. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D be a distribution over $\mathcal{Y}^{\mathcal{X}}$. Suppose D is bounded samplable relative to H, and suppose an adversary $A^{O,H}$ makes at most q queries to O, and at most q_H queries to H. Let D' be the distribution of function O sampled by sampling $r \leftarrow \mathcal{R}^{(q)}$ and letting $O(x) := D.eval^H(1^q, r, x)$. Then, we have*

$$\mathbf{Pr}_{O \leftarrow D}[A^{O,H}() = v] = \mathbf{Pr}_{O \leftarrow D'}[A^{O,H}() = v],$$

for any possible output value v of A.

Lemma 2.17 is an extension of the previous result that, if a distribution D is bounded samplable, then for each adversary A, we can implement D to completely fool A. This time, Lemma 2.17 says that if D is bounded samplable relative to a random oracle H, then for any adversary A, we can implement D to completely fool A, even when D is dependent on H and A also has access to H. The proof can be done by simple arguments using conditional probability. Due to page limitation, we defer the proof to the full version of this paper [15].

Similar to the non-relative case, if $D_1, \ldots, D_t$ are all distributions bounded samplable relative to H, and if they are conditionally independent relative to H, then it is easy to see that we can implement each oracle O_i sampled from D_i separately, which will fool any adversary which has oracle access to not only $O_1, \ldots, O_t$, but also to the random oracle H.

Let us give another lemma, to be used later. Note that the definition of the distribution $D^{\mathcal{Z}}$ and its dependence on H is given in Definition 2.3.

Lemma 2.18. *Suppose that D is an efficiently samplable distribution over $\mathcal{Y}^{\mathcal{X}}$ relative to a random oracle $H : \mathcal{A} \to \mathcal{B}$. If $\mathcal{Z}$ is any set, then $D^{\mathcal{Z}}$ is bounded samplable relative to H.*

Lemma 2.18 is generalization of the following: if D is an efficiently samplable distribution over a set $\mathcal{Y}$, then Zhandry points out in [21] that the distribution $D^{\mathcal{X}}$ can be 'constructed' for any set $\mathcal{X}$: if $\mathcal{R}$ is the randomness space for sampling D, and if $y = f(r)$ is the element of $\mathcal{Y}$ sampled using randomness $r \in \mathcal{R}$, then we can implement $O \leftarrow D^{\mathcal{X}}$ by first implementing a random function $\rho \in \mathcal{R}^{\mathcal{X}}$ and then letting $O(x) = f(\rho(x))$. In our terminology, the distribution $D^{\mathcal{X}}$ is bounded samplable.

Lemma 2.18 says that, when we form $D^{\mathcal{Z}}$ from an efficiently samplable distribution D of functions (relative to a random oracle H), the result is analogous: the distribution $D^{\mathcal{Z}}$ is bounded samplable (relative to H). The proof is similar, but the fact that we are dealing with functions, and also relative to a random oracle, makes this slightly more complex. Again, we defer the proof to the full version of this paper [15].

3 Relative Oracle Indistinguishability of Functions

In this paper, we are primarily interested in distributions of functions. We also consider the case where these distributions may be dependent on a random oracle $H : \mathcal{A} \to \mathcal{B}$, and the adversary has access to H as well.

Zhandry [20] defines "oracle indistinguishability" of two distributions D_1, D_2 over a set $\mathcal{Y}$. We adapt this notion to our case, giving the following definitions.

Definition 3.1 (Relative indistinguishability of functions). *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D_1, D_2 be two distributions on $\mathcal{Y}^{\mathcal{X}}$, which are conditionally independent relative to H. Then, we say that D_1 and D_2 are* indistinguishable relative to H*, if for any adversary A, the distinguishing advantage*

$$\mathbf{Adv}^{\mathsf{rel\text{-}dist}}_{D_1,D_2,H}(A) := \left|\mathbf{Pr}_{O\leftarrow D_1}\left[A^{O,H}() = 1\right] - \mathbf{Pr}_{O\leftarrow D_2}\left[A^{O,H}() = 1\right]\right|$$

is negligible.

Definition 3.2 (Relative oracle indistinguishability of functions). *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D_1, D_2 be two distributions over $\mathcal{Y}^{\mathcal{X}}$, which are conditionally independent relative to H. We say that D_1 and D_2 are* oracle-indistinguishable relative to H*, if, for any set $\mathcal{Z}$, and any adversary A, we have the following:*

$$\mathbf{Adv}^{\mathsf{oracle\text{-}rel\text{-}dist}}_{D_1,D_2,\mathcal{Z},H}(A) := \left|\mathbf{Pr}_{O\leftarrow D_1^{\mathcal{Z}}}\left[A^{O,H}() = 1\right] - \mathbf{Pr}_{O\leftarrow D_2^{\mathcal{Z}}}\left[A^{O,H}() = 1\right]\right|$$

is negligible.

Note that, when $\mathcal{A}$ and $\mathcal{B}$ are singleton sets, the random oracle H is trivial, and we obtain non-relativized definitions of the above. Moreover we are only interested in the case when D_1 and D_2 are conditionally independent relative to H, which would make sense since these are definitions of indistinguishability in the quantum random oracle model.

The following is our main result regarding oracle indistinguishability.

Theorem 3.3. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D_1, D_2 be two function distributions over $\mathcal{Y}^{\mathcal{X}}$ for some $\mathcal{X}, \mathcal{Y}$. Suppose that both $D_1^{\mathcal{Z}}$ and $D_2^{\mathcal{Z}}$ are bounded samplable relative to H, for any set $\mathcal{Z}$. Further, suppose that D_1 and D_2 are conditionally independent relative to H, and indistinguishable relative to H. Then, they are oracle-indistinguishable relative to H.*

Concretely, for any adversary $A^{O,H}$ making at most q queries to O and at most q_H queries to H, we can construct an adversary $A_{\mathsf{rd}}^{O',H}$ satisfying

$$\mathbf{Adv}^{\mathsf{oracle\text{-}rel\text{-}dist}}_{D_1,D_2,\mathcal{Z},H}(A) < 12q^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{rel\text{-}dist}}_{D_1,D_2,H}(A_{\mathsf{rd}})}.$$

Moreover, $A_{\mathsf{rd}}^{O',H}$ makes at most $2q$ queries to O' and $q_H + 2(q_{e_1} + q_{e_2})q$ queries to H. Here, q_{e_i} is the maximum number of queries to H needed by one invocation to the evaluation algorithm $D_i^{\mathcal{Z}}.eval^H()$, for $i = 1, 2$, respectively.

Theorem 3.3 says that, if two function distributions are indistinguishable (relative to H), and if they satisfy some additional conditions, then they are also oracle-indistinguishable (relative to H).

Our proof of Theorem 3.3 proceeds similarly as Zhandry's proof of the corresponding result in [20]. Therefore, we are going to defer the complete proof to the full version of this paper [15], but here let us describe some outline of the proof.

To prove oracle indistinguishability of indistinguishable distributions over a set, Zhandry uses 'small-range distribution' [20], given as follows.

Definition 3.4. *Given a distribution D on $\mathcal{Y}$, we define $\mathrm{SR}_r^D(\mathcal{X})$ as the following distribution on functions $O \in \mathcal{Y}^{\mathcal{X}}$:*

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

This can be applied to a distribution D over $\mathcal{Y}^{\mathcal{X}}$: since $\mathcal{Y}^{\mathcal{X}}$ is just a set, surely we may talk about a small-range distribution for D. Let us make this explicit:

Definition 3.5. *Given a function distribution D on $\mathcal{Y}^{\mathcal{X}}$, we define the* small-range distribution $\mathrm{SR}_r^D(\mathcal{Z})$ *as the following distribution on functions $O \in \mathcal{Y}^{\mathcal{Z}\times\mathcal{X}}$:*

- *For each $i \in [r]$, sample a function $f_i \in \mathcal{Y}^{\mathcal{X}}$ according to the distribution D.*
- *For each $z \in \mathcal{Z}$, sample a uniform random $i \in [r]$ and set $O(z) = f_i$.*

Following Definition 2.3, when D depends on the random oracle H, we interpret $\mathrm{SR}_r^D(\mathcal{Z})$ as follows:

Definition 3.6. *Given a function distribution D on $\mathcal{Y}^{\mathcal{X}}$ depending on a random oracle $H : \mathcal{A} \to \mathcal{B}$, we define the* small-range distribution $\mathrm{SR}_r^D(\mathcal{Z})$ *as follows. To jointly sample O from $\mathrm{SR}_r^D(\mathcal{Z})$ and also a particular $h : \mathcal{A} \to \mathcal{B}$ as realization of the random variable H, first sample $h \leftarrow \mathcal{B}^{\mathcal{A}}$ uniform randomly, and form $D|h$. Then,*

- *For each $i \in [r]$, sample a function $f_i \in \mathcal{Y}^{\mathcal{X}}$ according to $D|h$.*
- *For each $z \in \mathcal{Z}$, sample a uniform random $i \in [r]$ and set $O(z) = f_i$.*

Then, we have the following theorem.

Theorem 3.7. *Let $H : \mathcal{A} \to \mathcal{B}$ be a random oracle, and let D be a function distribution over $\mathcal{Y}^{\mathcal{X}}$ which is not necessarily independent from H. Suppose that A is an adversary making at most q queries to an oracle $O \in \mathcal{Y}^{\mathcal{X}}$, and at most q_H queries to the random oracle H. Then, we have*

$$\left|\mathbf{Pr}_{O\leftarrow \mathrm{SR}_r^D(\mathcal{Z})}[A^{O,H}() = 1] - \mathbf{Pr}_{O\leftarrow D^{\mathcal{Z}}}[A^{O,H}() = 1]\right| < \frac{16q^3}{r},$$

for any $r > 0$, and any set $\mathcal{Z}$.

Remark 3.8. Note that Theorem 3.7 holds, whether D is bounded samplable or not. The bound in the theorem does not depend on q_H either.

Just like the corresponding result in [20], Theorem 3.7 says that the distribution $D^{\mathcal{Z}}$, which is the distribution of an exponentially many independent samples of D indexed by $\mathcal{Z}$ is, in fact, indistinguishable from similar collection of samples, this time duplicated from only r independent samples. Theorem 3.7 also says that the result holds regardless of dependence to a random oracle H. We give the complete proof in the full version of this paper [15].

In the classical cases, we can prove oracle indistinguishability of two indistinguishable distributions by a hybrid argument over the adversarial queries: even though $O \leftarrow D^{\mathcal{Z}}$ can be considered as a collection of exponentially many independent samples of D, if a classical adversary A makes q queries $z_1, \ldots, z_q$, then all A examines are $O(z_1), \ldots, O(z_q) \leftarrow D$, and these can be swapped to samples from another indistinguishable distribution D' one by one.

On the other hand, in the quantum case, each query can be superposed, so the previous approach would not work. Small-range distribution solves this: once we switch to a small-range distribution of size r, then only r independent samples from a distribution D are involved, and they can be swapped to samples from another indistinguishable distribution D' one by one, and the resulting small-range distribution can be once again switched to $D'^{\mathcal{Z}}$. Hence, the proof of Theorem 3.3 is again a standard hybrid argument, which we defer to the full version of this paper [15].

4 Security Against Key Recovery and Security Under Random Leakage

In this section, we characterize further properties about a quantum-secure PRF, which will be useful later (to establish quantum security of NMAC for example). We first show that a secure PRF is also secure against key recovery. Using this, we prove that a secure PRF is secure under random leakage as well. This further enables us to study oracle security under random leakage for PRFs.

4.1 Security of PRFs Against Key Recovery

First, we have the following theorem:

Theorem 4.1. *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF. Suppose that both the domain and the codomain of f are superpolynomially large: $|\mathcal{X}|, |\mathcal{Y}| \geq 2^{\omega(\log \lambda)}$. Then, f is also secure against key recovery.*

Concretely, for any adversary $A^{f(k,\cdot)}$ making at most q queries to $f(k,\cdot)$ with uniform random $k \leftarrow \mathcal{K}$, we can construct an adversary A_{d} that makes at most $q+1$ queries such that

$$\mathbf{Adv}_f^{\mathsf{prf\text{-}kr}}(A) \leq \mathbf{Adv}_f^{\mathsf{prf}}(A_{\mathsf{d}}) + \frac{1}{|\mathcal{Y}|} + \frac{4q}{\sqrt{|\mathcal{X}|}}.$$

Classically, it is easy to prove that a secure PRF f is also secure against key recovery: if A is a classical key recovery attacker, then using A, we can construct a PRF distinguisher B: B^O runs A^O, while answering any query of A by its own query. In the end, if A outputs a candidate k, then B uses this k to determine whether O is a true random function ρ or a PRF instance $f(k,\cdot)$, by choosing an unqueried point $z \in \mathcal{X}$ and see if

$$f(k,z) = O(z).$$

If $O(\cdot) = f(k,\cdot)$ and if A correctly found the key k, then the above equation holds. On the other hand, if $O = \rho$, then $O(z)$ is uniform random, independent from $f(k,z)$, so the probability that $f(k,z) = O(z)$ is only $1/|\mathcal{Y}|$. This difference in probability can be used to distinguish the two cases.

On the other hand, if A is a quantum adversary, the case when $O(\cdot) = f(k,\cdot)$ is essentially the same as in the classical case. However, we may not apply the classical argument when O is a truly random function since the notion of "unqueried" point no longer makes sense under quantum (e.g., uniform superposition) queries. Therefore, we need a different approach in the quantum world. We defer the proof of Theorem 4.1 to the full version of this paper [15], due to page limitation. Note that it is possible to employ a coarse counting argument to prove key-recovery security, which works against both classical and quantum attacks. But it relies on specific settings of keyspace, domain and codomain, and the bound is typically not as tight as what our strategy can prove.

4.2 Security of PRFs Under Random Leakage

We show next that random leakage of the PRF key does not compromise the security of a PRF.

Theorem 4.2. *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF, with $\mathcal{X}$ and $\mathcal{Y}$ superpolynomially large. Then, f is also secure under random leakage.*

Concretely, for any adversary $A^{O,H}$ making at most q queries to O and q_H queries to H, we can construct adversaries A_{kr} and A_{d} such that

$$\mathbf{Adv}_f^{\mathsf{prf\text{-}rl}}(A) \le 2q_H\sqrt{\mathbf{Adv}_f^{\mathsf{prf\text{-}kr}}(A_{\mathsf{kr}})} + \mathbf{Adv}_f^{\mathsf{prf}}(A_{\mathsf{d}}).$$

Here, both A_{kr} and A_{d} make at most q oracle queries.

To prove Theorem 4.2, we are going to use the following lemma of Unruh.

Lemma 4.3 (One-Way to Hiding Lemma of) [17]. *Let $H : \mathcal{X} \to \mathcal{Y}$ be a random oracle. Consider an adversary A making at most q queries to H. Let B be an adversary that on input x does the following: pick $i \leftarrow \{1,\ldots,q\}$ and $y \leftarrow \mathcal{Y}$, run $A^H(x,y)$ until (just before) the ith query, then measure the ith query in the computational basis, and output the outcome. (When A makes less than i queries, B outputs $\perp \notin \mathcal{X}$.) Then, we have*

$$\left|\mathbf{Pr}_{x\leftarrow\mathcal{X}}[A^H(x,H(x))=1] - \mathbf{Pr}_{\substack{x\leftarrow\mathcal{X},\\ y\leftarrow\mathcal{Y}}}[A^H(x,y)=1]\right| \le 2q\sqrt{\mathbf{Pr}_{x\leftarrow\mathcal{X}}[B^H(x)=x]}.$$

Now, we are ready to prove Theorem 4.2:

Proof. Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF. To show that f is secure under random leakage, we need to show that for any set $\mathcal{W}$ and any adversary A, the advantage $\mathbf{Adv}_f^{\mathsf{prf\text{-}rl}}(A)$ is negligible. Suppose A makes at most q queries to O, and at most q_H queries to H. Then,

$$\begin{aligned}\mathbf{Adv}_f^{\mathsf{prf\text{-}rl}}(A) &= \left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{\rho,H}(w) = 1]\right| \\ &\le \left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{f(k,\cdot),H}(w) = 1]\right| \\ &\quad + \left|\mathbf{Pr}[A^{f(k,\cdot),H}(w) = 1] - \mathbf{Pr}[A^{\rho,H}(w) = 1]\right|.\end{aligned}$$

It suffices to bound both terms. First, let us bound

$$\left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{f(k,\cdot),H}(w) = 1]\right|.$$

Let us define the algorithm $A_1^H(k, w)$ as follows: it runs $A^{O,H}(w)$ while any H-query is answered by H-query of A_1 itself, and any O-query $|x\rangle$ is answered by $|x\rangle|f(k, x)\rangle$. And when $A^{O,H}(w)$ eventually halts with an output v, $A_1^H(k, w)$ outputs v and halts.

So,

$$\begin{aligned}A_1^H(k, H(k)) &= A^{f(k,\cdot),H}(H(k)), \\ A_1^H(k, w) &= A^{f(k,\cdot),H}(w).\end{aligned}$$

From Lemma 4.3, we have

$$\begin{aligned}&\left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{f(k,\cdot),H}(w) = 1]\right| \\ &= \left|\mathbf{Pr}[A_1^H(k, H(k)) = 1] - \mathbf{Pr}[A_1^H(k, w) = 1]\right| \\ &\le 2q_H\sqrt{\mathbf{Pr}[B_1^H(k) = k]},\end{aligned}$$

where the algorithm $B_1^H(k)$ can be described as follows: B_1 picks $i \leftarrow \{1, \ldots, q_H\}$, $w \leftarrow \mathcal{W}$, and runs $A_1^H(k, w) = A^{f(k,\cdot),H}(w)$ until the ith H-query, then measure the ith query and output the outcome.

Now, using A, we construct an adversary A_{kr} mounting key recovery attack on f. The algorithm A_{kr} has oracle access to $f(k, \cdot)$ for uniform random $k \leftarrow \mathcal{K}$, and A_{kr} works as follows: A_{kr} picks $i \leftarrow \{1, \ldots, q_H\}$, $w \leftarrow \mathcal{W}$, and runs $A^{f(k,\cdot),H}(w)$, while implementing $H : \mathcal{K} \to \mathcal{W}$ by a $2q_H$-wise independent function, until the ith H-query, then measure the ith query and output the outcome.

By construction, we have $\mathbf{Pr}[A_{\mathsf{kr}}^{f(k,\cdot)}() = k] = \mathbf{Pr}[B_1^H(k) = k]$. Therefore,

$$\left|\mathbf{Pr}[A^{f(k,\cdot),H}(H(k)) = 1] - \mathbf{Pr}[A^{f(k,\cdot),H}(w) = 1]\right| \le 2q_H\sqrt{\mathbf{Adv}_f^{\mathsf{prf\text{-}kr}}(A_{\mathsf{kr}})}.$$

Note that the adversary A_{kr} makes at most q queries to its oracle $f(k, \cdot)$.

Next, let us bound

$$\left|\mathbf{Pr}[A^{f(k,\cdot),H}(w)=1]-\mathbf{Pr}[A^{\rho,H}(w)=1]\right|.$$

This is straightforward: using A, we construct an adversary A_{d} attacking PRF security of f. The algorithm A_{d} has oracle access to O, which can be $f(k,\cdot)$ or a true random function ρ. Now, the algorithm A_{d} works as follows: A_{d} picks $w \leftarrow \mathcal{W}$, and runs $A^{O,H}(w)$, answering any O-query of A by an O-query of itself, and implementing $H : \mathcal{K} \to \mathcal{W}$ by a $2q_H$-wise independent function. When A halts and outputs a value v eventually, A_{d} also halts and outputs v.

By construction, we have

$$\begin{aligned}&\left|\mathbf{Pr}[A^{f(k,\cdot),H}(w)=1]-\mathbf{Pr}[A^{\rho,H}(w)=1]\right|\\&=\left|\mathbf{Pr}[A_{\mathsf{d}}^{f(k,\cdot)}()=1]-\mathbf{Pr}[A_{\mathsf{d}}^{\rho}()=1]\right|=\mathbf{Adv}_{f}^{\mathsf{prf}}(A_{\mathsf{d}}).\end{aligned}$$

The adversary A_{d} also makes at most q queries. This proves the theorem. □

4.3 Oracle-Secure PRF Under Random Leakage

In order to prove security of NMAC, we are going to use the notion of oracle security under random leakage, which we define as follows.

Definition 4.4 (Oracle security under random leakage). *Let $f : \mathcal{K}\times \mathcal{X} \to \mathcal{Y}$ be a PRF. We say that f is* oracle-secure under random leakage*, if for any sets $\mathcal{W}, \mathcal{Z}$, the following holds for any adversary A:*

$$\mathbf{Adv}^{\mathsf{os\text{-}rl}}_{f,\mathcal{Z},\mathcal{W}}(A) := \left|\mathbf{Pr}[A^{O_0,H}()=1]-\mathbf{Pr}[A^{O_1,H}()=1]\right| = negl()$$

where the oracles O_0, O_1 are defined as

$$\begin{aligned}O_0(z,x) &:= (H(\kappa(z)), f(\kappa(z),x)),\\ O_1(z,x) &:= (\rho_1(z), \rho_2(z,x)),\end{aligned}$$

and $H \leftarrow \mathcal{W}^{\mathcal{K}}, \kappa \leftarrow \mathcal{K}^{\mathcal{Z}}, \rho_1 \leftarrow \mathcal{W}^{\mathcal{Z}}, \rho_2 \leftarrow \mathcal{Y}^{\mathcal{Z}\times\mathcal{X}}$ are chosen uniform randomly, and independently.

We can show that any secure PRF f is also oracle-secure under random leakage:

Theorem 4.5. *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a secure PRF, with $\mathcal{X}$ and $\mathcal{Y}$ superpolynomially large. Then, f is also oracle-secure under random leakage.*

Concretely, for any adversary $A^{O,H}$ making at most q queries to O and at most q_H queries to H, we can construct an adversary A_{rl} such that

$$\mathbf{Adv}^{\mathsf{os\text{-}rl}}_{f,\mathcal{Z},\mathcal{W}}(A) < 12q^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{prf\text{-}rl}}_{f}(A_{\mathsf{rl}})},$$

where $A_{\mathsf{rl}}^{O,H}$ makes at most $2q$ queries to O, and $q_H + 2q$ queries to H.

Proof. Consider the distribution PRFRL_f over $(\mathcal{W} \times \mathcal{Y})^{\mathcal{X}}$ which is efficiently samplable relative to H: the randomness space $\mathcal{R}$ is just the key space $\mathcal{K}$ of f, and the evaluation algorithm is given by $\mathsf{PRFRL}_f.eval^H(k, x) := (H(k), f(k, x))$.

Consider another distribution RU over $(\mathcal{W} \times \mathcal{Y})^{\mathcal{X}}$ of f defined by $f(x) := (w, \rho(x))$, where $w \leftarrow \mathcal{W}$ and $\rho \leftarrow \mathcal{Y}^{\mathcal{X}}$ are chosen uniform randomly and independently.

It is clear that the oracle security of f under random leakage is merely restatement of the oracle indistinguishability of PRFRL_f and RU relative to H.

Since f is secure, we may use Theorem 4.2 to show that f is secure under random leakage, and this is equivalent to indistinguishability of PRFRL_f and RU relative to H. Therefore, we are going to use Theorem 3.3 to show that the two are oracle-indistinguishable relative to H.

By construction, PRFRL_f and RU are independent, and since PRFRL_f is efficiently samplable relative to H, $\mathsf{PRFRL}_f^{\mathcal{Z}}$ is bounded samplable relative to H for any set $\mathcal{Z}$, due to Lemma 2.18. Then, the only thing remaining to be proved to invoke Theorem 3.3 is that $\mathsf{RU}^{\mathcal{Z}}$ is bounded samplable for any set $\mathcal{Z}$. But this is to prove that the distribution of the oracle $O_1(z, x) = (\rho_1(z), \rho_2(z, x))$ is bounded samplable, which is now trivially true.

Concretely, for any adversary A attacking oracle security under random leakage of f making at most q queries to O and q_H queries to H, by Theorem 3.3, we have

$$\begin{aligned}\mathbf{Adv}^{\mathsf{os\text{-}rl}}_{f,\mathcal{Z},\mathcal{W}}(A) &= \mathbf{Adv}^{\mathsf{oracle\text{-}rel\text{-}dist}}_{\mathsf{PRFRL}_f,\mathsf{RU},\mathcal{Z},H}(A) \\ &< 12q^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{rel\text{-}dist}}_{\mathsf{PRFRL}_f,\mathsf{RU},H}(A_{\mathsf{rd}})},\end{aligned}$$

for some adversary $A_{\mathsf{rd}}^{O',H}$ attacking indistinguishability of PRFRL_f and RU relative to H, which makes at most $2q$ queries to O' and $q_H + 2(1+0)q$ queries to H, since 1 call to H is required to implement $\mathsf{PRFRL}_f^{\mathcal{Z}}$, and 0 calls to H are required to implement $\mathsf{RU}^{\mathcal{Z}}$.

Now we can trivially turn A_{rd} into A_{rl} attacking security of f under random leakage: $A_{\mathsf{rl}}^{O,H}(w) := A_{\mathsf{rd}}^{w\|O',H}()$, satisfying $\mathbf{Adv}^{\mathsf{rel\text{-}dist}}_{\mathsf{PRFRL}_f,\mathsf{RU},H}(A_{\mathsf{rd}}) = \mathbf{Adv}^{\mathsf{prf\text{-}rl}}_{f}(A_{\mathsf{rl}})$. Like A_{rd}, $A_{\mathsf{rl}}^{O,H}$ makes at most $2q$ queries to O and $q_H + 2q$ queries to H. □

5 Security of NMAC and Other Constructions

In this section, we prove the PRF security of cascade, NMAC, HMAC, augmented cascade, and AMAC, using ingredients we have developed so far.

5.1 Security of the Cascade

The cascade construction is not secure when queries of different block lengths are allowed. However, if we fix the total number l of blocks for all messages, then it becomes a quantum-secure PRF. Since its proof is a simpler version of that of NMAC, we only state the theorem below and refer the readers to the proof of NMAC in Sect. 5.2.

Theorem 5.1 (Security of the cascade construction). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ be a secure PRF. Then, $\mathsf{Casc}_l[f]$ is a secure PRF, for any fixed l.*

Concretely, for any adversary A of $\mathsf{Casc}_l[f]$ making at most q oracle queries, we can construct an adversary A_d making at most $4q$ oracle queries, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{Casc}_l[f]}(A) \leq 34lq^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{prf}}_f(A_\mathsf{d})}.$$

5.2 Security of NMAC

We are now ready to prove the security of NMAC as a quantum PRF.

Theorem 5.2 (NMAC security). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ be a secure PRF. Then, $\mathsf{NMAC}[f]$ is a secure PRF.*

Concretely, for any adversary A of $\mathsf{NMAC}[f]$ making at most q oracle queries, where each message has at most l message blocks, we can construct adversaries A_d, A_rl, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{NMAC}[f]}(A) \leq \mathbf{Adv}^{\mathsf{prf}}_f(A_\mathsf{d}) + 34(l+1)q^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{prf\text{-}rl}}_f(A_\mathsf{rl})}.$$

Moreover, A^O_d makes at most q queries to O, and $A^{O,H}_\mathsf{rl}$ makes at most $4q$ queries to O, and at most $6q$ queries to H.

Proof. Let A be an adversary making at most q oracle queries, where each message has at most l message blocks. We are going to define a sequence of games, where in each game, A has access to an oracle O. The only difference between the games is how the oracle O is defined.

Here's our first game N.

Game N : In this game, the oracle O is given exactly as $\mathsf{NMAC}[f]$: first, $k_1, k_2 \leftarrow \mathcal{K}$ are picked uniform randomly and independently. Then, for any message $x_1 \dots x_j$ of j-blocks ($j = 0, 1, \dots, l$), the oracle O is defined as

$$O(x_1 \dots x_j) = f(k_2, \mathsf{pad}(f(\dots f(f(k_1, x_1), x_2), \dots, x_j))).$$

In the next game G_0, the outer instance of the PRF f is swapped with a random function $H : \mathcal{K} \to \mathcal{K}$.

Game G_0: In this game, the oracle O is given as follows: first, $k \leftarrow \mathcal{K}$, $H \leftarrow \mathcal{K}^\mathcal{K}$ are picked uniform randomly and independently. Then, for any message $x_1 \dots x_j$ of j-blocks ($j = 0, 1, \dots, l$), the oracle O is defined as

$$O(x_1 \dots x_j) = H(f(\dots f(f(k, x_1), x_2), \dots, x_j)).$$

Continuing, for each $i = 1, \dots, l+1$, we define games G_i as follows.

Game G_i: In this game, the oracle O is given as follows: first, $H \leftarrow \mathcal{K}^\mathcal{K}$, $R \leftarrow \mathcal{K}^{\mathcal{X}^i}$, $R_j \leftarrow \mathcal{K}^{\mathcal{X}^j}$ (for $j = 0, \dots, i-1$) are picked uniform randomly and

independently. Then, for any message $x_1 \dots x_j$ of j-blocks ($j = 0, 1, \dots, l$), the oracle O is defined as

$$O(x_1 \dots x_j) = \begin{cases} R_j(x_1 \dots x_j) & \text{if } j = 0, \dots, i-1, \\ H(R(x_1 \dots x_i)) & \text{if } j = i, \\ H(f(\dots f(R(x_1 \dots x_i), x_{i+1}), \dots, x_j)) & \text{if } j = i+1, \dots, \text{l}. \end{cases}$$

Note that the game G_0 is in fact a special case of the above games G_i; when $i = 0$, the definition of O in the game G_i degenerates to

$$\begin{aligned} O() &= H(R()), \\ O(x_1 \dots x_j) &= H(f(\dots f(R(), x_1), \dots, x_j)), \end{aligned}$$

where $k = R() \in \mathcal{K}$ serves as the secret key in the game G_0.

Also, let's take a special look at the final game G_{l+1}: we have

$$\begin{aligned} O() &= R_0(), \\ O(x_1) &= R_1(x_1), \\ &\vdots \\ O(x_1 \dots x_l) &= R_l(x_1 \dots x_l). \end{aligned}$$

Therefore, in the game G_{l+1}, the oracle O is a true random function defined over the domain $\bigcup_{i=0}^{l} \mathcal{X}^i$.

For any game G and an adversary A, let $G(A)$ be the final output of A when A is executed in the game G. We see that

$$\begin{aligned} \mathbf{Adv}^{\mathsf{prf}}_{\mathsf{NMAC}[f]}(A) &= |\mathbf{Pr}[N(A) = 1] - \mathbf{Pr}[G_{l+1}(A) = 1]| \\ &\le |\mathbf{Pr}[N(A) = 1] - \mathbf{Pr}[G_0(A) = 1]| \\ &\qquad + |\mathbf{Pr}[G_0(A) = 1] - \mathbf{Pr}[G_{l+1}(A) = 1]| . \end{aligned}$$

First, it is easy to see that

$$|\mathbf{Pr}[N(A) = 1] - \mathbf{Pr}[G_0(A) = 1]| \le \mathbf{Adv}^{\mathsf{prf}}_{f}(A_{\mathsf{d}}),$$

for some adversary A_{d} attacking the PRF security of f; we can construct the adversary ${A_{\mathsf{d}}}^{O}$ distinguishing $f(k, \cdot)$ and $\rho \leftarrow \mathcal{K}^{\mathcal{X}}$ as follows: the adversary ${A_{\mathsf{d}}}^{O}$ picks $k' \leftarrow \mathcal{K}$, and runs A. For any query $x_1 \dots x_j$ of A (for $j \le l$), return

$$O(\mathsf{pad}(f(\dots f(f(k', x_1), x_2), \dots, x_j))).$$

When A eventually halts with an output v, A_{d} also halts with v.

Now, when $O(x) = f(k, x)$ for $k \leftarrow \mathcal{K}$, the query $x_1 \dots x_j$ is answered by $\mathsf{NMAC}[f]$, and when $O = \rho \leftarrow \mathcal{K}^{\mathcal{X}}$, then the function $H(k) := O(\mathsf{pad}(k))$ is a true random function uniformly random over $\mathcal{K}^{\mathcal{K}}$. In this case, the query of the adversary A is answered exactly like in the game G_0. In fact,

$$\mathbf{Adv}^{\mathsf{prf}}_{f}(A_{\mathsf{d}}) = |\mathbf{Pr}[N(A) = 1] - \mathbf{Pr}[G_0(A) = 1]| .$$

Next, we are going to construct an adversary $A_{\mathsf{os\text{-}rl}}$ attacking the oracle security of f under random leakage, with respect to the set $\mathcal{X}^{l-1}$ and the random oracle $H : \mathcal{K} \to \mathcal{K}$. The adversary $A_{\mathsf{os\text{-}rl}}^{O',H}$ can be described as follows.

1. $A_{\mathsf{os\text{-}rl}}$ has access to two oracles O', H, where $H : \mathcal{K} \to \mathcal{K}$ is a random oracle, and the oracle $O' : \mathcal{X}^{l-1} \times \mathcal{X} \to \mathcal{K} \times \mathcal{K}$ is either $O'_0(z,x) = (H(\kappa(z)), f(\kappa(z),x))$ or $O'_1(z,x) = (\rho_1(z), \rho_2(z,x))$, for uniform random and independent $\kappa \leftarrow \mathcal{K}^{\mathcal{X}^{l-1}}$, $\rho_1 \leftarrow \mathcal{K}^{\mathcal{X}^{l-1}}$, and $\rho_2 \leftarrow \mathcal{K}^{\mathcal{X}^{l-1}\times\mathcal{X}}$. Let us parse O' into two parts and let $O'(z,x) = (O^{(1)}(z), O^{(2)}(z,x))$. Here, $O^{(1)} : \mathcal{X}^{l-1} \to \mathcal{K}$ and $O^{(2)} : \mathcal{X}^{l-1} \times \mathcal{X} \to \mathcal{K}$.
2. $A_{\mathsf{os\text{-}rl}}$ picks a uniform random $i \leftarrow \{0, \ldots, l\}$. Also, $A_{\mathsf{os\text{-}rl}}$ implements independent uniform random functions $R_j \leftarrow \mathcal{K}^{\mathcal{X}^j}$ using bounded samplability, for $j = 0, \ldots, i-1$.
3. $A_{\mathsf{os\text{-}rl}}$ runs the adversary A until it halts, while answering any query $x_1 \ldots x_j$ of A (for $j = 0, 1, \ldots, l$) as $O(x_1 \ldots x_j)$, which is defined as follows:

$$O(x_1 \ldots x_j) = \begin{cases} R_j(x_1 \ldots x_j) & \text{if } j = 0, \ \ldots, \ i-1, \\ O^{(1)}(\mathbf{0}^{l-i-1} x_1 \ldots x_i) & \text{if } j = i, \\ H(f(\ldots f(O^{(2)}(\mathbf{0}^{l-i-1} x_1 \ldots x_i, x_{i+1}), x_{i+2}), \ldots, x_j)) & \text{if } j = i+1, \ \ldots, \ l. \end{cases}$$

   In the above, $\mathbf{0} \in \mathcal{X}$ is an arbitrarily fixed element of $\mathcal{X}$.
4. Eventually, when A halts with an output v, $A_{\mathsf{os\text{-}rl}}$ also halts, outputting v.

We remark that $A_{\mathsf{os\text{-}rl}}$ makes at most two O'-queries and two H-queries to answer one query of A (for computing and uncomputing). Since A makes at most q oracle queries, $A_{\mathsf{os\text{-}rl}}$ makes at most $2q$ queries to O' and $2q$ queries to H.

Now, conditioned on the event that a specific i is chosen on line 2, if the oracle O' is given as $O'_0(z,x) = (H(\kappa(z)), f(\kappa(z),x))$, then the oracle O is given as follows:

$$O(x_1 \ldots x_j) = \begin{cases} R_j(x_1 \ldots x_j) & \text{if } j = 0, \ \ldots, \ i-1, \\ H(\kappa(\mathbf{0}^{l-i-1} x_1 \ldots x_i)) & \text{if } j = i, \\ H(f(\ldots f(f(\kappa(\mathbf{0}^{l-i-1} x_1 \ldots x_i), x_{i+1}), x_{i+2}), \ldots, x_j)) & \text{if } j = i+1, \ \ldots, \ l. \end{cases}$$

We see that this oracle is identically distributed as the oracle in game G_i.

On the other hand, if the oracle O' is given as $O'_1(z,x) = (\rho_1(z), \rho_2(z,x))$, then we have:

$$O(x_1 \ldots x_j) = \begin{cases} R_j(x_1 \ldots x_j) & \text{if } j = 0, \ \ldots, \ i-1, \\ \rho_1(\mathbf{0}^{l-i-1} x_1 \ldots x_i) & \text{if } j = i, \\ H(f(\ldots f(\rho_2(\mathbf{0}^{l-i-1} x_1 \ldots x_i, x_{i+1}), x_{i+2}), \ldots, x_j)) & \text{if } j = i+1, \ \ldots, \ l. \end{cases}$$

We see that this oracle is identically distributed as the oracle in game G_{i+1}.

Hence for each i, we have

$$\begin{aligned}&\mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_0',H}() = 1 \mid i] - \mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_1',H}() = 1 \mid i] \\ &= \mathbf{Pr}[G_i(A) = 1] - \mathbf{Pr}[G_{i+1}(A) = 1].\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbf{Adv}_{f,\mathcal{X}^{l-1},\mathcal{K}}^{\mathsf{os\text{-}rl}}(A_{\mathsf{os\text{-}rl}}) &= \left|\mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_0',H}() = 1] - \mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_1',H}() = 1]\right| \\ &= \left|\sum_{i=0}^{l}\left(\mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_0',H}() = 1 \mid i] - \mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_1',H}() = 1 \mid i]\right)\mathbf{Pr}[i]\right| \\ &= \frac{1}{l+1}\left|\sum_{i=0}^{l}\left(\mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_0',H}() = 1 \mid i] - \mathbf{Pr}[A_{\mathsf{os\text{-}rl}}^{O_1',H}() = 1 \mid i]\right)\right| \\ &= \frac{1}{l+1}\left|\sum_{i=0}^{l}\left(\mathbf{Pr}[G_i(A) = 1] - \mathbf{Pr}[G_{i+1}(A) = 1]\right)\right| \\ &= \frac{1}{l+1}\left|\mathbf{Pr}[G_0(A) = 1] - \mathbf{Pr}[G_{l+1}(A) = 1]\right|.\end{aligned}$$

We then get

$$\left|\mathbf{Pr}[G_0(A) = 1] - \mathbf{Pr}[G_{l+1}(A) = 1]\right| = (l+1)\,\mathbf{Adv}_{f,\mathcal{X}^{l-1},\mathcal{K}}^{\mathsf{os\text{-}rl}}(A_{\mathsf{os\text{-}rl}}).$$

Now, by Theorem 4.5, we have proved the theorem. □

5.3 Security of HMAC

Here let us briefly discuss the quantum security of HMAC. The security of HMAC is formally studied in [1]. There, the security of HMAC is reduced to the security of NMAC, with an additional assumption on the compression function $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$. The assumption is that the 'dual' PRF of f, which is keyed by its data input as $f(\cdot, K)$ for $K \leftarrow \mathcal{X}$, is a secure PRF against a minor related-key attack: when the key K is chosen, the adversary may query $f(\cdot, K \oplus \mathsf{ipad})$ and $f(\cdot, K \oplus \mathsf{opad})$, and these two oracles should be indistinguishable from two independent random functions $\rho_1, \rho_2 : \mathcal{K} \to \mathcal{K}$. This reduction is still applicable to the quantum security, if we assume that the dual PRF of f is secure against the related-key attack. Hence, under this additional assumption, we can conclude that HMAC is a quantum-secure PRF.

Remark 5.3. In [12], Rötteler and Steinwandt showed that related-key attacks can be very powerful, when combined with the ability to make quantum superposed queries. Under a minor, reasonable assumption on the PRF f, if an oracle $O(\delta, x) = f(k \oplus \delta, x)$ is given to an adversary who can make quantum superposed queries, then the secret key k can be efficiently recovered. Therefore, if a quantum adversary is allowed to derive keys by XORing an arbitrary constant δ, then there exist essentially no quantum-secure PRFs against such an adversary.

However, in this case, we only need our dual PRF f to be *standard-secure* against this minor related-key attack, not quantum-secure: all we need is that the pair $(f(\mathsf{IV}, K \oplus \mathsf{ipad}), f(\mathsf{IV}, K \oplus \mathsf{opad})) \in \mathcal{K}^2$ is indistinguishable from $(k_1, k_2) \leftarrow \mathcal{K}^2$, and for this we do not need quantum security. Hence, it is a reasonable assumption to make that f is secure in the above sense.

5.4 Security of the Augmented Cascade and AMAC

We also show that the augmented cascade $\mathsf{ACSC}[f, \mathsf{Out}]$ is quantum-secure. The proof is very similar to the security proof of NMAC, but unlike the case of NMAC, we need to assume that the PRF f is secure under Out-leakage. (The security proof of NMAC also uses similar security of f under random leakage, but this can be proved from the ordinary PRF security of f.)

First, let us give the following definition, which is similar to Definition 2.7.

Definition 5.4 (Security of PRF under Out-leakage). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a PRF, and $\mathsf{Out} : \mathcal{Y} \to \mathcal{Z}$ be an unkeyed function. We say that f is* secure under Out-leakage*, if for any adversary A, we have the following:*

$$\mathbf{Adv}^{\mathsf{prf\text{-}ol}}_{f,\mathsf{Out}}(A) := \left|\mathbf{Pr}[A^{f(k,\cdot)}(\mathsf{Out}(k)) = 1] - \mathbf{Pr}[A^{\rho}(z) = 1]\right| = \mathit{negl}(),$$

where $k \leftarrow \mathcal{K}, z \leftarrow \mathcal{Z}, \rho \leftarrow \mathcal{Y}^{\mathcal{X}}$ are uniformly and independently random.

Remark 5.5. Definition 5.4 is not exactly the same as the definition given in [2]. Their version, in our notation, would be negligibility of

$$\left|\mathbf{Pr}[A^{f(k,\cdot)}(\mathsf{Out}(k)) = 1] - \mathbf{Pr}[A^{\rho}(\mathsf{Out}(k)) = 1]\right|.$$

Definition 5.4 is, in fact, two claims combined in one: the first is that $f(k, \cdot)$ remains pseudorandom even when $\mathsf{Out}(k)$ is leaked, and the second is that $\mathsf{Out}(k)$ itself is indistinguishable from a uniform random $z \leftarrow \mathcal{Z}$, for a uniform randomly chosen $k \leftarrow \mathcal{K}$. The definition in [2] is more general, but in order to obtain a PRF, eventually an output function Out close to regular should be selected, hence two definitions are essentially the same.

Now we may state the theorem showing that $\mathsf{ACSC}[f, \mathsf{Out}]$ is quantum-secure.

Theorem 5.6 (Quantum security of ACSC). *Let $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$ be a PRF, and $\mathsf{Out} : \mathcal{K} \to \mathcal{Y}$ be an unkeyed function. Suppose that f is secure under Out-leakage. Then, $\mathsf{ACSC}[f, \mathsf{Out}]$ is a secure PRF.*

Concretely, for any adversary A of $\mathsf{ACSC}[f, \mathsf{Out}]$ making at most q oracle queries, where each message has at most l message blocks, we can construct an adversary A_{ol} making at most $4q$ queries, such that

$$\mathbf{Adv}^{\mathsf{prf}}_{\mathsf{ACSC}[f,\mathsf{Out}]}(A) \le 34(l+1)q^{3/2}\sqrt{\mathbf{Adv}^{\mathsf{prf\text{-}ol}}_{f,\mathsf{Out}}(A_{\mathsf{ol}})}.$$

In the proof, we first use oracle security of f under Out-leakage to carry out the hybrid argument, and then relate the oracle security to the PRF security under Out-leakage, again using Theorem 3.3. In fact, the proof is almost identical to that of Theorem 5.2, and we will omit the proof.

Similar to the security of HMAC, the security of AMAC follows directly from the security of ACSC, with an additional assumption on the compression function $f : \mathcal{K} \times \mathcal{X} \to \mathcal{K}$, namely, that the dual of f, that is, $f(\cdot, K)$ for $K \leftarrow \mathcal{X}$, is a (standard-)secure PRF. This reduction is also applicable in the quantum security. Hence, with that additional assumption, we may conclude that AMAC is also quantum-secure.

Acknowledgements. We would like to thank the anonymous reviewers of Crypto 2017 for many helpful comments. The second author was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1601-07.

References

1. Bellare, M.: New proofs for NMAC and HMAC: security without collision-resistance. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 602–619. Springer, Heidelberg (2006). doi:10.1007/11818175_36
2. Bellare, M., Bernstein, D.J., Tessaro, S.: Hash-function based PRFs: AMAC and its multi-user security. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016. LNCS, vol. 9665, pp. 566–595. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49890-3_22
3. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1–15. Springer, Heidelberg (1996). doi:10.1007/3-540-68697-5_1
4. Bellare, M., Canetti, R., Krawczyk, H.: Pseudorandom functions revisited: the cascade construction and its concrete security. In: FOCS 1996, pp. 514–523. IEEE Computer Society (1996)
5. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38348-9_35
6. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40084-1_21
7. Gaži, P., Pietrzak, K., Rybár, M.: The exact PRF-security of NMAC and HMAC. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 113–130. Springer, Heidelberg (2014). doi:10.1007/978-3-662-44371-2_7
8. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. J. ACM **33**(4), 792–807 (1986)
9. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 207–237. Springer, Heidelberg (2016). doi:10.1007/978-3-662-53008-5_8
10. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: ISIT 2010, pp. 2682–2685. IEEE (2010)

11. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. In: ISITA 2012, pp. 312–316. IEEE (2012)
12. Rötteler, M., Steinwandt, R.: A note on quantum related-key attacks. Inf. Process. Lett. **115**(1), 40–44 (2015)
13. Santoli, T., Schaffner, C.: Using Simon's algorithm to attack symmetric-key cryptographic primitives. Quantum Inf. Comput. **17**(1&2), 65–78 (2017)
14. Song, F.: A note on quantum security for post-quantum cryptography. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 246–265. Springer, Cham (2014). doi:10.1007/978-3-319-11659-4_15
15. Song, F., Yun, A.: Quantum security of NMAC and related constructions. Cryptology ePrint Archive, Report 2017/509, full version of this paper (2017). http://eprint.iacr.org/2017/509
16. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29011-4_10
17. Unruh, D.: Revocable quantum timed-release encryption. J. ACM **62**(6), 49:1–49:76 (2015)
18. Vadhan, S.P.: Pseudorandomness. Foundations and trends® in theoretical computer science. Theoret. Comput. Sci. **7**(1–3), 1–336 (2012)
19. Watrous, J.: Zero-knowledge against quantum attacks. SIAM J. Comput. **39**(1), 25–58 (2009)
20. Zhandry, M.: How to construct quantum random functions. In: FOCS 2012, pp. 679–687. IEEE Computer Society (2012)
21. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32009-5_44