J. Cryptol. (2015) 28: 297–311
DOI: 10.1007/s00145-013-9169-2

Journal of
CRYPTOLOGY

From Non-adaptive to Adaptive Pseudorandom Functions*

Itay Berman† and Iftach Haitner†
School of Computer Science, Tel Aviv University, Tel Aviv, Israel
itayberm@post.tau.ac.il; iftachh@cs.tau.ac.il

Communicated by Jonathan Katz

Received 23 October 2012
Online publication 23 November 2013

Abstract. Unlike the standard notion of pseudorandom functions (PRF), a *non-adaptive* PRF is only required to be indistinguishable from a random function in the eyes of a *non-adaptive* distinguisher (i.e., one that prepares its oracle calls in advance). A recent line of research has studied the possibility of a *direct* construction of adaptive PRFs from non-adaptive ones, where direct means that the constructed adaptive PRF uses only few (ideally, constant number of) calls to the underlying non-adaptive PRF. Unfortunately, this study has only yielded negative results (e.g., Myers in Advances in Cryptology – EUROCRYPT 2004, pp. 189–206, 2004; Pietrzak in Advances in Cryptology – CRYPTO 2005, pp. 55–65, 2005).

We give an affirmative answer to the above question, presenting a direct construction of adaptive PRFs from non-adaptive ones. The suggested construction is extremely simple, a composition of the non-adaptive PRF with an appropriate pairwise independent hash function.

1. Introduction

A pseudorandom function family (PRF), introduced by Goldreich, Goldwasser, and Micali [11], cannot be distinguished from a family of *truly* random functions by an efficient distinguisher who is given an oracle access to a random member of the family. PRFs have an extremely important role in cryptography, allowing parties, which share a common secret key, to send secure messages, identify themselves and to authenticate messages [10,13]. In addition, they have many other applications, essentially in any setting that requires random function provided as black-box. Different PRF constructions are known in the literature, whose security is based on different hardness assumption. Constructions relevant to this work are those based on the existence of pseudorandom

* A preliminary version appeared in [2].

† Research supported by Check Point Institute for Information Security and ISF grant 1076/11.

© International Association for Cryptologic Research 2013

generators [11] (and thus on the existence of one-way functions [12]), and on, the so called, synthesizers [19].

In this work we study the question of constructing (adaptive) PRFs from *non-adaptive* PRFs. The latter primitive is a (weaker) variant of the standard PRF we mentioned above, whose security is only guaranteed to hold against non-adaptive distinguishers (i.e., ones that "write" all their queries before the first oracle call). Since a non-adaptive PRF can be easily cast as a pseudorandom generator or as a synthesizer, [11,19] tell us how to construct (adaptive) PRF from a non-adaptive one. If the input length of the underlying non-adaptive PRF is n, then the resulting length-preserving (i.e., mapping strings of length n to strings of the same length) (adaptive) PRF of both of these constructions makes $\Theta(n)$ calls to the underlying non-adaptive PRF. [11] was later improved to show that $w(\log n)$ sequential calls are sufficient (cf., [8, Sect. 3.8.4, Exe. 30]).

A recent line of work has tried to figure out whether more efficient reductions from adaptive to non-adaptive PRF's are likely to exist. In a sequence of works [5,18,20,21], it was shown that several "natural" approaches (e.g., composition or XORing members of the non-adaptive family with itself) are unlikely to work. See more in Sect. 1.3.

1.1. *Our Result*

We show that a simple composition of a non-adaptive PRF with an appropriate pairwise independent hash function, yields an adaptive PRF. To state our result more formally, we use the following definitions: a function family $\mathcal{F}$ is $T = T(n)$-adaptive PRF, if no distinguisher of running time at most T, can tell a random member of $\mathcal{F}$ from a random function with advantage larger than $1/T$. The family $\mathcal{F}$ is T-*non*-adaptive PRF, if the above is only guarantee to hold against non-adaptive distinguishers. Given two function families $\mathcal{F}_1$ and $\mathcal{F}_2$, we let $\mathcal{F}_1 \circ \mathcal{F}_2$ [resp., $\mathcal{F}_1 \oplus \mathcal{F}_2$] be the function family whose members are all pairs $(f, g) \in \mathcal{F}_1 \times \mathcal{F}_2$, and the action $(f, g)(x)$ is defined as $f(g(x))$ [resp., $f(x) \oplus g(x)$]. We prove the following statements (see Sect. 3 for the formal statements).

Theorem 1.1 (Informal). *Let $\mathcal{H}$ be an efficient pairwise-independent function family mapping strings of length n to $[T(n)]_{\{0,1\}^n}$, where $[T]_{\{0,1\}^n}$ is the first T elements (in lexicographic order) of $\{0,1\}^n$ and let $\mathcal{F}$ be a $(p \cdot T)$-non-adaptive PRF with input length n, where $p \in$ poly is a function of the evaluating time of $\mathcal{H}$. Then $\mathcal{F} \circ \mathcal{H}$ is a $(\sqrt[3]{T})$-adaptive PRF.*

For instance, assuming that $\mathcal{F}$ is a $(p(n) \cdot 2^{cn})$-non-adaptive PRF and that $\mathcal{H}$ maps strings of length n to $[2^{cn}]_{\{0,1\}^n}$, Theorem 1.1 yields the result that $\mathcal{F} \circ \mathcal{H}$ is a $(2^{\frac{cn}{3}})$-adaptive PRF.

Theorem 1.1 is only useful, however, for polynomial-time computable T's (in this case, the family $\mathcal{H}$ assumed by the theorem exists, see Sect. 2.2.2). Unfortunately, in the important case where $\mathcal{F}$ is only assumed to be polynomially secure non-adaptive PRF, no useful polynomial-time computable T is guaranteed to exist.[1]

[1] Clearly $\mathcal{F}$ is p-non-adaptive PRF for any $p \in$ poly, but applying Theorem 1.1 with $T \in$ poly, does not yield a polynomially secure adaptive PRF.

We suggest two different solutions for handling polynomially secure PRFs. In Appendix A we observe (following Bellare [1]) that a polynomially secure non-adaptive PRF is a T-non-adaptive PRF for some $T \in n^{\omega(1)}$. Since this T can be assumed without loss of generality to be a power of two, Theorem 1.1 yields a *non-uniform* (uses $\omega(1)$-bit advice) polynomially secure adaptive PRF that makes a single call to the underlying non-adaptive PRF. Our second solution is to use the following "combiner", to construct a (uniform) adaptively secure PRF, which makes $\omega(1)$ parallel calls to the underlying non-adaptive PRF. Intuitively, the combiner apply Theorem 1.1 on the polynomially secure non-adaptive PRF for $\omega(1)$ times, with respect to $T = n, n^2, n^3, \ldots, n^{\omega(1)}$, and finally XOR the outputs of these functions. Theorem 1.1 guarantees the security of at least one of the XORed functions, and thus security of the combiner follows.

Corollary 1.2 (Informal). *Let $\mathcal{F}$ be a polynomially secure non-adaptive PRF with input length n, let $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be an efficient pairwise-independent length-preserving function family and let $k(n) \in \omega(1)$ be polynomial-time computable function.*

For $n \in \mathbb{N}$ and $i \in [n]$, let $\widehat{\mathcal{H}}_n^i$ be the function family $\widehat{\mathcal{H}}_n^i = \{\widehat{h} \colon h \in \mathcal{H}_n\}$, where $\widehat{h}(x) = 0^{n-i}||h(x)_{1,\ldots,i}$ ('$||$' stands for string concatenation). Then the ensemble $\{\bigoplus_{i\in[k(n)]}(\mathcal{F}_n \circ \widehat{\mathcal{H}}_n^{\lfloor i\cdot\log n\rfloor})\}_{n\in\mathbb{N}}$ is a polynomially secure adaptive PRF.

1.2. *Proof Idea*

To prove Theorem 1.1 we first show that $\mathcal{F} \circ \mathcal{H}$ is indistinguishable from $\Pi \circ \mathcal{H}$, where Π being the set of *all* functions from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$ (letting $\ell(n)$ be $\mathcal{F}$'s output length), and then conclude the proof by showing that $\Pi \circ \mathcal{H}$ is indistinguishable from Π.

$\mathcal{F} \circ \mathcal{H}$ is indistinguishable from $\Pi \circ \mathcal{H}$. Let D be (a possibly adaptive) algorithm of running time $T(n)$, which distinguishes $\mathcal{F} \circ \mathcal{H}$ from $\Pi \circ \mathcal{H}$ with advantage $\varepsilon(n)$. We use D to build a *non-adaptive* distinguisher $\widehat{\mathsf{D}}$ of running time $p(n) \cdot T(n)$, which distinguishes $\mathcal{F}$ from Π with advantage $\varepsilon(n)$. Given an oracle access to a function ϕ, the distinguisher $\widehat{\mathsf{D}}^{\phi}(1^n)$ first queries ϕ on *all* the elements of $[T(n)]_{\{0,1\}^n}$. Next it chooses at uniform $h \in \mathcal{H}$, and uses the stored answers to its queries, to emulate $\mathsf{D}^{\phi\circ h}(1^n)$.

Since $\widehat{\mathsf{D}}$ runs in time $p(n) \cdot T(n)$, for some large enough $p \in \text{poly}$, makes *non-adaptive* queries, and distinguishes $\mathcal{F}$ from Π with advantage $\varepsilon(n)$, the assumed security of $\mathcal{F}$ yields the result that $\varepsilon(n) < \frac{1}{p(n)\cdot T(n)}$.

$\Pi \circ \mathcal{H}$ is indistinguishable from Π. We prove that $\Pi \circ \mathcal{H}$ is *statistically* indistinguishable from Π. Namely, even an unbounded distinguisher (that makes bounded number of calls) cannot distinguish between the families. The idea of the proof is fairly simple. Let D be an s-query algorithm trying to distinguish between $\Pi \circ \mathcal{H}$ and Π. We first note that the distinguishing advantage of D is bounded by its probability of finding a collision in a random $\phi \in \Pi \circ \mathcal{H}$ (in case no collision occurs, ϕ's output is uniform). We next argue that in order to find a collision in ϕ, the distinguisher D gains nothing from being adaptive. Indeed, assuming that D found no collision until the ith call, then it has only learned that h does not collide on these first i queries. Therefore, a random (or even a constant) query as the $(i+1)$ call, has the same chance to yield a collision, as any other query has. Hence, we assume without loss of generality that D is non-adaptive, and use the pairwise independence of $\mathcal{H}$ to conclude that D's

probability in finding a collision, and thus its distinguishing advantage, is bounded by $s(n)^2/2T(n)$.

Combining the above two observations, we conclude that an adaptive distinguisher whose running time is bounded by $\sqrt[3]{T(n)}$, cannot distinguish $\mathcal{F} \circ \mathcal{H}$ from Π (i.e., from a random function) with an advantage better than $\frac{T(n)^{\frac{2}{3}}}{2T(n)} + \frac{1}{p(n)T(n)} \leq 1/\sqrt[3]{T(n)}$. Namely, $\mathcal{F} \circ \mathcal{H}$ is a $(\sqrt[3]{T(n)})$-adaptive PRF.

1.3. *Related Work*

Maurer and Pietrzak [15] were the first to consider the question of building adaptive PRFs from non-adaptive ones. They showed that in the *information theoretic* model, a self composition of a non-adaptive PRF *does* yield an adaptive PRF.[2]

In contrast, the situation in the *computational model* (which we consider here) seems very different: Myers [18] proved that it is impossible to reprove the result of [15] via fully-black-box reductions. Pietrzak [20] showed that under the Decisional Diffie–Hellman (DDH) assumption, composition does not imply adaptive security. In [21] he showed that the existence of non-adaptive PRFs whose composition is not adaptively secure, yields the result that a key-agreement protocol exists. Finally, Cho, Lee, and Ostrovsky [5] generalized [21] by proving that composition of two non-adaptive PRFs is not adaptively secure iff a (uniform transcript) key agreement protocol exists. We mention that [5,18,20], and in a sense also [15], hold also with respect to XORing of the non-adaptive families.

A parallel line of work studied the notion and uses of *weak* PRFs (which are secure only against random quires). Damgård and Nielsen [6] showed how to use weak PRF directly in order to achieve private-key encryption, circumvent the need to first construct a PRF. Maurer and Sjödin [16] improved [6]'s construction and also gave a construction of PRF from weak PRF. Maurer and Tessaro [17] showed how to construct a PRF from an even weaker primitive—*constant-query* weak PRF. Recently, [7,14] studied weak PRFs also in the context of message authentication codes.

In a very recent subsequent work, Berman et al. [3] used more sophisticated hashing technique to improve the result presented here. Specifically, [3] use the so called Cuckoo hashing to give an optimal version of Theorem 1.1—the resulting PRF is an $O(T)$-adaptive PRF (however, [3] does not achieve qualitative improvement over Corollary 1.2—it still requires $\omega(1)$ calls to the underlying non-adaptive polynomially secure PRF to get an adaptive polynomially secure PRF).

2. Preliminaries

2.1. *Notations*

All logarithms considered here are in base two. We let '||' denote string concatenation. We use calligraphic letters to denote sets, uppercase for random variables, and lowercase for values. For an integer t, we let $[t] = \{1, \ldots, t\}$, and for a set $\mathcal{S} \subseteq \{0, 1\}^*$ with

[2] Specifically, assuming that the non-adaptive PRF is (Q, ε)-non-adaptively secure, no Q-query non-adaptive algorithm distinguishes it from random with advantage larger than ε, then the resulting PRF is $(Q, \varepsilon(1 + \ln \frac{1}{\varepsilon}))$-adaptively secure.

$|\mathcal{S}| \geq t$, we let $[t]_{\mathcal{S}}$ be the first t elements (in increasing lexicographic order) of $\mathcal{S}$. We let poly denote the set all polynomials, and let PPTM denote the set of probabilistic polynomial-time algorithms (i.e., Turing machines) that run in *strictly* polynomial time. A function $\mu\colon \mathbb{N} \to [0, 1]$ is *negligible*, denoted $\mu(n) = \mathrm{neg}(n)$, if $\mu(n) < 1/p(n)$ for every $p \in$ poly and large enough n.

Given a random variable X, we write $X(x)$ to denote $\Pr[X = x]$, and write $x \leftarrow X$ to indicate that x is selected according to X. Similarly, given a finite set $\mathcal{S}$, we let $s \leftarrow \mathcal{S}$ denote that s is selected according to the uniform distribution on $\mathcal{S}$. The *statistical distance* of two distributions P and Q over a finite set $\mathcal{U}$, denoted as $\mathrm{SD}(P, Q)$, is defined as $\max_{\mathcal{S}\subseteq\mathcal{U}} |P(\mathcal{S}) - Q(\mathcal{S})| = \frac{1}{2}\sum_{u\in\mathcal{U}} |P(u) - Q(u)|$.

2.2. *Ensemble of Function Families*

Let $\mathcal{F} = \{\mathcal{F}_n\colon \mathcal{D}_n \mapsto \mathcal{R}_n\}_{n\in\mathbb{N}}$ stands for an ensemble of function families, where each $f \in \mathcal{F}_n$ has domain $\mathcal{D}_n$ and its range contained in $\mathcal{R}_n$. Such an ensemble is *length preserving*, if $\mathcal{D}_n = \mathcal{R}_n = \{0, 1\}^n$ for every n.

Definition 2.1 (Efficient function family ensembles). A function family ensemble $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is efficient, if the following hold:

Samplable. $\mathcal{F}$ is samplable in polynomial time: there exists a PPTM that, given 1^n, outputs (the description of) a uniform element in $\mathcal{F}_n$.

Efficient. There exists a polynomial-time algorithm that given $x \in \{0, 1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

2.2.1. *Operating on Function Families*

Definition 2.2 (Composition of function families). Let $\mathcal{F}^1 = \{\mathcal{F}^1_n\colon \mathcal{D}^1_n \mapsto \mathcal{R}^1_n\}_{n\in\mathbb{N}}$ and $\mathcal{F}^2 = \{\mathcal{F}^2_n\colon \mathcal{D}^2_n \mapsto \mathcal{R}^2_n\}_{n\in\mathbb{N}}$ be two ensembles of function families with $\mathcal{R}^1_n \subseteq \mathcal{D}^2_n$ for every n. We define the composition of $\mathcal{F}^1$ with $\mathcal{F}^2$ as $\mathcal{F}^2 \circ \mathcal{F}^1 = \{\mathcal{F}^2_n \circ \mathcal{F}^1_n\colon \mathcal{D}^1_n \mapsto \mathcal{R}^2_n\}_{n\in\mathbb{N}}$, where $\mathcal{F}^2_n \circ \mathcal{F}^1_n = \{(f_2, f_1) \in \mathcal{F}^2_n \times \mathcal{F}^1_n\}$, and $(f_2, f_1)(x) := f_2(f_1(x))$.

Definition 2.3 (XOR of function families). Let $\mathcal{F}^1 = \{\mathcal{F}^1_n\colon \mathcal{D}^1_n \mapsto \mathcal{R}^1_n\}_{n\in\mathbb{N}}$ and $\mathcal{F}^2 = \{\mathcal{F}^2_n\colon \mathcal{D}^2_n \mapsto \mathcal{R}^2_n\}_{n\in\mathbb{N}}$ be two ensembles of function families with $\mathcal{R}^1_n, \mathcal{R}^2_n \subseteq \{0, 1\}^{\ell(n)}$ for every n. We define the XOR of $\mathcal{F}^1$ with $\mathcal{F}^2$ as $\mathcal{F}^2 \bigoplus \mathcal{F}^1 = \{\mathcal{F}^2_n \bigoplus \mathcal{F}^1_n\colon \mathcal{D}^1_n \cap \mathcal{D}^2_n \mapsto \{0, 1\}^{\ell(n)}\}_{n\in\mathbb{N}}$, where $\mathcal{F}^2_n \bigoplus \mathcal{F}^1_n = \{(f_2, f_1) \in \mathcal{F}^2_n \times \mathcal{F}^1_n\}$, and $(f_2, f_1)(x) := f_2(x) \oplus f_1(x)$.

2.2.2. *Pairwise Independent Hashing*

Definition 2.4 (Pairwise independent families). A function family $\mathcal{H} = \{h\colon \mathcal{D} \mapsto \mathcal{R}\}$ is pairwise independent (with respect to $\mathcal{D}$ and $\mathcal{R}$), if

$$\Pr_{h\leftarrow\mathcal{H}}\big[h(x_1) = y_1 \wedge h(x_2) = y_2\big] = \frac{1}{|\mathcal{R}|^2},$$

for every distinct $x_1, x_2 \in \mathcal{D}$ and every $y_1, y_2 \in \mathcal{R}$.

For every $\ell \in \text{poly}$, the existence of efficient pairwise independent family ensembles mapping strings of length n to strings of length $\ell(n)$ is well known [4]. In this paper we use efficient pairwise independent function family ensembles mapping strings of length n to the set $[T(n)]_{\{0,1\}^n}$, where $T(n) \leq 2^n$ and is without loss of generality a power of two.[3] Let $\mathcal{H}$ be an efficient length-preserving pairwise independent function family ensemble and assume that $t(n) := \log T(n)$ is polynomial-time computable. Then the function family $\widehat{\mathcal{H}} = \{\widehat{\mathcal{H}}_n = \{\widehat{h} \colon h \in \mathcal{H}, \widehat{h}(x) = 0^{n-t(n)} || h(x)_{1,\ldots,t(n)}\}\}$, is an efficient pairwise independent function family ensemble, mapping strings of length n to the set $[T(n)]_{\{0,1\}^n}$.

2.2.3. *Pseudorandom Functions*

Definition 2.5 (Pseudorandom functions). An efficient function family ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ is a $(T(n), \varepsilon(n))$-adaptive PRF, if for every oracle-aided algorithm (distinguisher) D of running time $T(n)$ and large enough n, we have

$$\left|\Pr_{f \leftarrow \mathcal{F}_n}\left[\mathsf{D}^f\left(1^n\right) = 1\right] - \Pr_{\pi \leftarrow \Pi_n}\left[\mathsf{D}^\pi\left(1^n\right) = 1\right]\right| \leq \varepsilon(n),$$

where Π_n is the set of all functions from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$. If we limit D above to be non-adaptive (i.e., it has to write all his oracle calls before making the first call), then $\mathcal{F}$ is called $(T(n), \varepsilon(n))$-non-adaptive PRF.

The ensemble $\mathcal{F}$ is a T-adaptive PRF, if it is a $(T, 1/T)$-adaptive PRF according to the above definition. It is polynomially secure adaptive PRF (for short, adaptive PRF), if it is a p-adaptive PRF for every $p \in \text{poly}$. Finally, it is super-polynomially secure adaptive PRF, if it is a T-adaptive PRF for some $T(n) \in n^{\omega(1)}$. The same conventions are also used for non-adaptive PRFs.

Clearly, a super-polynomially secure PRF is also polynomially secure. In Appendix A we prove that the converse is also true: a polynomially secure PRF is also super-polynomially secure PRF.

3. Our Construction

In this section we present the main contribution of this paper—a direct construction of an adaptive pseudorandom function family from a non-adaptive one.

Theorem 3.1 (Restatement of Theorem 1.1). *Let T be a polynomial-time computable integer function, let $\mathcal{H} = \{\mathcal{H}_n \colon \{0,1\}^n \mapsto [T(n)]_{\{0,1\}^n}\}_{n\in\mathbb{N}}$ be an efficient pairwise independent function family ensemble, and let $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$ be a $(p \cdot T, \varepsilon)$-non-adaptive PRF, where $p \in \text{poly}$ is such that $p(n) \geq e_T(n) + 2e_{\mathcal{H}}(n)$, for $e_T(n)$ being the evaluation time of $T(n)$ and $e_{\mathcal{H}}(n)$ the sampling and evaluation time of $\mathcal{H}_n$. Then $\mathcal{F} \circ \mathcal{H}$ is a $(s, \varepsilon + \frac{s^2}{2T})$-adaptive PRF for every function s such that $s(n) < T(n)$ for every $n \in \mathbb{N}$.*

[3] For our applications, see Sect. 3, we can always consider $T'(n) = 2^{\lfloor \log(T(n)) \rfloor}$, which only causes us a factor of two loss in the resulting security.

Theorem 3.1 yields the following simpler statement.

Corollary 3.2. *Let T, $\mathcal{H}$ and p be as in Theorem 3.1. Assuming $\mathcal{F}$ is a $(p \cdot T)$-non-adaptive PRF, then $\mathcal{F} \circ \mathcal{H}$ is a $(\sqrt[3]{T})$-adaptive PRF.*

Proof. Applying Theorem 3.1 with respect to $s(n) = \sqrt[3]{T(n)}$ and $\varepsilon(n) = \frac{1}{p(n)T(n)}$, yields the result that $\mathcal{F} \circ \mathcal{H}$ is a $(s(n), \frac{1}{p(n)T(n)} + \frac{s(n)^2}{2T(n)})$-adaptive PRF. Since $\frac{1}{p(n)T(n)} < \frac{1}{2s(n)}$ and $\frac{s(n)^2}{2T(n)} \leq \frac{1}{2s(n)}$, it follows that $\mathcal{F} \circ \mathcal{H}$ is a $(s, 1/s)$-adaptive PRF. □

To prove Theorem 3.1, we use the (non efficient) function family ensemble $\Pi \circ \mathcal{H}$, where $\Pi = \Pi_\ell$ (i.e., the ensemble of all functions from $\{0,1\}^n$ to $\{0,1\}^\ell$), and $\ell = \ell(n)$ is the output length of $\mathcal{F}$. We first show that $\mathcal{F} \circ \mathcal{H}$ is *computationally* indistinguishable from $\Pi \circ \mathcal{H}$, and complete the proof by showing that $\Pi \circ \mathcal{H}$ is *statistically* indistinguishable from Π.

3.1. $\mathcal{F} \circ \mathcal{H}$ is Computationally Indistinguishable from $\Pi \circ \mathcal{H}$

Lemma 3.3. *Let T, $\mathcal{F}$, $\mathcal{H}$ and p be as in Theorem 3.1. Then for every oracle-aided distinguisher D of running time $T(n)$, there exists a* non-adaptive *oracle-aided distinguisher $\widehat{\mathsf{D}}$ of running time $p(n) \cdot T(n)$ with*

$$\begin{aligned}&\left|\Pr_{g \leftarrow \mathcal{F}_n}\left[\widehat{\mathsf{D}}^g\left(1^n\right) = 1\right] - \Pr_{g \leftarrow \Pi_n}\left[\widehat{\mathsf{D}}^g\left(1^n\right) = 1\right]\right| \\ &= \left|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right] - \Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right]\right|\end{aligned}$$

for every $n \in \mathbb{N}$, where Π_n is the set of all functions from $\{0,1\}^n$ to $\{0,1\}^{\ell(n)}$.

In particular, the pseudorandomness of $\mathcal{F}$ yields the result that $\mathcal{F} \circ \mathcal{H}$ is computationally indistinguishable from the ensemble $\Pi \circ \mathcal{H}$ by an adaptive distinguisher of running time T.

Proof. The distinguisher $\widehat{\mathsf{D}}$ is defined as follows:

Algorithm 3.4 ($\widehat{\mathsf{D}}$).

Input: 1^n.
Oracle: *a function ϕ over $\{0,1\}^n$.*

1. *Compute $\phi(x)$ for every $x \in [T(n)]_{\{0,1\}^n}$.*
2. *Set $g = \phi \circ h$, where h is uniformly chosen in $\mathcal{H}_n$.*
3. *Emulate $\mathsf{D}^g(1^n)$: answer a query x to ϕ made by D with $g(x)$, using the information obtained in Step 1.*

Note that $\widehat{\mathsf{D}}$ makes $T(n)$ *non-adaptive* queries to ϕ, and it can be implemented to run in time $e_T(n) + e_{\mathcal{H}}(n) + T(n) + e_{\mathcal{H}}(n)T(n) \leq p(n)T(n)$. We conclude the proof by observing that in case ϕ is uniformly drawn from $\mathcal{F}_n$, the emulation of D done in $\widehat{\mathsf{D}}^\phi$ is identical to a random execution of D^g with $g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n$. Similarly, in case ϕ is uniformly drawn from Π_n, the emulation is identical to a random execution of D^π with $\pi \leftarrow \Pi_n \circ \mathcal{H}_n$. □

3.2. $\Pi \circ \mathcal{H}$ is Statistically Indistinguishable from Π

The following lemma is commonly used for proving the security of hash-based MACs (cf., [9, Proposition 6.3.6]), yet for completeness we give a full proof below.

Lemma 3.5. *Let n, T be integers with $T \leq 2^n$, and let $\mathcal{H}$ be a pairwise independent function family mapping string of length n to $[T]_{\{0,1\}^n}$. Let D be an (unbounded) s-query oracle-aided algorithm (i.e., making at most s oracle queries), then*

$$\left|\Pr_{g \leftarrow \Pi \circ \mathcal{H}}\left[\mathsf{D}^g = 1\right] - \Pr_{\pi \leftarrow \Pi}\left[\mathsf{D}^\pi = 1\right]\right| \leq s^2/2T,$$

where Π is the set of all functions from $\{0,1\}^n$ to $\{0,1\}^\ell$ (for some $\ell \in \mathbb{N}$).

Proof. We assume for simplicity that D is deterministic (the reduction to the randomized case is standard) and makes exactly s valid (i.e., inside $\{0,1\}^n$) distinct queries, and let $\Omega = (\{0,1\}^\ell)^s$. Consider the following random process:

Algorithm 3.6.

1. *Emulate D, while answering the ith query q_i with a uniformly chosen $a_i \in \{0,1\}^\ell$.*
   Set $\overline{q} = (q_1, \ldots, q_s)$ and $\overline{a} = (a_1, \ldots, a_s)$.
2. *Choose $h \leftarrow \mathcal{H}$.*
3. *Emulate D again, while answering the ith query q'_i with $a'_i = a_i$ (the same a_i from Step 1), if $h(q'_i) \notin \{h(q'_j)\}_{j \in [i-1]}$, and with $a'_i = a_j$, if $h(q'_i) = h(q'_j)$ for some $j \in [i-1]$.*
   Set $\overline{q'} = (q'_1, \ldots, q'_s)$ and $\overline{a'} = (a'_1, \ldots, a'_s)$.

Let $\overline{A}$, $\overline{Q}$, $\overline{A'}$, $\overline{Q'}$ and H be the (jointly distributed) random variables induced by the values of $\overline{q}$, $\overline{a}$, $\overline{q'}$, $\overline{a'}$, and h, respectively, in a random execution of the above process. It is not hard to verify that $\overline{A}$ is distributed the same as the oracle answers in a random execution of D^π with $\pi \leftarrow \Pi$, and that $\overline{A'}$ is distributed the same as the oracle answers in a random execution of D^g with $g \leftarrow \Pi \circ \mathcal{H}$. Hence, for proving Lemma 3.5, it suffices to bound the statistical distance between $\overline{A}$ and $\overline{A'}$.

Let Coll be the event that $H(\overline{Q}_i) = H(\overline{Q}_j)$ for some $i \neq j \in [s]$. Since the queries and answers in both emulations of Algorithm 3.6 are the same until a collision with respect to H occurs, it follows that

$$\Pr[\overline{A} \neq \overline{A'}] \leq \Pr[\mathrm{Coll}] \tag{1}$$

On the other hand, since H is chosen *after* $\overline{Q}$ is set, the pairwise independent of $\mathcal{H}$ yields the result that

$$\Pr[\mathrm{Coll}] \leq s^2/2T, \tag{2}$$

and therefore $\Pr[\overline{A} \neq \overline{A'}] \leq s^2/2T$. It follows that $\Pr[\overline{A} \in C] \leq \Pr[\overline{A'} \in C] + s^2/2T$ for every $C \subseteq \Omega$, yielding the result that $\mathrm{SD}(\overline{A}, \overline{A'}) \leq s^2/2T$. □

3.3. *Putting It Together*

We are now finally ready to prove Theorem 3.1.

Proof of Theorem 3.1. Let D be an oracle-aided algorithm of running time s with $s(n) < T(n)$. Claim 3.3 yields the result that

$$\left|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right] - \Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right]\right| \leq \varepsilon(n)$$

for large enough n, where Lemma 3.5 yields the result that

$$\left|\Pr_{g \leftarrow \Pi_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right] - \Pr_{\pi \leftarrow \Pi_n}\left[\mathsf{D}^\pi\left(1^n\right) = 1\right]\right| \leq s(n)^2/2T(n)$$

for every $n \in \mathbb{N}$. Hence, the triangle inequality yields the result that

$$\left|\Pr_{g \leftarrow \mathcal{F}_n \circ \mathcal{H}_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right] - \Pr_{\pi \leftarrow \Pi_n}\left[\mathsf{D}^\pi\left(1^n\right) = 1\right]\right| \leq \varepsilon(n) + s(n)^2/2T(n)$$

for large enough n, as requested. □

3.4. *Handling Unknown Security*

Corollary 3.2 is useful when the function T, which determines the security of the underlying non-adaptive PRF, is *efficiently computable* (or when considering non-uniform PRF constructions, see Sect. 1.1) and *known at construction time*. In this section we show how to handle the case where T is *not known* at construction time, and the case of polynomially secure non-adaptive PRF.

We use the following PRF "combiner".

Definition 3.7. Let $\mathcal{H}$ be a function family into $\{0,1\}^n$. For $i \in [n]$, let $\widehat{\mathcal{H}}^i$ be the function family into $[2^i]_{\{0,1\}^n}$ such that $\widehat{\mathcal{H}}^i = \{\widehat{h} \colon h \in \mathcal{H}\}$, for $\widehat{h}(x) := 0^{n-i}||h(x)_{1,\ldots,i}$.

Corollary 3.8. *Let $\mathcal{F}$ be a T-non-adaptive PRF sampled and evaluated in time $e_{\mathcal{F}} \in \mathrm{poly}$, let $\mathcal{H}$ be an efficient length-preserving pairwise independent function family ensemble sampled and evaluated in time $e_{\mathcal{H}} \in \mathrm{poly}$, and let $\mathcal{I}(n) \subseteq [n]$ be an index set computable (in n) in time $e_{\mathcal{I}} \in \mathrm{poly}$. Finally, let $G = \{G_n\}_{n \in \mathbb{N}}$, for $G_n := \bigoplus_{i \in \mathcal{I}(n)}(\mathcal{F}_n \circ \widehat{\mathcal{H}}^i_n)$.*

Then for every integer function t computable in time $e_t \in \mathrm{poly}$, with $t(n) \in \mathcal{I}(n)$ and $2^{t(n)} \leq T(n)/p(n)$ for large enough n, where $p \in \mathrm{poly}$ such that $p(n) \geq n + 3e_t(n) + 2e_{\mathcal{H}}(n)$, we see that G is a $(\sqrt[3]{2^t}/(q \cdot e_t))$-adaptive PRF, where $q \in \mathrm{poly}$ such that $q(n) \geq |\mathcal{I}(n)|\,(e_{\mathcal{I}}(n) + e_{\mathcal{H}}(n) + e_{\mathcal{F}}(n))$.

Before proving the corollary, let us first use it for constructing an adaptive PRF from a non-adaptive PRF whose security is *not known* at construction time. The resulting PRF makes logarithmic number of calls to the underlying non-adaptive PRF, and assuming the non-adaptive PRF is T-non-adaptive PRF, the resulting PRF is $\sqrt[6]{T}$-adaptive PRF.

Corollary 3.9. *Let $\mathcal{F}$ be a function family, let $\mathcal{H}$ be an efficient length-preserving pairwise independent function family ensemble and let $\mathcal{I}(n) = \{1, 2, 4, \ldots, 2^{\lfloor \log n \rfloor}\}$. Let $G = \{\bigoplus_{i \in \mathcal{I}(n)} (\mathcal{F}_n \circ \widehat{\mathcal{H}}_n^i)\}_{n \in \mathbb{N}}$.*

Assuming $\mathcal{F}$ is $(p \cdot T)$-non-adaptive PRF, for some function T with evaluation time e_T, then G is $(\sqrt[6]{T}/(q \cdot e_T))$-adaptive PRF, where p and q be as in the statement of Corollary 3.8 (replacing e_t with e_T)

Proof. For $n \in \mathbb{N}$ let $t(n) = \max\{i \in \mathcal{I}(n) \colon 2^i \leq T(n)\}$. Note that in order to compute t, it is suffice to compute T. Moreover, since every element in $\mathcal{I}(n)$ is the square of its predecessor, it follows that $T(n) \geq 2^{t(n)} \geq \sqrt{T(n)}$ for every $n \in \mathbb{N}$. Corollary 3.8 yields the result that G is $(\sqrt[3]{2^t}/(q \cdot e_T))$-adaptive PRF and therefore G is $(\sqrt[6]{T}/(q \cdot e_T))$-adaptive PRF. □

Proof of Corollary 3.8. It is easy to see that G is efficient, so it is left to argue for its security. In the following we assume for simplicity that the evaluation time of 2^n on input $n \in \mathbb{N}$ is bounded by n. Let t be an integer function computable in time e_t with $t(n) \in \mathcal{I}(n)$ and $2^{t(n)} \leq T(n)/p(n)$ for every $n \geq n^* \geq 0$. It follows that $\widehat{\mathcal{H}}^t = \{\widehat{\mathcal{H}}_n^{t(n)}\}_{n \in \mathbb{N}}$ is an efficient pairwise independent function family ensemble with evaluation and sampling time $e_{\widehat{\mathcal{H}}^t}(n) \leq e_{\mathcal{H}}(n) + e_t(n)$. Moreover, let $T^*(n) = 2^{t(n)}$, and note that $T^*(n)$ can be computed in time $e_{T^*}(n) \leq e_t(n) + t(n) \leq e_t(n) + n$. Thus, $\mathcal{F}$ is $(p \cdot T^*)$-non-adaptive PRF, $\widehat{\mathcal{H}}^t = \{\widehat{\mathcal{H}}_n^{t(n)} \colon \{0,1\}^n \mapsto [T^*(n)]_{\{0,1\}^n}\}_{n \in \mathbb{N}}$ is an efficient pairwise independent function family ensemble and $p(n) \geq n + 3e_t(n) + 2e_{\mathcal{H}}(n) \geq e_{T^*}(n) + 2e_{\widehat{\mathcal{H}}^t}(n)$. Hence, Corollary 3.2 yields the result that $\mathcal{F} \circ \widehat{\mathcal{H}}^t$ is a $(\sqrt[3]{T^*} = \sqrt[3]{2^t})$-adaptive PRF.

Assume towards a contradiction that there exists an oracle-aided distinguisher D that runs in time $T'(n) = \sqrt[3]{2^{t(n)}}/(q(n) \cdot e_t(n))$ and

$$\left|\Pr_{g \leftarrow G_n}\left[\mathsf{D}^g\left(1^n\right) = 1\right] - \Pr_{\pi \leftarrow \Pi_n}\left[\mathsf{D}^\pi\left(1^n\right) = 1\right]\right| > 1/T'(n) \tag{3}$$

for infinitely many n's. We use the following distinguisher for breaking the pseudorandomness of $\mathcal{F} \circ \widehat{\mathcal{H}}^t$:

Algorithm 3.10 ($\widehat{\mathsf{D}}$).

Input: 1^n.
Oracle: *a function ϕ over* $\{0,1\}^n$.

1. *For every $i \in \mathcal{I}(n) \setminus \{t(n)\}$, choose $g^i \leftarrow \mathcal{F}_n \circ \widehat{\mathcal{H}_n}^i$.*
2. *Set $g := \phi \oplus \bigoplus_{i \in \mathcal{I}(n) \setminus \{t(n)\}} g^i$.*
3. *Emulate $\mathsf{D}^g(1^n)$.*

Note that $\widehat{\mathsf{D}}$ can be implemented to run in time $|\mathcal{I}(n)|\,(e_{\mathcal{I}}(n) + e_{\mathcal{H}}(n) + e_{\mathcal{F}}(n)) \cdot e_t(n) \cdot \sqrt[3]{2^{t(n)}}/(q(n) \cdot e_t(n)) \leq \sqrt[3]{2^{t(n)}}$. Also note that in case ϕ is uniformly distributed over Π_n, then g (selected by $\widehat{\mathsf{D}}^\phi(1^n)$) is uniformly distributed in Π_n, where in case ϕ is uniformly distributed in $\mathcal{F}_n \circ \widehat{\mathcal{H}}_n^{t(n)} = (\mathcal{F} \circ \widehat{\mathcal{H}}^t)_n$ and $n \geq n^*$, then g is uniformly

distributed in G_n. It follows that

$$\begin{aligned}&\left|\Pr_{g\leftarrow(\mathcal{F}\circ\widehat{\mathcal{H}}^t)_n}\left[\widehat{\mathsf{D}}^g\left(1^n\right)=1\right]-\Pr_{\pi\leftarrow\Pi_n}\left[\widehat{\mathsf{D}}^\pi\left(1^n\right)=1\right]\right|\\&=\left|\Pr_{g\leftarrow G_n}\left[\mathsf{D}^g\left(1^n\right)=1\right]-\Pr_{\pi\leftarrow\Pi_n}\left[\mathsf{D}^\pi\left(1^n\right)=1\right]\right|\end{aligned}\tag{4}$$

for every $n\geq n^*$. In particular, Sect. 3 yields the result that

$$\left|\Pr_{g\leftarrow(\mathcal{F}\circ\widehat{\mathcal{H}}^t)_n}\left[\widehat{\mathsf{D}}^g\left(1^n\right)=1\right]-\Pr_{\pi\leftarrow\Pi_n}\left[\widehat{\mathsf{D}}^\pi\left(1^n\right)=1\right]\right|>\frac{q(n)\cdot e_t(n)}{\sqrt[3]{2^{t(n)}}}>\frac{1}{\sqrt[3]{2^{t(n)}}}$$

for infinitely many n's, in contradiction to the pseudorandomness of $\mathcal{F}\circ\widehat{\mathcal{H}}^t$ proven above. □

3.4.1. *Polynomial Security*

Corollary 3.9 immediately yields a construction of a polynomially secure adaptive PRF from a polynomially secure non-adaptive one.[4] The resulting PRF, however, makes logarithmic number of calls to the underlying non-adaptive PRF. Below we show how to construct a polynomially secure adaptive PRF that makes only $\omega(1)$ such calls.

Corollary 3.11 (Restatement of Corollary 1.2). *Let $\mathcal{F}$ be a polynomially secure non-adaptive PRF, let $\mathcal{H}$ be an efficient pairwise independent length-preserving function family ensemble and let $k(n)\in\omega(1)$ be a polynomial-time computable function. Then $G:=\{\bigoplus_{i\in[k(n)]}(\mathcal{F}_n\circ\widehat{\mathcal{H}}_n^{\lfloor i\cdot\log n\rfloor})\}_{n\in\mathbb{N}}$ is polynomially secure adaptive PRF.*

Proof. We show that G is r-adaptive PRF for every $r\in$ poly. In the following we assume for simplicity that the evaluation time of a polynomial and of a base two logarithm on input $n\in\mathbb{N}$, is bounded by n (even for short inputs).

Fix $c_T\in\mathbb{N}$ to be determined by the analysis and let $T(n)=n^{c_T}$ (note that $\mathcal{F}$ is T-non-adaptive PRF). Let $e_{\mathcal{F}}$ and $e_{\mathcal{H}}$ be bounds on the sampling and evaluation time of $\mathcal{F}$ and $\mathcal{H}$ respectively. Let $\mathcal{I}(n):=\{\lfloor\log n\rfloor,\lfloor 2\cdot\log n\rfloor,\ldots,\lfloor k(n)\cdot\log n\rfloor\}$ and let $e_{\mathcal{I}}$ be a bound on the evaluation time of $\mathcal{I}(n)$. Let $p(n)=n^{c_p}\geq 4n+2e_{\mathcal{H}}$ and let $q(n)=n^{c_q}\geq k(n)(e_{\mathcal{I}}(n)+e_{\mathcal{H}}(n)+e_{\mathcal{F}}(n))$ (i.e., q bounds the evaluating and sampling time of k, $\mathcal{I}$, $\mathcal{H}$ and $\mathcal{F}$). Finally, let $t(n)=\lfloor(c_T-c_p)\cdot\log n\rfloor$.

For large enough n we have (1) $t(n)\in\mathcal{I}(n)$, (2) $2^{t(n)}\leq T(n)/p(n)$ and (3) the evaluation time of $t(n)$ is at most n, and thus $p(n)\geq n+3e_t(n)+2e_{\mathcal{H}}(n)$ (i.e., p bounds the evaluating and sampling time of $2^{t(n)}$ and $\mathcal{H}$). Hence, Corollary 3.8 yields the result that G is an $(n^{(c_T-c_p)/3-c_q-1})$-adaptive PRF. Taking $c_T=3(c_r+c_q+1)+c_p$, for a fixed $c_r\in\mathbb{N}$, yields the result that G is an n^{c_r}-adaptive PRF. □

[4] In order to show that G of Corollary 3.9 is r-adaptive PRF, for some $r\in$ poly, it is suffice to take $\mathcal{F}$ that is r^6-non-adaptive PRF. If $\mathcal{F}$ is assumed to be polynomially secure non-adaptive PRF, then it is also r^6-non-adaptive PRF, as required

Acknowledgements

We are very grateful to Omer Reingold for very useful discussions, and for challenging the second author with this research question a long while ago. We also thank the anonymous referees for their useful comments.

Appendix A. From Polynomial to Super-Polynomial Security

The standard security definition for cryptographic primitives is that of *polynomial security*: any PPTM trying to break the primitive has only negligible success probability. Bellare [1] showed that for any polynomially secure primitive there exists a *single* negligible function μ, such that no PPTM can break the primitive with probability larger than μ. Here we take his approach a step further, showing that for any polynomially secure primitive, there exists a super-polynomial function T, such that no adversary of running time T breaks the primitive with probability larger than $1/T$.

In the following we identify algorithms with their string description. In particular, when considering algorithm A, we mean the algorithm defined by the string A (according to some canonical representation). A T-time algorithm makes at most $T(n)$ steps on input of length n.

We prove the following result.

Theorem A.1. *Let $v\colon \{0,1\}^* \times \mathbb{N} \mapsto [0,1]$ be a function with the following properties*: (1) $v(\mathsf{A}, n) = \mathrm{neg}(n)$ *for every oracle-aided* PPTM A; *and* (2) *if the distributions induced by random executions of $\mathsf{A}^f(x)$ and $\mathsf{B}^f(x)$ are the same for any input $x \in \{0,1\}^n$ and function f (each distribution describes the algorithm's output and oracle queries), then $v(\mathsf{A}, n) = v(\mathsf{B}, n)$.*[5]

Then there exists a non-decreasing integer function $T(n) \in n^{\omega(1)}$ such that $v(\mathsf{A}, n) \leq 1/T(n)$ for every T-time algorithm A and large enough n.

Remark A.2 (Applications). Let f be a polynomially secure OWF (i.e., $\Pr[\mathsf{A}(f(U_n)) \in f^{-1}(f(U_n))] = \mathrm{neg}(n)$ for any PPTM A). Applying Theorem A.1 with $v(\mathsf{A}, n) := \Pr[\mathsf{A}(f(U_n)) \in f^{-1}(f(U_n))]$, yields the result that f is super-polynomially secure OWF (i.e., there exists $T(n) \in n^{\omega(1)}$ such that $\Pr[\mathsf{A}(f(U_n)) \in f^{-1}(f(U_n))] \leq 1/T(n)$ for any algorithm of running time T and large enough n).

Similarly, for a polynomially secure PRF $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ (see Definition 2.5), applying Theorem A.1 with $v(\mathsf{A}, n) := |\Pr_{f \leftarrow \mathcal{F}_n}[\mathsf{A}^f(1^n) = 1] - \Pr_{\pi \leftarrow \Pi_n}[\mathsf{A}^\pi(1^n) = 1]|$, where Π_n is the set of all functions with the same domain/range as $\mathcal{F}_n$, yields the result that $\mathcal{F}$ is super-polynomially secure PRF.

Proof of Theorem A.1. Given an algorithm A and an integer i, let A_i denote the variant of A that on input of length n, halts after n^i steps (hence, A_i is a PPTM for any fix $i \in \mathbb{N}$). Let $\mathcal{S}_i$ be the first i strings in $\{0,1\}^*$, according to some canonical order, viewed

[5] That is, v is determined by the algorithm's behavior.

as descriptions of i algorithms. Let $\mathcal{I}(n) = \{1\} \cup \{i \in [n] \colon \forall \mathsf{A} \in \mathcal{S}_i, k \geq n \colon v(\mathsf{A}_i, k) < 1/k^i\}$, let $t(n) = \max \mathcal{I}(n)$ and let $T(n) = n^{t(n)}$. Claim A.3 states that t is unbounded and non-decreasing.

Let A be a T-time algorithm, and let i_{A} be the first integer such that $\mathsf{A} \in \mathcal{S}_{i_{\mathsf{A}}}$. By Claim A.3 $\exists n^* \in \mathbb{N}$ such that $t(n) > i_{\mathsf{A}}$ for every $n \geq n^*$. Fix $n \geq n^*$. Since $\mathsf{A} \in \mathcal{S}_{t(n)}$, we have $v(\mathsf{A}_{t(n)}, n) < 1/n^{t(n)} = 1/T(n)$.

In addition, since A is of running time T, the second property of v yields the result that $v(\mathsf{A}, n) = v(\mathsf{A}_{t(n)}, n)$, and therefore $v(\mathsf{A}, n) < 1/T(n)$. □

Claim A.3. The function t is unbounded and non-decreasing.

Proof. To see that t is unbounded, fix $i \in \mathbb{N}$, and for each $\mathsf{A} \in \mathcal{S}_i$, let n_{A} be the first integer such that $v(\mathsf{A}_i, n) < 1/n^i$ for every $n \geq n_{\mathsf{A}}$ (such n_{A} exists by the first property of v), and let $n_i = \max\{n_{\mathsf{A}} \colon \mathsf{A} \in \mathcal{S}_i\} \cup \{i\}$. It follows that $i \in [n_i]$ and that $v(\mathsf{A}_i, n) < 1/n^i$ for every $n \geq n_i$ and $\mathsf{A} \in \mathcal{S}_i$, yielding that $t(n_i) \geq i$.

Intuitively, t is non-decreasing since once an algorithm is taken into consideration in $\mathcal{I}(n^*)$, for some $n^* \in \mathbb{N}$, it will be taken into consideration in $\mathcal{I}(n)$ for any $n > n^*$. Fix $n^* \in \mathbb{N}$. To formally argue the above we show that $t(n) \geq t^* = t(n^*)$ for every $n > n^*$. The definition of t yields the result that $v(\mathsf{A}_{t^*}, k) \leq 1/k^{t^*}$ for every $\mathsf{A} \in \mathcal{S}_{t^*}$ and $k \geq n^*$. It immediately follows that $v(\mathsf{A}_{t^*}, k) \leq 1/k^{t^*}$ for every $\mathsf{A} \in \mathcal{S}_{t^*}$ and $k \geq n > n^*$. Hence, $t^* \in \mathcal{I}(n)$, and thus $t(n) \geq t^*$. □

A.1. *Non-uniform Security*

Theorem A.1 holds only for uniform algorithms (i.e., Turing machines). Here we prove a similar result for the non-uniform case (i.e., polynomially bounded circuits). In the following we consider adversaries that are families of circuits, denoted with $\mathcal{A} = \{\mathsf{A}_n\}_{n \in \mathbb{N}}$. A circuit A is of size (at most) s, if A has at most s gates. Similarly, a circuit family $\mathcal{A} = \{\mathsf{A}_n\}_{n \in \mathbb{N}}$ is of size s, here s is a function, if A_n is of size $s(n)$ for every $n \in \mathbb{N}$. The family $\mathcal{A}$ is *polynomially bounded*, if it is of size p for some $p \in \text{poly}$.

Theorem A.4. *Let S be the set of all circuits and let $v \colon \mathcal{S} \mapsto [0, 1]$*[6] *be a function with $v(\mathsf{A}_n) = \text{neg}(n)$ for every oracle-aided polynomially bounded circuit family $\mathcal{A} = \{\mathsf{A}_n\}_{n \in \mathbb{N}}$. Then there exists a non-decreasing integer function $T(n) \in n^{\omega(1)}$ and $n^* \in \mathbb{N}$, such that $v(\mathsf{A}_n) \leq 1/T(n)$ for every circuit family $\mathcal{A} = \{\mathsf{A}_n\}_{n \in \mathbb{N}}$ of size T and $n \geq n^*$.*

Proof. We use the following approach (adopted from [1]): for integer pair (n, s), let $\mathcal{C}_{n,s}$ be the set of all n-input, s-size circuits. Fix $\mathsf{B}_{n,s} \in \mathcal{C}_{n,s}$ with $v(\mathsf{B}_{n,s}) \geq v(\mathsf{C})$ for all $\mathsf{C} \in \mathcal{C}_{n,s}$ (note that $\mathsf{B}_{n,s}$ is well defined since $\mathcal{C}_{n,s}$ is finite). For $i \in \mathbb{N}$, let $\mathcal{B}^i = \{\mathsf{B}_{n,n^i}\}_{n \in \mathbb{N}}$ and let $\mathcal{I}(n) = \{0\} \cup \{i \in [n] \colon \forall k \geq n \colon v(\mathsf{B}_{k,k^i}) < 1/k^i\}$. Namely, for every $i \in \mathcal{I}(n)$ and $k \geq n$, the "success" of any circuit family of size k^i is bounded by $1/k^i$. Let $t(n) = \max \mathcal{I}(n)$ and let $T(n) = n^{t(n)}$. Claim A.5 states that t is a non-decreasing unbounded integer function. Hence, to complete the proof, it is left to show that there

[6] In the uniform case the second parameter of v was used to represent the length of the input given to the algorithm. In contrast, in the non-uniform case, a circuit can only receive a single input length, and there is no need to give the input length as a parameter. Thus v's domain is restricted only to $\mathcal{S}$.

exists $n^* \in \mathbb{N}$ such that $v(\mathsf{A}_n) \leq 1/T(n)$ for every circuit family $\mathcal{A} = \{\mathsf{A}_n\}$ of size T and $n \geq n^*$.

Indeed, let $\mathcal{A} = \{\mathsf{A}_n\}_{n\in\mathbb{N}}$ be a circuit family of size T, let $n^* \in \mathbb{N}$ be such that $t(n^*) \geq 1$ (such n^* is guaranteed to exist by Claim A.5) and fix $n \geq n^*$. The definition of t yields the result that $v(\mathsf{B}_{n,n^{t(n)}}) < 1/n^{t(n)} = 1/T(n)$. Finally, the definition of $B_{n,n^{t(n)}}$ yields the result that $v(\mathsf{A}_n) \leq v(\mathsf{B}_{n,n^{t(n)}})$, and therefore $v(\mathsf{A}_n) \leq 1/T(n)$. □

Claim A.5. The function t is a non-decreasing unbounded integer function.

Proof. To see that t is unbounded, we fix $i \in \mathbb{N}$ and show that $\exists n \in \mathbb{N}\colon t(n) \geq i$. Consider the circuit family $\mathcal{B}^i$, let $n_{\mathcal{B}^i}$ be the first integer such that $v(\mathsf{B}_{n,n^i}) < 1/n^i$ for every $n \geq n_{\mathcal{B}^i}$ (note that such $n_{\mathcal{B}^i}$ exists by the property of v) and let $n_i = \max\{n_{\mathcal{B}^i}, i\}$. It follows that $i \in [n_i]$ and that $v(\mathsf{B}_{n,n^i}) < 1/n^i$ for every $n \geq n_i$, yielding that $t(n_i) \geq i$.

To see that t is non-decreasing, we fix $n^* \in \mathbb{N}$, and show that $t(n) \geq t^* = t(n^*)$ for every $n > n^*$. The definition of t yields the result that $v(\mathsf{B}_{k,k^{t^*}}) < 1/k^{t^*}$ for every $k \geq n^*$. It immediately follows that $v(\mathsf{B}_{k,k^{t^*}}) < 1/k^{t^*}$ for every $k \geq n > n^*$. Hence, $t^* \in \mathcal{I}(n)$, and thus $t(n) \geq t^*$. □

References

[1] M. Bellare, A note on negligible functions. *J. Cryptol.*, 271–284 (2002). doi:10.1007/s00145-002-0116-x

[2] I. Berman, I. Haitner, From non-adaptive to adaptive pseudorandom functions, in *Theory of Cryptography, 9th Theory of Cryptography Conference, TCC 2012* (2012), pp. 357–368

[3] I. Berman, I. Haitner, I. Komargodski, M. Naor, Hardness preserving reductions via cuckoo hashing, in *Theory of Cryptography, 10th Theory of Cryptography Conference, TCC 2013* (2013), pp. 40–59

[4] L.J. Carter, M.N. Wegman, Universal classes of hash functions. *J. Comput. Syst. Sci.*, 143–154 (1979). doi:10.1145/800105.803400

[5] C. Cho, C.-K. Lee, R. Ostrovsky, Equivalence of uniform key agreement and composition insecurity, in *Advances in Cryptology – CRYPTO 2010* (2010), pp. 447–464

[6] I. Damgård, J.B. Nielsen, Expanding pseudorandom functions; or: from known-plaintext security to chosen-plaintext security, in *Advances in Cryptology – CRYPTO 2002* (2002), pp. 449–464

[7] Y. Dodis, E. Kiltz, K. Pietrzak, D. Wichs, Message authentication, revisited, in *Advances in Cryptology – EUROCRYPT 2012* (2012), pp. 355–374

[8] O. Goldreich, *Foundations of Cryptography: Basic Tools* (Cambridge University Press, Cambridge, 2001)

[9] O. Goldreich, *Foundations of Cryptography – VOLUME 2: Basic Applications* (Cambridge University Press, Cambridge, 2004)

[10] O. Goldreich, S. Goldwasser, S. Micali, On the cryptographic applications of random functions, in *Advances in Cryptology – CRYPTO '84* (1984), pp. 276–288

[11] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions. *J. ACM*, 792–807 (1986). doi:10.1445/6490.6503

[12] J. Håstad, R. Impagliazzo, L.A. Levin, M. Luby, A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 1364–1396 (1999). doi:10.1137/s0097539793244708

[13] M. Luby, *Pseudorandomness and Cryptographic Applications*. Princeton Computer Science Notes. (Princeton University Press, Princeton, 1996). ISBN 978-0-691-02546-9

[14] V. Lyubashevsky, D. Masny, Man-in-the-middle secure authentication schemes from LPN and weak PRFS. *IACR Cryptol. ePrint Arch.* **2013**, 92 (2013)

[15] U.M. Maurer, K. Pietrzak, Composition of random systems: when two weak make one strong, in *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004* (2004), pp. 410–427

[16] U.M. Maurer, J. Sjödin, A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security, in *Advances in Cryptology – EUROCRYPT 2007* (2007), pp. 498–516

[17] U.M. Maurer, S. Tessaro, Basing PRFS on constant-query weak PRFS: minimizing assumptions for efficient symmetric cryptography, in *Advances in Cryptology – ASIACRYPT 2008* (2008), pp. 161–178

[18] S. Myers, Black-box composition does not imply adaptive security, in *Advances in Cryptology – EUROCRYPT 2004* (2004), pp. 189–206

[19] M. Naor, O. Reingold, Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci.*, 336–375 (1999). doi:10.1006/jcss.1998.1618

[20] K. Pietrzak, Composition does not imply adaptive security, in *Advances in Cryptology – CRYPTO 2005* (2005), pp. 55–65

[21] K. Pietrzak, Composition implies adaptive security in minicrypt, in *Advances in Cryptology – EUROCRYPT 2006* (2006), pp. 328–338