Bootstrapping Obfuscators via Fast Pseudorandom Functions

Benny Applebaum *

School of Electrical Engineering, Tel-Aviv University
benny.applebaum@gmail.com

Abstract. We show that it is possible to upgrade an obfuscator for a weak complexity class **WEAK** into an obfuscator for arbitrary polynomial size circuits, assuming that the class **WEAK** can compute pseudorandom functions. Specifically, under standard intractability assumptions (e.g., hardness of factoring, Decisional Diffie-Hellman, or Learning with Errors), the existence of obfuscators for $\mathbf{NC}^1$ or even $\mathbf{TC}^0$ implies the existence of general-purpose obfuscators for **P**. Previously, such a bootstrapping procedure was known to exist under the assumption that there exists a fully-homomorphic encryption whose decryption algorithm can be computed in **WEAK**. Our reduction works with respect to virtual black-box obfuscators and relativizes to ideal models.

1 Introduction

General-purpose program obfuscation allows us to transform an arbitrary computer program into an "unintelligible" form while preserving its functionality. The latter property is formalized via the notion of *Virtual Black-Box* which asserts that the code of the obfuscated program reveals nothing more than what can be learned via oracle access to its input-output behavior. The seminal result of [7] shows that general purpose obfuscation is impossible in the standard model. Nevertheless, in a sequence of recent exciting works [10, 8, 6], it was shown that general-purpose obfuscation can be achieved in idealized models such as the Generic Colored Matrix Model, or the Generic Graded Encoding model.

All these works share a similar outline. First it is shown how to use the idealized model to obfuscate a weak complexity class such as $\mathbf{NC}^1$, and then the weak obfuscator is bootstrapped into a general-purpose obfuscator for arbitrary polynomial-size circuits.[1] The bootstrapping step in all these works employs a fully homomorphic encryption [11] whose decryption algorithm can be implemented in $\mathbf{NC}^1$. While recent constructions of FHE (e.g., [9]) make the latter

* Supported by Alon Fellowship, ISF grant 1155/11, Israel Ministry of Science and Technology (grant 3-9094), GIF grant 1152/2011, and the Check Point Institute for Information Security..

[1] The class $\mathbf{NC}^1$ is the class of polynomial-size circuits with logarithmic depth and bounded fan-in gates.

assumption reasonable, the existence of FHE is still a strong public-key assumption and it is natural to ask whether it can be relaxed. Indeed, $\mathbf{NC}^1$ obfuscation already seems to put us at the "Heights of Cryptomania" [12], and so one may suspect whether the extra power of FHE is really needed for bootstrapping.

1.1 Our Results

In this note we show that bootstrapping can be based on a "Minicrypt" type assumption.

Theorem 1 (main theorem – informal). *Assume that the complexity class* **WEAK** *can compute a pseudorandom function (PRF). Then an obfuscator for* **WEAK** *(in some idealized model) can be bootstrapped into an obfuscator for every polynomial-size circuit family.*

(See Theorem 5 for a formal statement.) Since practical and theoretical implementations of PRFs tend to be highly efficient, we can instantiate the theorem with relatively low complexity classes. For example, by relying on PRFs constructions from [18, 5], we derive the following corollary.

Corollary 1. *Assuming the hardness of factoring, Decisional Diffie-Hellman, or Learning with Errors, the following holds. If* $\mathbf{TC}^0$ *can be obfuscated (in some idealized model), then every polynomial-size circuit family can be obfuscated as well.*[2]

Tightness. One may ask whether the PRF assumption in Theorem 1 can be further relaxed and replaced with the existence of a weaker primitive such as a one-way function or a pseudorandom generator. We note that the answer seems to be negative as such primitives can be computed in $\mathbf{NC^0}$ (under standard assumptions [4]), a class which is learnable and therefore trivially obfuscatable in the standard model. Therefore, such a strengthening of Theorem 1 would contradict the impossibility results of [7].

In fact, we do not expect to prove a statement like Corollary 1 for classes lower than $\mathbf{TC}^0$, as $\mathbf{TC}^0$ is the lowest complexity class for which the impossibility results of [7] apply. More generally, [7] essentially show that if a complexity class **WEAK** contains a PRF then it cannot be obfuscated in the standard model. Our results complement the picture by showing that if such a class can be obfuscated in some idealized model, then everything can be obfuscated. This suggests the existence of a zero-one law: if an idealized model admits non-trivial obfuscation (i.e., for some class which is non-obfuscatable in the standard model) then it admits general purpose obfuscation for all (polynomial-size) circuits.

[2] The class $\mathbf{TC}^0$ is the class of all Boolean circuits with constant depth and polynomial size, containing only unbounded-fan in AND gates, OR gates, and majority gates. This class is a subclass of $\mathbf{NC}^1$.

Virtual Black-Box vs. Indistinguishability Obfuscation. Our results hold with respect to the strongest security notion of *Virtual Black-Box* (VBB) obfuscation [7] relative to some ideal model. An alternative (weaker) notion of security is *indistinguishability Obfuscation*. The latter notion is highly attractive as it may be achievable in the standard model (no impossibility results are known), and, quite surprisingly, it suffices for a wide range of applications [19]. In this work we focus on VBB obfuscators, as we believe that when *constructing* obfuscators it is best to strive for the strongest form of security. (See a detailed discussion in [6].) We do not know whether our results apply in the indistinguishability setting, and leave this question for future research. Interestingly, the FHE-based bootstrapping works in both settings [10, 8].

1.2 Techniques

Our main technical tool is randomized encoding (RE) of functions [17, 4]. Intuitively, a function $f(x)$ is encoded by a randomized function $\hat{f}(x; r)$ if the distribution $\hat{f}(x)$, induced by a random choice of r, reveals nothing but $f(x)$. Formally, a sample from $\hat{f}(x)$ can be *decoded* to $f(x)$, and vice versa, given $f(x)$ one can efficiently *simulate* the distribution $\hat{f}(x)$ – so the functions are essentially "equivalent". REs become non-trivial (and useful) when their complexity is smaller than the complexity of f. This "equivalence" between a complicated function f to a simpler encoding $\hat{f}$, was exploited in various applications to reduce a complex task to a simpler one. (See the surveys [1] and [16].) We can adopt a similar approach in our context as well.

In order to obfuscate a function f taken from a family $\mathcal{F}$ let us obfuscate its low-complexity encoding $\hat{f}$ and release the latter obfuscated program composed with the decoder algorithm. For this to work, let us assume the existence of universal decoder and universal simulator that work uniformly for all functions in $\mathcal{F}$. (This can be guaranteed by encoding the evaluation function of the collection $\mathcal{F}$ which maps a circuit of f and an input x to the value $f(x)$.)

Unfortunately, this approach is somewhat problematic as the encoding $\hat{f}$ employs internal randomness. One potential solution is to treat the randomness as an additional input and let the user of the obfuscated program choose it. While decoding still succeeds, this solution fails to be secure. Once the randomness r is revealed, the function f can be fully recovered. (Technically, universal simulation cannot be achieved anymore.) Another (flawed) solution, is to fix some secret randomness r and obfuscate the mapping $x \mapsto \hat{f}(x; r)$. Unfortunately, the privacy of the encoding holds only when fresh randomness is being used, and one can easily recover the circuit of f when r is fixed.

The problem is solved via the extra use of a PRF. Specifically, we choose a PRF $h \overset{R}{\leftarrow} \mathcal{H}$ and obfuscate the function $\hat{f}(x; h(x))$. The security of the PRF ensures that this function behaves essentially as a standard encoding whose internal randomness is freshly chosen in each invocation, and so simulation succeeds. By using the low-complexity encoding from [3], the complexity of $\hat{f}$ is dominated by the complexity of the PRF $\mathcal{H} \in \mathbf{WEAK}$ (assuming that the class **WEAK** satisfies some basic closure properties), and one can prove Theorem 1.

We note that a similar usage of PRF-derandomized randomized encoding was made by [14] in the context of Functional Encryption.

2 Preliminaries

We say that a function $\varepsilon : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c > 0$ there exists an integer $n_0 \in \mathbb{N}$ such that $\varepsilon(n) < n^{-c}$ for every $n > n_0$. We will sometimes use $\mathrm{neg}(\cdot)$ to denote an unspecified negligible function.

One-way functions. An efficiently computable function $g : \{0,1\}^* \to \{0,1\}^*$ is *one-way* if for every (non-uniform) efficient adversary $\mathcal{A}$ we have that

$$\Pr_{x \overset{R}{\leftarrow} \{0,1\}^n}[\mathcal{A}(1^n, f(x)) \in f^{-1}(f(x))] < \mathrm{neg}(n). \tag{1}$$

Circuit families. A family of polynomial-size boolean circuits $\mathcal{F}$ is an infinite sequence of circuit families $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ where for every $n \in \mathbb{N}$ the family $\mathcal{F}_n$ consists of boolean circuits with n inputs, $m(n)$ outputs, and circuit size $\ell(n)$ where m, ℓ are polynomials in n. For such a family there is always a universal efficient evaluator F such that for every length parameter n, circuit $f \in \mathcal{F}_n$ and input $x \in \{0,1\}^n$, we have that $F(f, x) = f(x)$.

Pseudorandom functions [13]. Let $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$ be a family of polynomial-size boolean circuits and let $\mathcal{K}$ be a PPT sampling algorithm that on input 1^n samples a circuit in $\mathcal{H}_n$. (The probability distribution induced by $\mathcal{K}$ is not necessarily uniform.) We say that $\mathcal{H}$ is a *pseudorandom function family* (PRF) if for every (non-uniform) efficient oracle-aided adversary $\mathcal{A}$ we have that

$$\left| \Pr_{h \overset{R}{\leftarrow} \mathcal{K}(1^n)}[\mathcal{A}^h(1^n) = 1] - \Pr_{R_n}[\mathcal{A}^{R_n}(1^n) = 1] \right| \leq \mathrm{neg}(n), \tag{2}$$

where R_n is a uniformly chosen function with the same input and output lengths as the functions in $\mathcal{H}_n$. To simplify notation, we will typically make the sampler implicit and write $h \overset{R}{\leftarrow} \mathcal{H}_n$ with the understanding that the distribution is induced by some efficient sampler.

2.1 Randomized Encoding of Functions

Let F be a polynomial-time computable function that maps n bits to $m(n)$ bits. Intuitively, a randomized function $\hat{F}$ is an "encoding" of F if for every input x the distribution $\hat{F}(x)$ reveals the value of $F(x)$ but no other additional information. We formalize this via the notion of *computationally private randomized encoding* from [3].

Definition 1 (Computational randomized encoding). *Let $F : \{0,1\}^n \to \{0,1\}^{m(n)}$ be an efficiently computable function and let $\hat{F}$ be an efficiently computable randomized function. We say that $\hat{F}$ is a* computational randomized encoding *of F (or encoding for short), if there exist an efficient decoder algorithm* D *and an efficient probabilistic simulator algorithm* Sim *that satisfy the following:*

– **Perfect correctness.** *For every n and every input $x \in \{0,1\}^n$,*

$$\mathsf{D}(1^n, \hat{F}(x)) = F(x).$$

– **Computational privacy.** *For every non-uniform efficient oracle-aided adversary $\mathcal{A}$ we have*

$$\left|\Pr[\mathcal{A}^{\hat{F}_n}(1^n) = 1] - \Pr[\mathcal{A}^{\mathsf{Sim}(1^n, F_n(\cdot))}(1^n) = 1]\right| \leq \mathrm{neg}(n), \qquad (3)$$

where $\hat{F}_n$ and F_n are the restrictions of $\hat{F}$ and F to n-bit inputs, and both oracles are probabilistic functions (and fresh randomness is used in each invocation).

Encoding Collections. We encode a family of polynomial-size boolean circuits $\mathcal{F}$ by encoding its evaluation algorithm $F(f, x)$. Specifically, for every $f \in \mathcal{F}$ we define the randomized function $\hat{f}(x; r) = \hat{F}(f, x; r)$ where $\hat{F}$ is the encoder of F. By definition, the decoder D and the simulator Sim of F apply universally for all $f \in \mathcal{F}$. Formally, for every n and $f \in \mathcal{F}_n$:

$$\mathsf{D}(1^{|x|}, \hat{f}(x)) = f(x) \quad \forall x \in \{0,1\}^n.$$

Also for every non-uniform efficient oracle-aided adversary $\mathcal{A}$ and every sequence of functions $\{f_n\}$ and their encodings $\left\{\hat{f}_n\right\}$ we have that

$$\left|\Pr[\mathcal{A}^{\hat{f}_n}(1^n) = 1] - \Pr[\mathcal{A}^{\mathsf{Sim}(1^n, f_n(\cdot))}(1^n) = 1]\right| \leq \mathrm{neg}(n). \qquad (4)$$

(The oracles in (4) are simply the restriction of the oracles in (3) to inputs of the form $(f_n, \cdot)$ and so (4) follows immediately from (3).)

2.2 Obfuscation

Definition 2 (Virtual Black-Box Obfuscator [7]). *Let $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ be a family of polynomial-size boolean circuits. An obfuscator $\mathcal{O}$ for $\mathcal{F}$ is a PPT algorithm which maps a circuit $f \in \mathcal{F}_n$ to a new circuit $[f]$ (not necessarily in $\mathcal{F}$) such that the following properties hold:*

1. ***Preserving Functionality.*** *For every $n \in \mathbb{N}$, every $f \in \mathcal{F}_n$ and every input $x \in \{0,1\}^n$*

$$\Pr_{[f] \xleftarrow{R} \mathcal{O}(f)}[[f](x) \neq f(x)] \leq \mathrm{neg}(n).$$

2. ***Polynomial slowdown.*** *There exists a polynomial p such that for every $n \in \mathbb{N}$ and $f \in \mathcal{F}_n$ the circuit $\mathcal{O}(f)$ is of size at most $p(|f|)$.*
3. ***Virtual Black-Box.*** *For every (non-uniform) efficient adversary $\mathcal{A}$ there exists a (non-uniform) efficient simulator* Sim *such that for every n and every $f \in \mathcal{F}_n$:*

$$\left|\Pr[\mathcal{A}(\mathcal{O}(f)) = 1] - \Pr[\mathsf{Sim}^f(1^{|f|}, 1^n) = 1]\right| \leq \operatorname{neg}(n). \tag{5}$$

A complexity class $\mathbf{C}$ is obfuscatable if there exists an efficiently computable mapping that maps every efficiently computable function family $\mathcal{F} \in \mathbf{C}$ (represented by its evaluator F) to an obfuscator $\mathcal{O}$ for $\mathcal{F}$.

Obfuscation in an idealized model. An idealized model is captured by a sequence of probabilistic stateful oracles $\mathcal{M} = \{\mathcal{M}_n\}_{n\in\mathbb{N}}$ indexed by a security parameter n. We consider obfuscators which are implementable relative to $\mathcal{M}$. This means that the obfuscator $\mathcal{O}$ is allowed to make oracle queries to $\mathcal{M}_n$ and that Eq. 5 should hold even when $\mathcal{A}$ is allowed to query $\mathcal{M}_n$. (The circuit f cannot have oracle gates to $\mathcal{M}$.) In this case, Properties (1) and (2) should hold for every possible coins of $\mathcal{M}$.

3 Our Reduction

Let $\mathcal{F}$ be a family of polynomial-size boolean circuits with an evaluator F. We will construct an obfuscator $\mathcal{O}$ for $\mathcal{F}$ based on the following ingredients. (1) An encoding $\hat{F}$ of F; (2) a pseudorandom function family $\mathcal{H}$ where the output length of functions in $\mathcal{H}_n$ equals to the length of the random input of $\hat{F}_n$; and (3) a weak obfuscator $\mathsf{weak}\mathcal{O}$ for the circuit family $\mathcal{G} = \{\mathcal{G}_n\}$ where $\mathcal{G}_n$ contain all circuits of the form

$$g_{f,h} : x \mapsto \hat{F}(f, x; h(x)), \qquad \forall f \in \mathcal{F}_n, h \in \mathcal{H}_n.$$

We allow the weak obfuscator to be implementable in some idealized model $\mathcal{M}$, but assume that the PRF and the randomized encoding are implemented in the standard model and make no calls to $\mathcal{M}$.

Construction 2. *Given a circuit $f \in \mathcal{F}_n$ the obfuscator $\mathcal{O}^{\mathcal{M}_n}$ does the following:*

– *Sample a random $h \overset{R}{\leftarrow} \mathcal{H}$ and obfuscate $g_{f,h}$ by $[g] := \mathsf{weak}\mathcal{O}^{\mathcal{M}_n}(g_{f,h})$.*
– *Output the circuit $[f] = \mathsf{D} \circ [g]$ where D is the RE decoder and $\circ$ denotes function composition.*

Note that the construction is syntactically well defined as $g_{f,h}$ makes no calls to the oracle $\mathcal{M}$. (For this purpose, we had to assume that the PRF and the randomized encoding do not use $\mathcal{M}$.)

It is easy to verify that $[f]$ preserves the functionality of f.

Lemma 1. *The obfuscator $\mathcal{O}$ is functionality preserving.*

Proof. Fix some x and $h \in \mathcal{H}$, and let us condition on the event that the weak obfuscator preserves the functionality, namely, $[g](x) = g_{f,h}(x)$. Then, by the correctness of the encoding, we have

$$[f](x) = \mathsf{D}([g](x)) = \mathsf{D}(g_{f,h}(x)) = \mathsf{D}(\hat{F}(f, x; h(x))) = f(x).$$

Since the weak obfuscator is correct with all but negligible probability the claim follows. $\square$

In the next section, we will prove that the obfuscator is secure.

Lemma 2. *The obfuscator $\mathcal{O}$ satisfies the Virtual Black-Box property relative to $\mathcal{M}$.*

3.1 Security (Proof of Lemma 2)

Let $\mathcal{A}$ be an efficient adversary for the new obfuscator $\mathcal{O}$. Our goal is to construct a simulator Sim that simulates $\mathcal{A}$. For this aim, let us first define an oracle-aided adversary $\mathcal{B}$ for the weak obfuscator $\mathsf{weak}\mathcal{O}$ as follows. Given an obfuscated circuit $[f]$, the adversary $\mathcal{B}$ applies $\mathcal{A}$ to the circuit $\mathsf{D} \circ [f]$ where D is the (universal) decoder of the encoding. If $\mathcal{A}$ makes oracle queries to the oracle $\mathcal{M}_n$ then $\mathcal{B}$ answers them using his own oracle $\mathcal{M}_n$. Let $\mathsf{weakSim}$ be the simulator of $\mathsf{weak}\mathcal{O}$ which simulates the adversary $\mathcal{B}$, and let reSim be the (universal) RE simulator.

We define the simulator $\mathsf{Sim}^f(1^n)$ for the adversary $\mathcal{A}$ as follows. Invoke the oracle-aided weak simulator $\mathsf{weakSim}^g(1^n)$, and whenever $\mathsf{weakSim}$ makes a new oracle query $x \in \{0,1\}^n$, answer it with $\mathsf{reSim}(1^n, f(x))$, where the latter is computed via the help of the oracle f. If x was previously queried, respond with the same answer as before.

Fix some $f \in \mathcal{F}_n$. We will prove that

$$\left|\Pr[\mathcal{A}^{\mathcal{M}_n}(\mathcal{O}(f)) = 1] - \Pr[\mathsf{Sim}^f(1^n) = 1]\right| \leq \varepsilon_1(n) + \varepsilon_2(n) + \varepsilon_3(n) \qquad (6)$$

where ε_1 (resp., $\varepsilon_2, \varepsilon_3$) upper-bounds the distinguishing advantage of efficient adversaries against the weak obfuscator (resp., against the PRF, against the encoding). Through the proof, we simplify notation by omitting the unary input 1^n; also for a binary random variable Y we write $\Pr[Y]$ to denote $\Pr[Y = 1]$.

Let $\hat{f}(x; r)$ be the circuit that computes the encoding of $f(x)$, i.e., $\hat{f}(x; r) = \hat{F}(f, x; r)$. For a function $h \in \mathcal{H}$, let $\hat{f}_h$ denote the circuit that computes $\hat{f}(x; h(x))$ and let $\mathcal{O}(f; h)$ denote the output of the obfuscator $\mathcal{O}$ with input f and PRF h. Then, by definition, for every h, we have that

$$\Pr[\mathcal{A}^{\mathcal{M}_n}(\mathcal{O}^{\mathcal{M}_n}(f; h))] = \Pr[\mathcal{B}^{\mathcal{M}_n}(\mathsf{weak}\mathcal{O}^{\mathcal{M}_n}(\hat{f}_h))], \qquad (7)$$

Also, by the VBB property of the weak obfuscator, for every h we have that

$$\left|\Pr[\mathcal{B}^{\mathcal{M}_n}(\mathsf{weak}\mathcal{O}^{\mathcal{M}_n}(\hat{f}_h))] - \Pr[\mathsf{weakSim}^{\hat{f}_h}]\right| \leq \varepsilon_1(n). \qquad (8)$$

By relying on the security of the PRF we will prove the following claim.

Claim 3.

$$\left|\Pr_{h \overset{R}{\leftarrow} \mathcal{H}_n}[\mathsf{weakSim}^{\hat{f}_h}] - \Pr[\mathsf{weakSim}^{\hat{f}}]\right| \leq \varepsilon_2(n),$$

where $\hat{f}(\cdot)$ is viewed as a randomized function and repeated queries to this function are answered consistently based on the first answer.

Proof. If the claim does not hold, we can break the PRF as follows. Let T^R be an oracle aided adversary which calls $\mathsf{weakSim}$ and whenever $\mathsf{weakSim}$ makes a query x to its oracle, T answers the query with $\hat{f}(x; R(x))$. Observe that if R is a truly random function then T accepts with probability exactly $\Pr[\mathsf{weakSim}^{\hat{f}}]$. On the other hand, if the oracle is $R \overset{R}{\leftarrow} \mathcal{H}_n$ then the acceptance probability is exactly $\Pr_{h \overset{R}{\leftarrow} \mathcal{H}_n}[\mathsf{weakSim}^{\hat{f}_h}]$. It follows that T breaks the security of the PRF, and the claim follows. □

Finally, we rely on the privacy of the randomized encoding to prove the following claim.

Claim 4.

$$\left|\Pr[\mathsf{weakSim}^{\hat{f}}] - \Pr[\mathsf{Sim}^{f}]\right| \leq \varepsilon_3(n),$$

where $\hat{f}(\cdot)$ is viewed as a randomized function and repeated queries to this function are answered consistently based on the first answer.

Proof. Recall that $\mathsf{Sim}^{f(\cdot)}$ is simply $\mathsf{weakSim}^{\mathsf{reSim}(f(\cdot))}$, where repeated queries are answered consistently. Therefore, if the claim does not hold, we can use $\mathsf{weakSim}$ to distinguish between the randomized functions $\mathsf{reSim}(f(\cdot))$ and $\hat{f}(\cdot)$, in contradiction to the privacy of the RE (Eq. 4). (The distinguisher will simply invoke $\mathsf{weakSim}$ and will answer repeated queries based on the first answer.) □

The lemma (i.e., Eq. 6) now follows from Eqs. 7 and 8 and Claims 3 and 4. The "furthermore" part follows by noting that the proof relativizes. □

4 Main Result

We say that a circuit complexity class **WEAK** is *admissible* if it satisfies the following basic properties:

1. The class **WEAK** contains the class $\mathbf{NC^0}$;
2. (Closure under concatenation) If each output bit of a multi-output function f is computable in **WEAK** then so is f;
3. (Closure under composition) if $f \in \mathbf{WEAK}$ and $g \in \mathbf{WEAK}$ then $g \circ f$ is in **WEAK**, where $\circ$ denotes function composition.

We can now prove our main theorem.

Theorem 5. *Let* **WEAK** *be an admissible complexity class, and let $\mathcal{M}$ be some probabilistic oracle (an idealized model). Assume that there exists a one-way function and a pseudorandom function $\mathcal{H}$ in* **WEAK**. *Then, if* **WEAK** *can be obfuscated relative to $\mathcal{M}$, every family of polynomial-size circuits can be obfuscated relative to $\mathcal{M}$.*

Proof. First, we claim that, under the above assumption, any efficiently computable function F has an encoding $\hat{F}$ computable in **WEAK**. In [3] it was shown, based on Yao's garbled circuit technique, that $F(x)$ can be encoded by an encoding $\hat{F}(x;r)$ which is reducible to a minimal-stretch pseudorandom generator $G : \{0,1\}^{\kappa} \to \{0,1\}^{\kappa+1}$ via a non-adaptive $\mathbf{NC^0}$ reduction. Namely, $\hat{F}(x;r)$ can be written as $g(x, r, G(r_1), \ldots, G(r_t))$ where g is an $\mathbf{NC^0}$ function and the r_i's are sub-blocks of the random string r. Such a PRG is also reducible to a one-way function via a (non-adaptive) $\mathbf{NC^0}$ reduction by [4, 15] (see also [2, Remark 4.5]). Therefore, $\hat{F}(x;r)$ can be written as $g'(x, r, G'(r_1), \ldots, G'(r_{t'}))$ where g' is in $\mathbf{NC^0}$ and G' is a one-way function. We can now instantiate G' with a one-way function in **WEAK** whose existence is promised by the theorem's hypothesis. Since **WEAK** is closed under concatenation, we can view the t' copies $G'(r_1), \ldots, G'(r_{t'})$ as a single function in **WEAK**. Now, the resulting encoding can be written as a composition of an $\mathbf{NC^0}$ function with a function in **WEAK** which results in a **WEAK** function.[3]

Next, we observe that, since **WEAK** is closed under concatenation, the assumption implies the existence of a PRF in **WEAK** whose output length is equal to the randomness complexity of the encoding. (By using direct product, one can transform a **WEAK** PRF with a single output bit into a new **WEAK** PRF with arbitrary polynomial number of output bits.) It follows that the circuit family $\mathcal{G}$ (from Construction 2) can be written as a composition of two **WEAK** functions, and the theorem follows from Lemmas 1 and 2. □

Remark. We assume nothing on the complexity of the *sampler* of the PRF, and therefore the sampler can preprocess the function $h \in \mathcal{H}$. (This preprocessing is exploited in fast implementations of PRFs [18, 5].) As a result, the additional one-wayness assumption does not seem to follow from the fact that $\mathcal{H}$ is in **WEAK**.[4]

Under standard intractability assumptions, one can instantiate **WEAK** with $\mathbf{NC}^1$ or even $\mathbf{TC}^0$. Indeed, the existence of PRFs in $\mathbf{TC}^0$ can be based on the hardness of factoring, the DDH assumption or the intractability of lattice/learning problems [18, 5], and the existence of one-way functions in $\mathbf{TC}^0$ (or even in $\mathbf{NC^0}$) follow from these assumptions as well [4]. Therefore Corollary 1 follows from Theorem 5.

[3] We note that the construction of the RE makes a non black-box use of the code of the one-way function. This does not affect the overall argument as none of these primitives makes calls to the oracle $\mathcal{M}$.

[4] The one-wayness assumption becomes redundant if the mapping $(k, x) \mapsto h_k(x)$ (where $h_k = \mathcal{K}(1^n; k)$) is computable in **WEAK**.

Acknowledgement. We thank Alon Rosen and Zvika Brakerski for useful discussions.

References

1. B. Applebaum. Randomly encoding functions: A new cryptographic paradigm - (invited talk). In *ICITS*, pages 25–31, 2011.
2. B. Applebaum. *Cryptography in Constant Parallel Time*. Springer Publishing Company, Incorporated, 2014.
3. B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications. *Computional Complexity*, 15(2):115–162, 2006.
4. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.*, 36(4):845–888, 2006.
5. A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology - EUROCRYPT*, pages 719–737, 2012.
6. B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai. Protecting obfuscation against algebraic attacks. In *Advances in Cryptology - EUROCRYPT*, pages 221–238. Springer, 2014.
7. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *J. of the ACM*, 59(2):6, 2012.
8. Z. Brakerski and G. N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *TCC*, pages 1–25. Springer, 2014.
9. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, 2011.
10. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS*, pages 40–49, 2013.
11. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC*, pages 169–178, 2009.
12. C. Gentry. Encrypted messages from the heights of cryptomania. In *TCC*, pages 120–121, 2013.
13. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. of the ACM*, 33:792–807, 1986.
14. S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology - CRYPTO*, pages 162–179. Springer, 2012.
15. I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput*, 42(3):1405–1430, 2013.
16. Y. Ishai. Randomization techniques for secure computation. In M. Prabhakaran and A. Sahai, editors, *Secure Multi-Party Computation*, volume 10 of *Cryptology and Information Security Series*, pages 222–248. IOS press, Amsterdam, 2012.
17. Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *IEEE 41st Annual Symposium on Foundations of Computer Science, FOCS*, pages 294–304, 2000.

18. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. of the ACM*, 51(2):231–262, 2004.
19. A. Sahai and B. Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC*, pages 475–484, 2014.