A Cookbook for Black-Box Separations and a Recipe for UOWHFs

Kfir Barhum and Thomas Holenstein

Department of Computer Science,
ETH Zurich, 8092 Zurich, SWITZERLAND

Abstract. We present a new framework for proving fully black-box separations and lower bounds. We prove a general theorem that facilitates the proofs of fully black-box lower bounds from a one-way function (OWF).
Loosely speaking, our theorem says that in order to prove that a fully black-box construction does not securely construct a cryptographic primitive **Q** (e.g., a pseudo-random generator or a universal one-way hash function) from a OWF, it is enough to come up with a large enough set of functions $\mathcal{F}$ and a parameterized oracle (i.e., an oracle that is defined for every $f \in \{0,1\}^n \to \{0,1\}^n$) such that $\mathcal{O}_f$ breaks the security of the construction when instantiated with f and the oracle satisfies two local properties.
Our main application of the theorem is a lower bound of $\Omega(n/\log(n))$ on the number of calls made by any fully black-box construction of a universal one-way hash function (UOWHF) from a general one-way function. The bound holds even when the OWF is regular, in which case it matches to a recent construction of Barhum and Maurer [4].

Keywords: Complexity-Based Cryptography, One-Way Functions, Universal One-Way Hash Functions, Black-Box Constructions, Lower Bounds

1 Introduction

1.1 Cryptographic Primitives and Black-Box Constructions

An important question in complexity-based cryptography is understanding which cryptographic primitives (e.g., one-way functions, pseudo-random generators) are implied by others. In principle, an implication between two primitives can be proved as a logical statement (e.g., the existence of one-way functions implies the existence of pseudo-random generators). However, most proofs of such implications (with very few exceptions, e.g., [2]) are in fact so-called fully black-box constructions.

Informally, a black-box construction of a primitive **Q** from a primitive **P** is a pair of algorithms, called *construction* and *reduction*, such that

the construction, using only the functionality of **P**, implements **Q** and the reduction, using only the functionality of **P** and the one of a potential breaker algorithm, breaks **P** whenever the breaker algorithm breaks **Q**. As a corollary, such a black-box construction establishes that the existence of **P** implies the existence of **Q**. One of many such examples is the construction of a one-way function from a weak one-way function [18].

After futile attempts to prove that the existence of one-way functions implies that of key agreement, Impagliazzo and Rudich [10] proved the first black-box separation result: They showed that there is no fully black-box construction of key agreement from one-way functions. Their seminal work inspired a plethora of similar results and nowadays one identifies two main types of black-box separation results: black-box separations of a primitive **Q** from a primitive **P** and lower bounds on some complexity parameter (e.g., seed length, number of calls to the underlying primitive, etc.) in the construction of **Q** from **P**. Besides [10], the work of Simon [17], where he shows that there is no fully black-box construction of a collision-resistant hash function from a one-way function, is an example of the former. As an example of the latter, Kim *et. al.* [11] established a lower bound of $\Omega(\sqrt{k/\log(n)})$ on the number of queries of any construction of a universal one-way hash function that compresses k bits from a one-way permutation on n bits. This was later improved by Gennaro *et. al.* [5] to $\Omega(k/\log(n))$.

Reingold *et. al.* [13] were the first to formalize a model for and study the relations between different notions of "black-boxness" of cryptographic constructions.

A key property of a fully black-box construction of **Q** from **P** is the requirement that it constructs **Q** efficiently even when given black-box access to a non-efficient implementation of **P**. A proof technique utilizing this property, which is implicit in many black-box separations, involves an (inefficient) oracle instantiation of the primitive **P** and an appropriate (inefficient) breaker oracle B. The separation is usually proved by showing that B breaks the security of the candidate construction for **Q**, but at the same time no efficient oracle algorithm that has black-box oracle access to both the breaker and the primitive (in particular, the potential reduction) breaks the security property of the underlying instantiation of **P**.

1.2 Our Contribution

In constructions based on one-way functions (or permutations), i.e., when $\mathbf{P} = \mathbf{OWF}$, the oracle that implements **OWF** is usually set to be a random permutation, which is one-way with very high probability even in the

presence of a non-uniform algorithm. On the other hand, the proof that the breaker algorithm for the constructed primitive $\mathbf{Q}$ does not help invert the permutation is repeated in an "ad-hoc" manner in many separation proofs, e.g., in [17, 6] and also in a recent result on lower bounds on the number of calls made by any construction of a pseudo-random generator from a one-way function [9].

Thus, while in many separation proofs the task of finding the right breaker oracle is different (this is inherent, as each time it is required to break the security of a different primitive), we observe that the proof that it does not help in inverting the underlying one-way function can be facilitated and unified to a large extent. To that end, we prove a general theorem that facilitates the proof of black-box separations (Theorem 1). In particular, we show that any circuit with access to an oracle that satisfies two local properties, does not help to invert many functions.

Our framework allows proving separation results that exclude the existence of reductions with very weak security requirements. In this work we focus on the important case where the black-box construction is so-called fixed-parameter. That is, for a security parameter ρ, both the construction algorithm and the reduction access the primitive and breaker of security ρ only. All black-box constructions found in the literature are in fact fixed-parameter constructions. We believe that adapting the approach of [9], it is possible to extend our results to the most general case.

Our proof uses the encoding technique from [5], which was already adapted to the special cases in [6] and [9]. We also use the bending technique that originated in [17] and was subsequently used in [7] and [9].

As an application, in Section 4 we prove a lower bound of $\Omega(n/\log^3(n))$ on the number of calls made by any fully black-box construction of a universal one-way hash function (UOWHF) from a one-way function $f : \{0,1\}^n \to \{0,1\}^n$. This can be further improved to $\Omega(n/\log(n))$ (see Section 5 in [3]).

UOWHFs are a fundamental cryptographic primitive, most notably used for obtaining digital signatures. They were studied extensively since their introduction by Naor and Yung [12], who showed a simple construction that makes only one call to the underlying one-way function whenever, additionally, the function is a permutation. Rompel [14] showed a construction based on any one-way function, and the most efficient construction based on general one-way functions is due to Haitner *et. al.* [8]. Their construction makes $\tilde{O}(n^6)$ calls to a one-way function $f : \{0,1\}^n \to \{0,1\}^n$. Note that the bound given in [5] does not say anything for the mere construction of a UOWHF (e.g., for a function which

compresses one bit), and prior to our work it would have been possible to conjecture that there exists a construction of a UOWHF from a general one-way function that makes only one call to the underlying one-way function. Our bound matches exactly and up to a log-factor the number of calls made by the constructions of [4] and [1], respectively.

Our result can be understood as an analog to that of Holenstein and Sinha, who show a bound of $\Omega(n/\log(n))$ on the number of calls to a one-way function that are made by a construction of a pseudo-random generator. We observe (details are omitted) that the recent result of [9] can be explained in our framework. Our characterization of UOWHFs (presented in Section 4.1) is inspired by their characterization of pseudo-random generators. For some candidate constructions, our proof also utilizes their BreakOW oracle. Our main technical contribution in Section 4.2 is the oracle BreakPI and the proof that it satisfies the conditions of our theorem from Section 3.

2 Preliminaries

2.1 The Computational Model

A function $p = p(\rho)$ is polynomial if there exists a value c such that $p(\rho) = \rho^c$. A machine M is efficient if there exists a polynomial p such that on every input $x \in \{0,1\}^*$, $M(x)$ halts after at most $p(|x|)$ steps. A function $s : \mathbb{N}^+ \to \mathbb{N}^+$ is a security function if for every $\rho \in \mathbb{N}^+$ it holds that $s(\rho+1) \geq s(\rho)$, and s is efficiently computable (i.e., there exists an efficient machine M that on input 1^ρ outputs $s(\rho)$). For a security function s we define $\frac{1}{s} : \mathbb{N}^+ \to \mathbb{R}^+$ as $\frac{1}{s}(\rho) \stackrel{\text{def}}{=} \frac{1}{s(\rho)}$. A function $f : \mathbb{N}^+ \to \mathbb{R}^+$ is negligible if for all polynomial security functions p it holds that $f(\rho) < \frac{1}{p(\rho)}$ for all large enough ρ.

An (n, n')-oracle circuit $C^{(?)}$ is a circuit that contains special oracle gates of input length n and output length n'. An (n_1, n'_1, n_2, n'_2)-oracle circuit $C^{(?)}$ is a circuit that contains two types of oracle gates, where the ith type contains n_i input gates and n'_i output gates.

A family of functions $f = \{f_\rho\}_{\rho \in \mathbb{N}^+}$ is uniformly efficiently computable if there exists an efficient machine M such that for every $\rho \in \mathbb{N}^+$ it holds that $M(1^{(\rho)})$ outputs a circuit that implements f_ρ. A non-uniform algorithm $A = \{A_\rho\}_{\rho \in \mathbb{N}^+}$ is a parameterized family of circuits A_ρ. A non-uniform algorithm A implements the parametrized functions family $f = \{f_\rho\}_{\rho \in \mathbb{N}^+}$, if each A_ρ implements f_ρ.

A non-uniform oracle algorithm $A^{(?)} = \{{A^{(?)}}_\rho\}_{\rho \in \mathbb{N}^+}$ is a parameterized family of oracle circuits. Let $A^{(?)}$ be an oracle algorithm. A parametrized

family of functions $f = \{f_\rho\}_{\rho\in\mathbb{N}^+}$ (resp., an algorithm $B = \{B_\rho\}_{\rho\in\mathbb{N}^+}$) is compatible with $A^{(?)}$ if for all $\rho > 0$ it holds that f_ρ (resp., B_ρ) is compatible with A_ρ. In this case we define the algorithm $A^{(f)} \stackrel{\text{def}}{=} \{A_\rho^{f_\rho}\}$ (resp., $A^{(B)} \stackrel{\text{def}}{=} \{A_\rho^{B_\rho}\}$).

Uniform generation of oracle algorithms. The construction and reduction algorithms in fully black-box constructions are assumed to work for any[1] input/output lengths of the primitive and breaker functionalities, and therefore are modeled in the following way: In addition to the security parameter ρ, both the construction and the reduction algorithms take as input information about the input/output lengths of the underlying primitive f_ρ and the breaker algorithm B_ρ.

A uniform oracle algorithm is a machine M that on input $M(1^\rho, n(\rho), n'(\rho))$ outputs an $(n(\rho), n'(\rho))$-oracle circuit $A_\rho^{(?)}$. For a uniform oracle algorithm M and a parameterized family of functions $f = \{f_\rho : \{0,1\}^{n(\rho)} \to \{0,1\}^{n'(\rho)}\}_{\rho\in\mathbb{N}^+}$, define $M^{(f)} \stackrel{\text{def}}{=} \{A_\rho^{(f_\rho)}\}_{\rho\in\mathbb{N}^+}$, where $A_\rho^{(?)} \stackrel{\text{def}}{=} M(1^\rho, n(\rho), n'(\rho))$. For a non-uniform algorithm A, the family $M^{(A)}$ is defined analogously.

Let $s = s(\rho)$ be a security function. An s-non-uniform two oracle algorithm is a machine M such that for every $\rho, n_1, n_1', n_2, n_2' \in \mathbb{N}^+$ and every $a \in \{0,1\}^{s(\rho)}$, it holds that $M(1^\rho, n_1, n_1', n_2, n_2', a)$ outputs an (n_1, n_1', n_2, n_2')-two oracle circuit $A_{\rho,a}^{(?,?)}$ with at most $s(\rho)$ oracle gates. Note that the last requirement is essential and is implicit in the case of an efficient uniform oracle algorithm, where the number of oracle gates is bounded by the polynomial that bounds the running time of the algorithm. For an s-non-uniform two oracle algorithm M, a non-uniform algorithm B and a family of functions f, we formally define $M^{[B,f]} \stackrel{\text{def}}{=} (M, B, f)$.

2.2 Modeling Cryptographic Primitives

In order to state our results in their full generality, and in particular to exclude reductions that are allowed to use non-uniformity and are considered successful in inverting the one-way function even if they invert only a negligible fraction of the inputs of the function, the following two

[1] A-priori, for a fixed security parameter ρ there is no bound on the input length the construction is expected to work, as long as the series of the input-output lengths is bounded by *some* polynomial.

definitions are very general, and extend Definitions 2.1 and 2.3 from [13]. The example of modeling a one-way function follows the definition.

Definition 1 (Cryptographic Primitive). *A primitive* $\mathbf{Q}$ *is a pair* $\langle F_{\mathbf{Q}}, R_{\mathbf{Q}} \rangle$*, where* $F_{\mathbf{Q}}$ *is a set of parametrized families of functions* $f = \{f_\rho\}_{\rho \in \mathbb{N}^+}$ *and* $R_{\mathbf{Q}}$ *is a relation over triplets* $\langle f_\rho, C, \epsilon \rangle$ *of a function* $f_\rho \in f$ *(for some* $f \in F_{\mathbf{Q}}$*), a circuit* C *and a number* $\epsilon > 0$*. We define that* C $(\mathbf{Q}, \epsilon)$-breaks f_ρ *if and only if* $\langle f_\rho, C, \epsilon \rangle \in R_{\mathbf{Q}}$*.*

The set $F_{\mathbf{Q}}$ specifies all the correct implementations (not necessarily efficient) of $\mathbf{Q}$ and the relation $R_{\mathbf{Q}}$ captures the security property of $\mathbf{Q}$, that is, it specifies for every concrete security parameter implementation, how well a breaker algorithm performs with respect to the security property of the primitive.

Finally, let $s = s(\rho)$ be a security function, $B = \{B_\rho\}_{\rho \in \mathbb{N}^+}$ be a non-uniform algorithm, and $f \in F_{\mathbf{Q}}$. We say that B $(\mathbf{Q}, \frac{1}{s})$-breaks f if $\langle f_\rho, B_\rho, \frac{1}{s(\rho)} \rangle \in R_{\mathbf{Q}}$ for infinitely many values ρ. Let us fix an s-non-uniform two oracle algorithm R. We say that $R^{[B,f]}$ $(\mathbf{Q}, \frac{1}{s})$-breaks f if for infinitely many values ρ there exists an $a \in \{0,1\}^{s(\rho)}$ (called advice) such that $\langle f_\rho, R_{\rho,a}^{(B_\rho, f_\rho)}, \frac{1}{s(\rho)} \rangle \in R_{\mathbf{Q}}$, where $R_{\rho,a}^{(?,?)} = R(1^\rho, n, n', b, b', a)$.

The usual notion of polynomial security of a primitive is captured by the following definition: B $\mathbf{Q}$-breaks f if there exists a polynomial $p = p(\rho)$ such that B $(\mathbf{Q}, \frac{1}{p})$-breaks f.

A primitive $\mathbf{Q}$ exists if there exists an efficient uniform algorithm M that implements an $f \in F_{\mathbf{Q}}$, and for every efficient uniform algorithm M' that, on input 1^ρ outputs a circuit, it holds that $\{M'(1^\rho)\}_{\rho \in \mathbb{N}^+}$ does not $\mathbf{Q}$-break f.

Observe that the requirement that M' outputs a circuit is made without loss of generality and captures the standard definition of an efficient randomized machine M' that breaks a primitive. Given such an M' that tosses at most $r = r(\rho)$ random coins, there exists[2] a (now deterministic) efficient uniform machine M'' that on input 1^ρ outputs a circuit C_ρ with $m(\rho) + r(\rho)$ input gates and $n(\rho)$ output gates that computes the output of M for all strings of length $m(\rho)$, and therefore $\mathbf{Q}$-breaks the primitive.

2.3 One-Way Functions

Our model for describing a primitive is very general and captures the security properties of many cryptographic primitives. As an example, we

[2] For example, by the canonical encoding of an efficient machine as in the Cook-Levin Theorem.

bring a standard definition of a one-way function and then explain how it can be described in our model.

Definition 2 (One-Way Function). *A* one-way function $f = \{f_\rho\}_{\rho\in\mathbb{N}^+}$ *is an efficiently uniformly computable family of functions* $f_\rho : \{0,1\}^{n(\rho)} \to \{0,1\}^{m(\rho)}$*, such that for every efficient randomized machine* A*, the function that maps* ρ *to*

$$\Pr_{x\xleftarrow{r}\{0,1\}^{m(\rho)}}\left[A(1^\rho, f_\rho(x)) \in f_\rho^{-1}(f_\rho(v))\right] \text{ is negligible.}$$

In order to model a one-way function (OWF), we set $f = \{f_\rho\}_{\rho\in\mathbb{N}^+} \in F_{\mathbf{OWF}}$, where $f_\rho : \{0,1\}^{n(\rho)} \to \{0,1\}^{m(\rho)}$, if and only if $n = n(\rho)$ and $m = m(\rho)$ are polynomial security functions. We say that $F_{\mathbf{OWF}}$ contains a collection of sets of functions $\mathcal{F} = \{\mathcal{F}_\rho\}_{\rho\in\mathbb{N}^+}$, if for every family $f' = \{f'_\rho\}_{\rho\in\mathbb{N}^+}$, where $f'_\rho \in \mathcal{F}_\rho$ for every ρ, it holds that $f' \in F_{\mathbf{OWF}}$.

In this case, for a function $f_\rho \in f \in F_{\mathbf{OWF}}$, a circuit C that inverts f_ρ on an ϵ-fraction of its inputs, and $\epsilon' > 0$, set $\langle f, C, \epsilon'\rangle \in R_{\mathbf{OWF}}$ if and only if $\epsilon \geq \epsilon'$. The definition is general, and allows for the circuit C to implicitly use randomness. In such a case, for f_ρ as before, a circuit with C with $m(\rho) + r(\rho)$ input bits that computes an output $x \in \{0,1\}^{n(\rho)}$, and a value $\epsilon' > 0$, define $\langle f_\rho, C, \epsilon'\rangle \in R_{\mathbf{OWF}}$ if and only if $\epsilon \geq \epsilon'$, where ϵ is the probability over uniform $z \in \{0,1\}^{r(\rho)}$ and $x \in \{0,1\}^{n(\rho)}$ that $C(f_\rho(x), z)$ outputs an $x' \in f_\rho^{-1}(f_\rho(x))$.

2.4 Fully Black-Box Cryptographic Constructions

Finally, we bring the standard definition of a fixed-parameter fully black-box construction of a primitive $\mathbf{Q}$ from a primitive $\mathbf{P}$, which is usually implicit in the literature. The construction algorithm G is an efficient uniform oracle algorithm and the security reduction R is an efficient uniform two-oracle algorithm. For every security parameter ρ and a function $f_\rho : \{0,1\}^{n(\rho)} \to \{0,1\}^{n'(\rho)}$, G's output on $(1^\rho, n, n')$ is an (n, n')-oracle circuit $g_\rho^{(?)}$ such that $\{g_\rho^{(f_\rho)}\}_{\rho\in\mathbb{N}^+}$ implements $\mathbf{Q}$. The reduction algorithm works as follows: For a security parameter ρ and f as before, and additionally a breaker circuit $B : \{0,1\}^{b(\rho)} \to \{0,1\}^{b'(\rho)}$, the reduction R on input $(1^\rho, n, n', b, b')$ outputs an (n, n', b, b')-two-oracle circuit $R_\rho^{(?,?)}$. The security property property requires that indeed the series of circuits $\{R_\rho^{(B_\rho, f_\rho)}\}_{\rho\in\mathbb{N}^+}$ $\mathbf{P}$-breaks f. We emphasize that the vast majority (if not all) of the constructions of primitives from a one-way function found in the literature are in fact fixed-parameter fully black-box constructions. Formally:

Definition 3 (fixed-parameter fully black-box construction of Q from P). *An efficient uniform oracle algorithm G and an efficient uniform two oracle algorithm R are a* fixed-parameter fully-BB construction *of a primitive $\mathbf{Q} = \langle F_{\mathbf{Q}}, R_{\mathbf{Q}} \rangle$ from a primitive $\mathbf{P} = \langle F_{\mathbf{P}}, R_{\mathbf{P}} \rangle$ if for every $f \in F_{\mathbf{P}}$:*

1. **(correctness)** *$G^{(f)}$ implements $f' \in F_{\mathbf{Q}}$.*
2. **(security)** *For every algorithm B: If B $\mathbf{Q}$-breaks $G^{(f)}$ then $R^{(B,f)}$ $\mathbf{P}$-breaks f.*

For a super-polynomial security function $s = s(\rho)$ (e.g., $s(\rho) = 2^{\sqrt{\rho}}$), the following definition of a fully black-box construction is significantly weaker than the standard one in the following three aspects: First, it requires that reduction only mildly breaks the one-way property of the function f (whenever the breaker breaks the constructed primitive in the standard polynomial sense). Second, the reduction algorithm does not have to be efficient or uniform (but the non-uniformity is limited to an advice of length s). Lastly, it allows the reduction to make s calls to its oracles[3].

Definition 4 (s-weak fixed-parameter fully black-box construction of Q from P).

*A uniform oracle algorithm G and an s-non-uniform two oracle algorithm R are an s-*weak fixed-parameter fully-BB construction *of a primitive $\mathbf{Q} = \langle F_{\mathbf{Q}}, R_{\mathbf{Q}} \rangle$ from a primitive $\mathbf{P} = \langle F_{\mathbf{P}}, R_{\mathbf{P}} \rangle$ if for every $f \in F_{\mathbf{P}}$:*

1. **(correctness)** *$G^{(f)}$ implements an $f' \in F_{\mathbf{Q}}$.*
2. **(security)** *For every non-uniform algorithm B: If B $\mathbf{Q}$-breaks $G^{(f)}$ then $R^{[B,f]}$ $(\mathbf{P}, 1/s)$-breaks f.*

2.5 Random Permutations and Regular Functions

Let n and i be two integers such that $0 \leq i \leq n$. We denote the set of all permutations on $\{0,1\}^n$ by $\mathcal{P}_n$. Let $\mathcal{X}, \mathcal{Y}$ be sets. We denote by $(\mathcal{X} \to \mathcal{Y})$ the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. A function $f : \mathcal{X} \to \mathcal{Y}$ is regular if $|\{x' : f(x) = f(x')\}|$ is constant for all $x \in \mathcal{X}$. A family of functions $f = \{f_\rho\}_{\rho \in \mathbb{N}^+}$ is a regular function if for every ρ the function f_ρ is regular. We denote by $\mathcal{R}_{n,i}$ the set of all regular functions from $\{0,1\}^n$ to itself such that the image of f contains 2^i values. E.g., $\mathcal{R}_{n,n} = \mathcal{P}_n$ is the set of all permutations, and $\mathcal{R}_{n,0}$ is the set of all constant functions.

[3] In Definition 3 the limitation on the number of queries made to the oracles is implicit as R is an efficient algorithm, and so its output circuit has at most a polynomially number of oracle gates.

2.6 Bending a Function and Image Adaptation

It will be useful for us to compare the run of a circuit with oracle access to a function f to a run that is identical except that the output of one specific value is altered.

For a fixed function $f : \{0,1\}^n \to \{0,1\}^n$ and $y', y'' \in \{0,1\}^n$, set

$$f_{(y',y'')}(x) \stackrel{\text{def}}{=} \begin{cases} y'' & \text{if } f(x) = y' \\ f(x) & \text{otherwise.} \end{cases}$$

Similarly, for two fixed functions $f, f' : \{0,1\}^n \to \{0,1\}^n$ and a set $S \subset \{0,1\}^n$, we define the image adaptation [4] of f to f' on S to be the function

$$f_{(S,f')}(x) \stackrel{\text{def}}{=} \begin{cases} f'(x) & \text{if } x \in f^{-1}(f(S)) \\ f(x) & \text{otherwise.} \end{cases}$$

3 A General Theorem for Proving Strong Black-Box Separations

3.1 Deterministic Parametrized Oracles and Local Sets

The following definition allows to model general parameterized oracles, that is, oracles that, for any function f from some set of functions and any q from some query domain, return a value a from some answer set. We observe that many of the oracles used for black-box separations found in the literature could be described in such a way.

Let $\mathcal{X}, \mathcal{Y}, \mathcal{D}$ and $\mathcal{R}$ be sets. A deterministic parametrized oracle for a class of functions $(\mathcal{X} \to \mathcal{Y})$ is an indexed collection $\mathcal{O} = \{\mathcal{O}_f\}_{f \in \mathcal{X} \to \mathcal{Y}}$, where $\mathcal{O}_f : \mathcal{D} \to \mathcal{R}$. We call f, $\mathcal{D}$, and $\mathcal{R}$ the function parameter, the domain, and the range of the oracle, respectively.

Our first example of a deterministic parametrized oracle is the evaluation oracle $\mathcal{E}$ for functions on $\{0,1\}^n$, which on a query q returns the evaluation of f on q. In this case we have that $\mathcal{X} = \mathcal{Y} = \mathcal{D} = \mathcal{R} = \{0,1\}^n$ and $\mathcal{E}_f(q) \stackrel{\text{def}}{=} f(q)$.

The next two definitions capture two important local properties of parametrized oracles. We believe that they are natural and observe that many of the oracles devised for separation results satisfy them.

[4] We mention that if f is a permutation, the condition $f(x) = y$ can be replaced by $x = f^{-1}(y)$, and similarly for $f_{(S,f')}$ check whether $x \in S$, which is what one may expect initially from such a definition.

Intuitively, a determining set is an indexed collection of sets that determine the output of the oracle for every function f and query q in the following sense: If for two functions f and f' it holds that their corresponding oracle outputs differ for some q, then for one of them (f or f') it holds that the local change of an image adaptation of one of the functions to agree with that of the other on its determining set changes the output of the oracle. Formally:

Definition 5. *Let $\mathcal{O}$ be a deterministic parametrized oracle. A* determining set *$\mathcal{I}^{\mathcal{O}}$ for a class of functions $\mathcal{F} \subset (\mathcal{X} \to \mathcal{Y})$ is an indexed collection $\{\mathcal{I}^{\mathcal{O}}_{f,q}\}_{f\in\mathcal{F},q\in\mathcal{D}}$ of subsets of $\mathcal{X}$, such that for every $f, f' \in \mathcal{F}$ and every query $q \in \mathcal{D}$: If $\mathcal{O}_f(q) \neq \mathcal{O}_{f'}(q)$, then it holds that either the image adaptation of f to f' on $\mathcal{I}^{\mathcal{O}}_{f',q}$ changes $\mathcal{O}_f(q)$ (i.e., $\mathcal{O}_{f_{\left(\mathcal{I}^{\mathcal{O}}_{f',q},f'\right)}}(q) \neq \mathcal{O}_f(q)$), or the image adaptation of f' to f on $\mathcal{I}^{\mathcal{O}}_{f,q}$ changes $\mathcal{O}_{f'}(q)$. $\mathcal{I}^{\mathcal{O}}$ is a t-*determining set *if for every function $f \in \mathcal{F}$ and query $q \in \mathcal{D}$ it holds that $|\mathcal{I}^{\mathcal{O}}_{f,q}| \leq t$.*

In the example of the evaluation oracle, we observe that it has a 1-determining set. Indeed, setting $\mathcal{I}^{\mathcal{E}}_{f,q} \stackrel{\text{def}}{=} \{q\}$ satisfies the required definition, since if for any $f, f' \in \{0,1\}^n \to \{0,1\}^n$ and $x \in \{0,1\}^n$ for which $f(x) \neq f'(x)$ it holds that $f_{(\{x\},f')}(x) = f'(x) \neq f(x)$.

Consider an oracle $\mathcal{O}$ with a determining set $\mathcal{I}^{\mathcal{O}}$ for some class of permutations $\mathcal{F}$. Fix $f, f' \in \mathcal{F}$ and $q \in \mathcal{D}$. The following two propositions are immediate from the definition of determining sets:

Proposition 1. *If $\mathcal{O}_f(q) \neq \mathcal{O}_{f'}(q)$ and $f(x) = f'(x)$ for all $x \in \mathcal{I}^{\mathcal{O}}_{f',q}$ (in this case we say that f agrees with f' on $\mathcal{I}^{\mathcal{O}}_{f',q}$), then adapting f' to agree with f on $\mathcal{I}^{\mathcal{O}}_{f,q}$ changes $\mathcal{O}_{f'}(q)$.*

Proposition 2. *If for all $x \in \mathcal{I}^{\mathcal{O}}_{f,q} \cup \mathcal{I}^{\mathcal{O}}_{f',q}$ it holds that $f(x) = f'(x)$ (in this case we say that the functions* agree *on their determining sets), then $\mathcal{O}_f(q) = \mathcal{O}_{f'}(q)$.*

Proposition 2 establishes that determining sets indeed determine the output of the oracle in the following sense: If we know the value $\mathcal{O}_f(q)$ for a query q and a function f, and, moreover, we know that functions f', f agree on their determining sets for q, then this information already determines for us the value $\mathcal{O}_{f'}(q)$.

The next local property of an oracle captures the fact that it is in some sense "stable". For a function f and query q as before, and a value y in

the image set of f, a bending set for f, q, and y is a set of all potentially "sensitive" y' values: For any value y which is not in the image of f on its determining set, and for any value y' which is not in the bending set, the oracle's answer to query q does not change for the local adaptations of f from y' to y. That is, it holds that $\mathcal{O}_{f_{(y',y)}}(q) = \mathcal{O}_f(q)$. Formally:

Definition 6. *Let $\mathcal{O}$ be a deterministic parametrized oracle. A* bending set *$\mathcal{B}^{\mathcal{O}}$ for $\mathcal{F}$ is an indexed collection $\{\mathcal{B}^{\mathcal{O}}_{f,q,y}\}_{f\in\mathcal{F},q\in\mathcal{D},y\in\mathcal{Y}}$ of subsets of $\mathcal{Y}$, such that for every function $f \in \mathcal{F}$, query $q \in \mathcal{D}$, for every target image $y \in \mathcal{Y}$, and for every source image $y' \notin \mathcal{B}^{\mathcal{O}}_{f,q,y}$, it holds that $\mathcal{O}_f(q) = \mathcal{O}_{f_{(y',y)}}(q)$. We say that $\mathcal{B}^{\mathcal{O}}$ is a t-*bending set *if for every function $f \in \mathcal{F}$, query $q \in \mathcal{D}$ and $y \in \mathcal{Y}$ it holds that $|\mathcal{B}^{\mathcal{O}}_{f,q,y}| \leq t$.*

For the example of the evaluation oracle, we observe that it also has a 1-bending set. Setting $\mathcal{B}^{\mathcal{E}}_{f,q,y} \stackrel{\text{def}}{=} \{f(q)\}$ (for the relevant f, q and y) satisfies the required definition. Indeed, for any $y' \neq f(q)$ and $y'' \in \mathcal{Y}$, it holds that $\mathcal{E}_{f_{(y',y'')}}(q) = f_{(y',y'')}(q) = f(q) = \mathcal{E}_f(q)$.

Finally, a deterministic parametrized algorithm $\mathcal{O}$ is t-stable for a class of functions $\mathcal{F}$ if there exist $(\mathcal{I}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}})$ that are a t-determining set and a t-bending set for $\mathcal{F}$, respectively, and at least one of them is not empty.

We note that determining and bending sets always exist unconditionally (just choose the entire domain and range of f, for every determining and bending set, respectively). The challange is finding an oracle that allows to break a primitive and at the same time is t-stable.

3.2 A t-Stable Oracle $\mathcal{O}_f$ Inverts Only a Few Functions

The next lemma, which first appeared in [5] and was subsequently adapted to many other separation results, e.g., [6, 15, 9], establishes an information-theoretic bound on the number of functions an oracle-aided algorithm can invert from a set $\mathcal{F}$ if the oracle is t-stable for $\mathcal{F}$. Essentially, it shows that given an oracle circuit $A^{(?)}$ with access to such an oracle $\mathcal{O}$, it is possible to encode a function $f \in \mathcal{F}$ that A inverts well using significantly fewer bits than $\log(|\mathcal{F}|)$, such that f can still be fully reconstructed, or equivallently, that the encoding is injective.

Lemma 1 (Encoding Lemma). *Let $A^{(?)}$ be an oracle circuit making at most c calls to its oracle, and let $\mathcal{O} = \{\mathcal{O}_f\}_{f\in\{0,1\}^n\to\{0,1\}^n}$ be a deterministic parameterized oracle such that for a class of permutations $\mathcal{F} \subseteq \mathcal{P}_n$ it is t-stable with sets $(\mathcal{I}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}})$. Then, for at most*

$d_n = d_n(c,t) = \left(\binom{2^n}{b}\right)^2 \cdot ((2^n - b)!)$, *where* $b \stackrel{\text{def}}{=} \frac{2^n}{3 \cdot c^2 \cdot t}$, *of the permutations* f *in* $\mathcal{F}$, *it holds that* $\Pr_{x \xleftarrow{r} \{0,1\}^n}\left[A^{\mathcal{O}_f}(f(x)) = x\right] > \frac{1}{c}$.

The proof is a generalized version of the encoding technique of [5].

The next theorem is proved by means of a reduction to Lemma 1 and expressing canonically a regular function using premutations. Detailed proofs of the lemma and the theorem are available in [3].

Theorem 1 (Black-Box Separation Factory). *Let* $s = s(\rho)$ *be a security function, and* $p = p(\rho)$ *be a polynomial function. Let* $(G, A) = (G^{(?)}, A^{(?,?)})$ *be a uniform oracle algorithm and an* s*-non-uniform two-oracle algorithm, respectively. Let* $\mathcal{F} = \{\mathcal{F}_\rho\}_{\rho>0}$, *where* $\mathcal{F}_\rho \subset \mathcal{R}_{n(\rho),i(\rho)}(P_\rho, I_\rho)$, *be contained in* $F_{\mathbf{OWF}}$, *and* $\mathcal{O} = \{\mathcal{O}_\rho\}_{\rho>0}$, *where* $\mathcal{O}_\rho = \{\mathcal{O}_{\rho,f}\}_{f \in \mathcal{F}_{n(\rho),i(\rho)}(P_\rho,I_\rho)}$, *such that for all large enough* ρ*:*

1. $\mathcal{O}_{\rho,f}$ $(\mathbf{Q}, \frac{1}{p(\rho)})$*-breaks* $g_\rho^{(f)}$ *for every* $f \in \mathcal{F}_\rho$, *where* $g_\rho^{(?)} \stackrel{\text{def}}{=} G(1^\rho, n, n')$.
2. $\mathcal{O}_\rho$ *is* t*-stable with sets* $(\mathcal{I}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}})$ *for* $\mathcal{F}_\rho$ *such that* $2^{s(\rho)} \cdot d_i(s(\rho), t) < |\mathcal{F}_\rho|$ *holds, where* d_i *is as in Lemma 1.*

Then (G, A) *is not an* s*-weak fixed-parameter fully black-box construction of* $\mathbf{Q}$ *from* **OWF**.

4 A Lower Bound on the Number of Calls for a Fixed-Parameter Fully Black-Box Construction of UOWHF from OWF

In this section we prove our second main result, namely a lower bound on the number of calls made by the construction algorithm G in any fully black-box construction (G, R) of **UOWHF** from **OWF**. Our bound is achieved by showing a sequence of efficient fixed-parameter fully black-box constructions, where each primitive is constructed from the one that precedes it, and by proving the lower bound on the number of calls a construction makes on the last primitive. A diagram of the reduction sequence is depicted in Figure 1.

4.1 A Characterization of Universal One-Way Hash Functions

Loosely speaking, a universal one-way hash function is a keyed compressing function for which the probability that an adversary wins the following game is very small: First the adversary chooses a preimage v. Then a

random key for the UOWHF is chosen. Finally, the adversary "wins" the game he finds a different preimage v' that maps to the same value under the chosen key. Formally:

Definition 7 (UOWHF). *A* universal one-way hash function $h = \{h_\rho\}_{\rho\in\mathbb{N}^+}$ *is a family of uniformly efficiently computable keyed functions* $h_\rho : \{0,1\}^{\kappa(\rho)} \times \{0,1\}^{m(\rho)} \to \{0,1\}^{m'(\rho)}$ *with* $m'(\rho) < m(\rho)$ *such that for any pair of efficient randomized algorithms* (B_1, B_2) *the function mapping* ρ *to*

$$\Pr_{\substack{(v,\sigma)\xleftarrow{r}B_1(\rho)\\ k\xleftarrow{r}\{0,1\}^{\kappa(\rho)}\\ v'\xleftarrow{r}B_2(k,v,\sigma)}}\left[h_\rho(k,v) = h_\rho(k,v') \wedge v \neq v'\right]$$

is negligible. The family h *is an* ℓ-bit compressing *UOWHF, where* $\ell = \ell(\rho)$*, if* $m(\rho) - m'(\rho) \geq \ell(\rho)$ *for all large enough* ρ*.*

The primitive $\mathbf{UOWHF} = (F_{\mathbf{UOWHF}}, R_{\mathbf{UOWHF}})$ is defined implicitly analogously to the way **OWF** was defined for one-way functions

Domain extension of a UOWHF. The definition of a UOWHF only guarantees that h_ρ is compressing (i.e., it is possible that $\ell(\rho) = 1$). The first reduction we use is a domain extension of a UOWHF, that allows to construct an ℓ-bit compressing UOWHF from a UOWHF. Shoup [16] shows a fully-black box construction of a ℓ-bit compressing UOWHF from one that compresses only one bit, which is the minimal requirement from any UOWHF.

Lemma 2 (UOWHF domain extension). *There exists a fixed-parameter fully black-box construction of an* ℓ*-bit compressing UOWHF* $h'_\rho : \{0,1\}^{\log(\ell)\cdot\kappa(\rho)} \times \{0,1\}^{m+\ell} \to \{0,1\}^m$ *from a one-bit compressing UOWHF* $h_\rho : \{0,1\}^{\kappa(\rho)} \times \{0,1\}^{m+1} \to \{0,1\}^m$*. In order to evaluate* h'_ρ *the construction makes exactly* $\ell(\rho)$ *calls to* h_ρ*. The security reduction* $R_\rho^{h_\rho,B}$ *makes* ℓ *calls to its* h_ρ *oracle, and exactly one call to the breaker* $B_\rho = (B_1, B_2)_\rho$ *oracle. Furthermore, if* B_ρ $(\ell\text{-}\mathbf{UOWHF}, \epsilon)$*-breaks* h'_ρ*, then the reduction* $(\mathbf{UOWHF}, \frac{\epsilon}{\ell})$*-breaks* h_ρ*.*

We observe that the security definition for UOWHFs involves an interaction, and allows the adversary to save its state using σ. It will be more convenient for us to work with an equivalent non-interactive version. The following definition of collision resistance is tightly related to that of a UOWHF by the lemma that follows it, where we denote by $a\|b$ the concatenation of a and b.

Definition 8 (RP-CRHF). *A* random preimage collision resistance hash function *is an efficiently uniformly computable family of functions* $h_\rho : \{0,1\}^{m(\rho)} \to \{0,1\}^{m'(\rho)}$ *with* $m'(\rho) < m(\rho)$*, such that for every efficient randomized machine* B *the function mapping* ρ *to*

$$\Pr_{\substack{v \xleftarrow{r} \{0,1\}^{m(\rho)} \\ v' \xleftarrow{r} B(\rho, v)}} \left[h_\rho(v) = h_\rho(v') \wedge v \neq v' \right] \text{ is negligible.}$$

The family h *is an* ℓ-bit compressing *RP-CRHF, where* $\ell = \ell(\rho)$*, if additionally it holds that* $m(\rho) - m'(\rho) \geq \ell(\rho)$ *for all large enough* ρ*.*

The primitives **RP-CRHF** and $\log^2(\rho)$-**RP-CRHF** are defined analogously.

Lemma 3 (UOWHF to RP-CRHF, folklore). *Let* $h = \{h_\rho\}_{\rho \in \mathbb{N}^+}$ *be a UOWHF. Then the family* $h'_\rho : \{0,1\}^{\kappa(\rho)+m(\rho)} \to \{0,1\}^{\kappa(\rho)+m'(\rho)}$ *given by* $h'_\rho(k \| v) \stackrel{\text{def}}{=} (k \| h_\rho(k, v))$ *is an RP-CRHF.*

Pseudo-injective Functions. Our last reduction establishes that padding the output of a $\log^2(\rho)$-**RP-CRHF** yields a primitive that is both a one-way function, and behaves like an injective function. A pseudo-injective function is an efficiently uniformly computable family $g = \{g_\rho\}_{\rho \in \mathbb{N}^+}$ of length preserving functions $g_\rho : \{0,1\}^{m(\rho)} \to \{0,1\}^{m(\rho)}$ such that for a uniformly chosen input $v \in \{0,1\}^{m(\rho)}$ it is impossible to find another input $v' \neq v$ such that both map to the same value under g_ρ. We stress that pseudo-injective functions exists unconditionally: Any permutation is a pseudo-injective function. Formally:

Definition 9 (Pseudo-Injectivity). *A* pseudo-injective function $g = \{g_\rho\}_{\rho \in \mathbb{N}^+}$ *is a uniformly efficiently computable family of functions* $g_\rho : \{0,1\}^{m(\rho)} \to \{0,1\}^{m(\rho)}$*, such that for all uniform efficient algorithms* A *the function mapping* ρ *to*

$$\Pr_{\substack{v \xleftarrow{r} \{0,1\}^{m(\rho)} \\ v' \xleftarrow{r} A(1^\rho, v)}} \left[g_\rho(v') = g_\rho(v) \wedge v' \neq v \right] \text{ is negligible.}$$

Similarly to before, the primitive $\mathbf{PI} = \langle F_{\mathbf{PI}}, R_{\mathbf{PI}} \rangle$ corresponds to a pseudo-injective function. Next, we consider the primitive $\mathbf{OWF} \wedge \mathbf{PI}$ that corresponds to all functions which are both a one-way function and a pseudo-injective function. Formally, it holds that $f \in F_{\mathbf{OWF} \wedge \mathbf{PI}}$ if and only if $f \in F_{\mathbf{OWF}}$ *and* $f \in F_{\mathbf{PI}}$. For a breaker circuit C, a function

$f_\rho \in f \in F_{\mathbf{OWF}\wedge\mathbf{PI}}$, and a number ϵ it holds that $\langle f_\rho, C, \epsilon\rangle \in R_{\mathbf{OWF}\wedge\mathbf{PI}}$ if and only if $\langle f_\rho, C, \epsilon\rangle \in R_{\mathbf{OWF}}$ *or* $\langle f_\rho, C, \epsilon\rangle \in R_{\mathbf{PI}}$. It turns out that padding any $\log^2(n)$-**RP-CRHF** to a length-preserving function, yields a function which is both a one-way function and a pseudo-injective function.

Lemma 4 ($\log^2(\rho)$-RP-CRHF to OWF$\wedge$PI). *Let $h = \{h_\rho\}_{\rho\in\mathbb{N}^+}$ be an RP-CRHF that compresses $\ell(\rho) \stackrel{\text{def}}{=} m(\rho) - m'(\rho)$ bits, where $\ell(\rho) \geq \log^2(\rho)$. Then the family $\{h'_\rho\}_{\rho\in\mathbb{N}^+}$, where $h'_\rho(v) \stackrel{\text{def}}{=} h_\rho(v)\|0^{\ell(\rho)}$, is a one-way function and a pseudo-injective function.*

Due to limitations of space the proof is omitted. The composition of the constructions depicted in Lemmas 2, 3 and 4 establishes a fixed-parameter fully black-box construction of an $h' \in F_{\mathbf{OWF}\wedge\mathbf{PI}}$ from any $h \in F_{\mathbf{UOWHF}}$ that makes $\log^2(\rho)$ calls to the underlying UOWHF. The security reduction makes $\log^2(\rho)$ calls (to both its oracles) in order to break the security of the underlying UOWHF, and if B $(\mathbf{OWF}\wedge\mathbf{PI}, \frac{1}{p})$-breaks the constructed h for some polynomial p and breaker B, then the reduction $(\mathbf{UOWHF}, \frac{1}{p'})$-breaks h, where p' is a different polynomial such that $p'(\rho) > 4 \cdot p(\rho) \cdot \log^2(\rho)$. Thus we obtain:

Corollary 1. *Suppose that (G, R) is an s'-weak fixed-parameter fully black-box construction of* **UOWHF** *from* **OWF** *that makes at most $r' = r'(\rho)$ queries to* **OWF**. *Then there exists an s-weak fixed-parameter fully black-box construction of* **OWF**$\wedge$**PI** *from* **OWF** *that makes $r'(\rho) \cdot \log^2(\rho)$ calls to the underlying one-way function, where $s(\rho) \stackrel{\text{def}}{=} s'(\rho) \cdot \log^2(\rho)$.*

Therefore, in order to show that there is no s'-weak fixed-parameter fully black-box construction of **UOWHF** from **OWF**, where the construction makes r' calls to the one-way functions, it is sufficient to show that there is no s-weak fixed-parameter fully black-box construction of **OWF**$\wedge$**PI** from **OWF** that makes r calls, where $s(\rho) \stackrel{\text{def}}{=} s'(\rho) \cdot \log^2(\rho)$ and $r(\rho) \stackrel{\text{def}}{=} r'(\rho) \cdot \log^2(\rho)$. This is the goal of the next section.

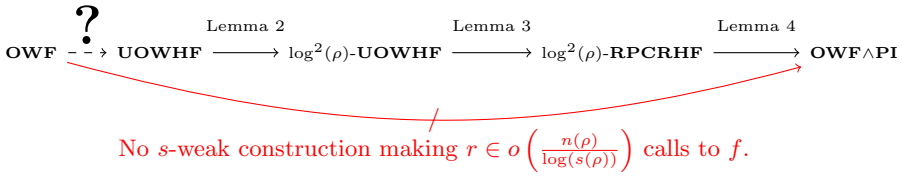

Fig. 1. Fully Black-Box Constructions Diagram.

4.2 A Lower Bound on the Number of Calls for an s-Weak Fixed-Parameter Fully Black-Box Construction of OWF∧PI from OWF

As explained, a lower bound on a construction of **OWF∧PI** from **OWF** yields a very good (up to a $\log^2$-factor) bound on the construction of **UOWHF**. Our proof utilizes the machinery from Section 3. Let us introduce some notation. For an (n,n)-oracle circuit $g^{(?)}:\{0,1\}^m\to\{0,1\}^m$, a function $f:\{0,1\}^n\to\{0,1\}^n$ and a value $v\in\{0,1\}^m$, denote by $X_g(f,v)$ and $Y_g(f,v)$ the sets of queries and answers made to and received from f during the evaluation of $g^{(f)}(v)$, respectively.

For any potential construction (G,R) denote by $r=r(\rho)$ the number of queries g_ρ makes when instantiated for security parameter ρ with a one-way function $f_\rho:\{0,1\}^\rho\to\{0,1\}^\rho$, that is we set $n(\rho)\stackrel{\text{def}}{=}n'(\rho)\stackrel{\text{def}}{=}\rho$. Additionally, let $s=s(\rho)$ be a super-polynomial security function smaller than $2^{\frac{\rho}{10}}$. I.e., for all polynomials p and all large enough ρ it holds that $p(\rho)<s(\rho)<2^{\frac{\rho}{10}}$. We prove that if $r(\rho)<\frac{n(\rho)}{2000\cdot\log(s(\rho))}$ holds for all large enough ρ, then (G,R) is not an s-weak fixed-parameter fully black-box construction of **OWF∧PI** from **OWF**.

Theorem 2. *For all super-polynomial security functions $s=s(\rho)<2^{\frac{\rho}{10}}$ and $r=r(\rho)$ there is no s-weak fixed-parameter fully black-box construction of* **OWF∧PI** *from* **OWF** *such that $g_\rho^{(?)}:\{0,1\}^{m(\rho)}\to\{0,1\}^{m(\rho)}$ makes at most $r(\rho)$ calls to the underlying one-way function, where $n(\rho)\stackrel{\text{def}}{=}n'(\rho)\stackrel{\text{def}}{=}\rho$ and $g_\rho^{(?)}\stackrel{\text{def}}{=}G(1^\rho,n,n')$, and $r(\rho)\leq\frac{n(\rho)}{2000\cdot\log(s(\rho))}$ holds for all large enough ρ.*

Proof. Without loss of generality, we assume that the construction g makes exactly $r(\rho)\stackrel{\text{def}}{=}\frac{n(\rho)}{2000\cdot\log(s(\rho))}$ different queries. Whenever this is not the case, it is always possible to amend G so that it behaves exactly as before, but on input $(1^\rho,n,n')$ it outputs an (n,n')-oracle circuit with $r(\rho)$ oracle gates, and additionally, all queries are different.

Let s and r be a pair of security functions such that s is super-polynomial, that is, for every polynomial p and large enough ρ it holds that $s(\rho)>p(\rho)$, and that $r(\rho)=\frac{n(\rho)}{2000\cdot\log(s(\rho))}$ holds for all sufficiently large ρ.

We now explain how to construct the oracle $\mathcal{O}=\{\mathcal{O}_\rho\}_{\rho\in\mathbb{N}^+}$ and the collection of sets of functions $\mathcal{F}=\{\mathcal{F}_\rho\}_{\rho\in\mathbb{N}^+}$. For each security paramter ρ we define the oracle $\mathcal{O}_\rho$ and the set $\mathcal{F}_\rho$ *independently* of the oracles and function sets chosen for other security parameters. It will always hold that $\mathcal{F}_\rho\subset\{0,1\}^n\to\{0,1\}^n$, and so the constructed $\mathcal{F}$ is contained in $F_{\mathbf{OWF}}$.

Therefore, from now on we omit the security parameter in our notation, but formally all our parameters depend on the security parameter ρ. In particular, $g^{(?)}$ is the construction that the uniform construction algorithm G outputs for security parameter $\rho = n = n'$ with a function $f_\rho : \{0,1\}^\rho \to \{0,1\}^\rho$.

Analogously to [9], for every security parameter we break either the one-wayness property of the constructed function, or its pseudo-injectivity. For the oracle circuit $g^{(?)} : \{0,1\}^m \to \{0,1\}^m$, we check whether when g is evaluated with a random permutation $f \xleftarrow{r} \mathcal{P}_n$ and a random input $v \xleftarrow{r} \{0,1\}^m$, the output $g^f(v)$ is significantly correlated with any subset of the set of oracle answers returned by f on the calls made to it during the evaluation of $g^f(v)$ (recall that these are denoted by $Y_g(f,v)$). To this end, we bring the procedure STA (for safe to answer), which returns true if and only if there is no such correlation:

Procedure STA(w, Q) (on $w \in \{0,1\}^m$ and $Q \subset \{0,1\}^n$ of size r)

---

for all $B \subseteq Q$ **do**
. **if** $\Pr_{f' \xleftarrow{r} \mathcal{P}_n, v' \xleftarrow{r} \{0,1\}^m} \left[g^{(f')}(v') = w \,\middle|\, B \subseteq Y_g(f', v') \right] \geq 2^{-m+\frac{n}{30}}$
. **return** `false`
return `true`

We set $p(g)$, the probability that for a random permutation f and a random input v, the output $g^f(v)$ is correlated with some subset of the answers $Y_g(f,v)$. Define

$$p(g) \stackrel{\text{def}}{=} \Pr_{f \xleftarrow{r} \mathcal{P}_n, v \xleftarrow{r} \{0,1\}^m} \left[\text{STA}(g^{(f)}(v), Y_g(f,v)) \right] \ . \tag{1}$$

We stress that both the output of STA (for any value y and a set Q), and the value $p(g)$ do not depend on any specific permutation, but rather on a combinatorial property of the construction as a whole, which averages over all permutations.

As explained, we set the oracle $\mathcal{O}$ and the set $\mathcal{F}$ based on the value $p(g)$. In case that $p(g) > \frac{1}{2}$ we set the oracle $\mathcal{O} \stackrel{\text{def}}{=} \text{BreakOW}_g \stackrel{\text{def}}{=} \{\text{BreakOW}_{g,f}\}_{f \in \{0,1\}^n \to \{0,1\}^n}$, where we use the oracle BreakOW_g from [9], which is described next. In [9] it is implicitly proved that there exists a set $\mathcal{F} \subset \mathcal{P}_n$ of size $|\mathcal{F}| > \frac{|\mathcal{P}_n|}{5}$, such that $\text{BreakOW}_{g,f}$ $(\mathbf{OWF}, \frac{1}{4})$-breaks g^f for all $f \in \mathcal{F}$, and that BreakOW_g is $2^{\frac{n}{5}}$-stable for $\mathcal{F}$, in which case condition (2) in Theorem 1 is satisfied.

Algorithm $\mathrm{BreakOW}_{g,f}(w)$ (on input $w \in \{0,1\}^m$)

for all $v \in \{0,1\}^m$ **do**
. **if** $g^{(f)}(v) = w$ **then**
. **if** $\mathrm{STA}(w, Y_g(f,v))$ **then**
. **return** v
return $\perp$

In the case $p(g) \leq \frac{1}{2}$ we show that when f is chosen uniformly at random from a set of regular degenerate functions, it is often the case that the construction $g^{(f)}$ is not injective, and therefore there exists an oracle which breaks the pseudo-injectivity of $g^{(f)}$. The challenge is to find a breaker oracle that is t-stable. The next lemmas establish that the oracle BreakPI satisfies the required conditions in this case.

Formally, for a construction circuit g we define the oracle $\mathrm{BreakPI}_g = \{\mathrm{BreakPI}_{g,f}\}_{f \in \{0,1\}^n \to \{0,1\}^n}$ that for a function f is given by:

Algorithm $\mathrm{BreakPI}_{g,f}(v)$ (on input $v \in \{0,1\}^m$)

for all $v' \in \{0,1\}^m$ **do**
. **if** $g^{(f)}(v) = g^{(f)}(v')$ **and** $v' \neq v$ **then**
. **if** $Y_g(f,v) = Y_g(f,v')$ **then**
. **return** v'
return $\perp$

Now, we fix $i \stackrel{\text{def}}{=} \frac{n}{200 \cdot r} = 10 \cdot \log(s)$. We show that for a $\frac{1}{6}$-fraction of the functions f in $\mathcal{R}_{n,i}$ it holds that $\mathrm{BreakPI}_{g,f}$ breaks the pseudo-injectivity of $g^{(f)}$.

Lemma 5. *Let $g : \{0,1\}^m \to \{0,1\}^m$ be an r-query oracle construction with $p(g) \leq \frac{1}{2}$. Then for a $\frac{1}{6}$-fraction of the functions in $\mathcal{R}_{n,\frac{n}{200 \cdot r}}$ it holds that*

$$\Pr_{v \xleftarrow{r} \{0,1\}^m} \left[\mathrm{BreakPI}_{g,f}(v) \text{ outputs } v' \text{ s.t. } v \neq v' \wedge g^{(f)}(v) = g^{(f)}(v') \right] \geq \frac{1}{24}. \tag{2}$$

The proof of the Lemma appears in [3]. We conclude from Lemma 5 that if $p(g) \leq \frac{1}{2}$, there exists a partition P of $\{0,1\}^n$ to sets of size 2^{n-i} and an image-set I of size 2^i, such that (2) holds for at least a $\frac{1}{6}$-fraction of the functions $f \in \mathcal{R}_{n,i}(P,I)$. Set $\mathcal{F} \subset \mathcal{R}_{n,i}(P,I)$ to be the set of all functions for which (2) holds. It follows that $|\mathcal{F}| \geq \frac{1}{6} \cdot 2^i$, as $|\mathcal{R}_{n,i}(P,I)| = |\mathcal{P}_i|$.

We next show that for the class of functions $\mathcal{R}_{n,i}(P,I)$ the oracle can be implemented such that it is stable.

Lemma 6. *Let $i \in \mathbb{N}^+$ and $I \subset \{0,1\}^n$ of size 2^i and P a partition of $\{0,1\}^n$ to sets of size 2^{n-i}. Then there exists an implementation of the oracle* BreakPI_g *that is n-stable for $\mathcal{R}_{n,i}(P,I)$.*

The proof of the Lemma appears in [3]. It is left to check (the simple calculation is omitted) that $2^s \cdot d_i(s,n) < |\mathcal{F}|$.

We have shown that the conditions of Theorem 1 hold, and therefore we conclude that there is no s-weak fixed-parameter fully black-box construction of **OWF**$\wedge$**PI** from **OWF**. The theorem is proved. □

4.3 Deriving the Lower Bound

We are now ready to derive our lower bound for constructions of a universal one-way hash function from a one-way function:

Corollary 2. *Let s' be a security function such that $s(n) \stackrel{\text{def}}{=} s'(n) \cdot \log^2(n)$ is a super-polynomial security function for which $s(n) < 2^{\frac{n}{10}}$ holds. Then there is no s-weak fixed-parameter fully black-box construction of* **UOWHF** *from* **OWF**, *where the construction makes at most $r'(n) = \frac{n}{2000 \cdot \log(s(n)) \cdot \log^2(n)}$ calls to a one-way function $f = \{f_n : \{0,1\}^n \to \{0,1\}^n\}_{n \in \mathbb{N}^+}$.*

Proof. We apply Corollary 1 with Theorem 2. □

Corollary 3. *There is no fixed-parameter fully black-box construction of* **UOWHF** *from* **OWF**, *where the construction makes at most $r = r(n)$ calls to a OWF $f = \{f_n : \{0,1\}^n \to \{0,1\}^n\}_{n \in \mathbb{N}^+}$, where $r \in o\left(\frac{n}{\log^3(n)}\right)$.*

Proof. Let $r \in o\left(\frac{n}{\log^3(n)}\right)$. Then there exists a super-constant function $\alpha = \alpha(n)$, such that the function $r'(n)$ given by $r'(n) \stackrel{\text{def}}{=} r(n) \cdot \alpha(n)$ is still in $o\left(\frac{n}{\log^3(n)}\right)$. The bound follows immediately from Corollary 2 applied with $s(n) \stackrel{\text{def}}{=} 2^{\alpha(n) \cdot \log(n)}$. □

Acknowledgments. We thank the anonymous reviewers for their helpful comments.

References

1. Scott Ames, Rosario Gennaro, and Muthuramakrishnan Venkitasubramaniam. The generalized randomized iterate and its application to new efficient constructions of uowhfs from regular owfs. In *ASIACRYPT*, LNCS 7658, pp. 154–171. Springer, 2012.
2. Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pp. 106–115. IEEE Computer Society, 2001.
3. Kfir Barhum and Thomas Holenstein. A Cookbook for Black-Box Separations and a Recipe for UOWHFs. Full version available as ECCC report TR12-173.
4. Kfir Barhum and Ueli Maurer. UOWHFs from OWFs: Trading regularity for efficiency. In *LATINCRYPT 2012*, LNCS 7533, pp. 234–253. Springer, 2012.
5. Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM J. Comput.*, 35(1):217–246, 2005.
6. Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *FOCS*, pp. 669–679. IEEE Computer Society, 2007.
7. Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, LNCS 5444, pp. 202–219. Springer, 2009.
8. Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *EUROCRYPT*, LNCS 6110, pp. 616–637. Springer, 2010.
9. Thomas Holenstein and Makrand Sinha. Constructing a pseudorandom generator requires an almost linear number of calls. *CoRR*, abs/1205.4576. 2012.
10. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pp. 44–61. ACM, 1989.
11. Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *FOCS*, pp. 535–542. IEEE Computer Society, 1999.
12. Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pp. 33–43. ACM, 1989.
13. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, LNCS 2951, pp. 1–20. Springer, 2004.
14. John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pp. 387–394. ACM, 1990.
15. Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM J. Comput.*, 39(7):3058–3088, 2010.
16. Victor Shoup. A composition theorem for universal one-way hash functions. In *EUROCRYPT*, LNCS 1807, pp. 445–452. Springer, 2000.
17. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT*, LNCS 1403, pp. 334–345. Springer, 1998.
18. Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pp. 80–91. IEEE Computer Society, 1982.