Poly-Many Hardcore Bits for Any One-Way Function

Mihir Bellare[1] Stefano Tessaro[2]

December 27, 2013

Abstract

We show how to extract an arbitrary polynomial number of simultaneously hardcore bits from any one-way function. Our construction is based on differing-input obfuscation.

[1] Department of Computer Science & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: mihir@eng.ucsd.edu. URL: http://cseweb.ucsd.edu/~mihir/. Supported in part by NSF grants CNS-0904380, CCF-0915675, CNS-1116800 and CNS-1228890.

[2] Department of Computer Science, University of California Santa Barbara, Santa Barbara, California 93106, USA. Email: tessaro@cs.ucsb.edu. URL: http://www.cs.ucsb.edu/~tessaro/.

Contents

1 Introduction

Let f be a one-way function, and h a function that has the same domain as f. We say that h is hardcore for f if the distributions $(f, h, f(x), h(x))$ and $(f, h, f(x), r)$ are computationally indistinguishable when x is chosen at random from the domain of f and r is a random $|h(x)|$-bit string.[1] We will refer to the output length of h, which is the number of (simultaneously) hardcore bits produced by h, as the *span* of h, and say that f achieves a certain span if there exists a hardcore function h for f with the span in question. Hardcore predicates are hardcore functions with span one.

Introduced in [10, 33, 20], hardcore functions have played a central role in the theory of cryptography. Historically, their motivating application was to achieve semantically-secure public-key encryption [20]. Here the public-key of the encryption scheme is a trapdoor, one-way permutation f together with hardcore function h, and the ciphertext encrypting message m is $(f(x), h(x) \oplus m)$ for random x. The span thus determines the number of message bits that can be encrypted. Since then, many other usages have emerged.

Our work presents hardcore functions with *arbitrary polynomial span* for *arbitrary one-way functions*, resolving a problem that had remained open, despite significant effort, since the 1980s. The tool we use is differing-input obfuscation (diO), an extension of indistinguishability obfuscation (iO) [5] introduced by Ananth, Boneh, Garg, Sahai and Zhandry (ABGSZ) [4].

Prior work. Early results gave hardcore predicates (ie. span one) for specific one-way functions including discrete exponentiation modulo a prime, RSA and Rabin [10, 33, 31, 9, 24, 3, 27, 21, 16]. Eventually, in an impactful and influential work, Goldreich and Levin [19] gave a hardcore predicate for any one-way function. Extensions of these results are able to achieve logarithmic span [32, 26, 19, 1, 15].

Hardcore functions with polynomial span[2] have been provided for specific, algebraic functions including by Håstad, Schrift and Shamir [22] for discrete exponentiation modulo a composite, by Catalano, Gennaro and Howgrave-Graham [14] for the Pailler function [28], and by Akavia, Goldwasser and Vaikuntanathan [2] for certain LWE-based functions. Peikert and Waters showed that lossy trapdoor functions [29] achieve polynomial span, yielding further examples of specific one-way functions with polynomial span [29, 17].

The basic question that remained open from this work was whether polynomial span is achievable for *any* one-way function. One answer was provided by [6], who showed that UCE-security (specifically, relative to split, computationally-unpredictable sources) of a function h with polynomial span suffices for h to be hardcore for any one-way function, but the assumption made is close to the desired conclusion and an instantiation of h would simply have to assume UCE-security, no functions achieving this being known under more standard assumptions.

Constructions and results. Let G be a PRF [18] and let $h(x) = \mathsf{G}(gk, x)$ where gk is a random key for G that is embedded in the description of h. The distributions $(f(x), h(x))$ and $(f(x), r)$ are then indistinguishable when r is a random string of length $|h(x)|$, but not the distributions $(f, h, f(x), h(x))$ and $(f, h, f(x), r)$ as required for a secure hardcore function, because the description of h contains the PRF key, revealing which compromises the security of the PRF. Our first construction is the natural one, namely to let h be an obfuscation of the circuit $\mathsf{G}(gk, \cdot)$. The difficulty is to show that this works assuming an achievable form of obfuscation. As explained

[1] In the formal definitions in Section 2, both one-way functions and hardcore functions are families. Think here of f, h as instances chosen at random from the respective families, their descriptions public.

[2] Once one can obtain a particular polynomial number of output bits, one can always expand to an arbitrary polynomial via a PRG, so we do not distinguish these cases.

in more depth below, we are able to resolve this difficulty by an extension of the Sahai-Waters technique [30] in which we couple punctured PRFs [11, 25, 13] with diO-security rather than iO-security. However, this requires that f is injective. This yields our first result, a construction of a hardcore function with polynomial span for any injective one-way function.

For the case of a general (arbitrary) one-way function, we modify the construction so that $\mathsf{G}(gk, \cdot)$ is applied, not to x, but to $f(x)$. This at first sounds insecure, because if a circuit doing this was provided, even obfuscated, an adversary knowing $f(x)$ could provide it to the circuit and get back the hardcore bits $h(x)$. Our circuit, however, will take input x rather than $f(x)$, itself computing the latter and returning the result on it of $\mathsf{G}(gk, \cdot)$. The proof, again expanded on below, makes crucial use of diO-security. This yields our second result, a construction of a hardcore function with polynomial span for any one-way function.

Beyond the one-wayness of f, both results assume nothing other than a diO-secure obfuscator. (Punctured PRFs are not an extra assumption since we are already assuming a one-way function.) In fact, by exploiting a result of Boyle, Chung and Pass (BCP) [12], the assumption for the first result can be reduced from diO to plain iO. We do not know how to do this for the second result.

A CLOSER LOOK. We now take a closer look at the proofs to highlight the technical difficulties and new techniques. Recall that the guarantee of iO [5, 30] is that the obfuscations of two circuits are indistinguishable if the circuits themselves are equivalent, meaning return the same output on all inputs. Differing-input obfuscation [4] relaxes the equivalence condition, asking instead that it only be hard, given the (unobfuscated) circuits, to find an input where they differ. See Section 2 for formal definitions. Both our proofs are sequences of hybrids in which we exploit obfuscation security several times. Only one, crucial one of these steps will use diO, the rest relying just on iO.

Recall that in the injective case, h is an obfuscation of the circuit $\mathsf{G}(gk, \cdot)$ where gk is a random key for punctured PRF G, so that $h(x) = \mathsf{G}(gk, x)$. We consider an adversary $\mathcal{H}$ provided with $f, h, f(x^*), r^*$ and want to move from the real game, in which $r^* = \mathsf{G}(gk, x^*)$, to the random game, in which r^* is random. We begin by using the SW technique [30] to move to a game in which r^* is random and h is an obfuscation of the circuit C^1 that embeds the target input x^*, a punctured PRF key, and a random point r^*, returning the latter when called on $x = x^*$ (the trigger) and otherwise returning $\mathsf{G}(gk, x)$, computed via the punctured key. (This move relies on iO and punctured PRF security, and does not require diO.) While this has made r^* random as desired, h is not what it should be in the random game, where it is in fact an obfuscation of the real circuit $\mathsf{G}(gk, \cdot)$. The difficulty is to move h back to an obfuscation of this real circuit while leaving r^* random. We realize that such a move must exploit the one-wayness of f, which has not so far been used. A one-wayness adversary, given $f(x^*)$ and aiming to find x^*, needs to run $\mathcal{H}$. The problem is that $\mathcal{H}$ needs an obfuscation of the above-described circuit C^1 as input, and construction of C^1 requires knowing the very point x^* that the one-wayness adversary is trying to find. The difficulty is inherent rather than merely one of proof, for the forms of obfuscation being used give no guarantee that an obfuscation of C^1 does not reveal x^*. We get around this by changing the trigger check from $x = x^*$ to $f(x) = f(x^*)$, so that now the circuit can embed $f(x^*)$ rather than x^*, a quantity there is no harm in revealing. The new check is equivalent to the old if f is injective, which is where we use this assumption. But we have still not arrived at the random game. We note that our modified circuit C^2 and the target circuit $\mathsf{G}(gk, \cdot)$ of the random game are inherently non-equivalent, and iO would not apply. However, these circuits differ only at input x^*. We exploit the one-wayness of f to prove that it is hard to find this input even given the two circuits. The assumed diO-security of the obfuscator now implies that the obfuscations of these circuits are indistinguishable, allowing us to conclude. Finally, since we exploit diO only for circuits that differ at one (hard to find) point, BCP [12] says that iO in fact suffices, making iO the only assumption needed for the result beyond the necessary one-wayness of f.

The above argument makes crucial use of the assumption that f is injective. To handle an arbitrary one-way function, we modify the construction so that h is an obfuscation of the circuit $\mathsf{G}(gk, f(\cdot))$ where gk is a random key for punctured PRF G. Thus, $h(x) = \mathsf{G}(gk, f(x))$. We consider an adversary $\mathcal{H}$ provided with $f, h, f(x^*), r^*$ and want to move from the real game, in which $r^* = \mathsf{G}(gk, f(x^*))$, to the random game, in which r^* is random. We begin, as before, by using iO and punctured PRF security to move to a game in which r^* is random and h is an obfuscation of the circuit C^1 that embeds $y^* = f(x^*)$, a punctured PRF key, and a random point r^*, returning the latter when called on x such that $f(x) = y^*$ (the trigger) and otherwise returning $\mathsf{G}(gk, f(x))$, computed via the punctured key. Having made r^* random, we now need to revert h back to an obfuscation of the real circuit $\mathsf{G}(gk, f(\cdot))$. But C^1 and this real circuit differ only on inputs x that are pre-images of y^* under f. We use the one-wayness of f to show that it is hard to find any such differing input from the circuits, and then invoke diO-security of the obfuscator to conclude. This time we cannot invoke BCP [12] to say that iO suffices because the final circuits could differ on exponentially-many inputs, so our assumption remains diO.

Discussion and related work. Random oracles (ROs) are "ideal" hardcore functions, able to provide polynomial span for any one-way function. This fact underlies, and is implicit in, the BR93 ROM public-key encryption scheme [7]. Our results, akin to [6, 23], can thus be seen as instantiating the RO in a natural ROM construction, in particular showing hardcore functions in the standard model that are just as good as those in the ROM. As a consequence, we are able to instantiate the RO in the BR93 scheme to obtain a standard-model scheme.

Extractable iO (xiO), introduced by BCP [12], implies diO [4], and hence suffices for our results. The two notions are very close, the difference being that diO can be seen as xiO allowing non blackbox extractors, but diO is defined in a way that does not explicitly talk of extraction, which we have found convenient in our proofs.

Interestingly, our second construction is non blackbox, in that the hardcore function depends on f and uses a circuit for computing f, reminescent of HSW [23]. The hardcore function in our first result is universal, meaning h does not depend on f, another reason the first result is not subsumed by the second. (The first reason is that the first result can rely only on iO while the second requires diO.) Also interestingly, the hardcore function constructed in our second result is the reverse of the hash function used to instantiate FDH in HSW [23]: in our case, the circuit being obfuscated first applies a one-way function and then a punctured PRF, while in their case it first applies a punctured PRF and then a one-way function.

Our work adopts the standard definition of a one-way function in which any polynomial-time adversary must have negligible inversion advantage. Polynomial span is known to be achievable for any *exponentially* hard to invert function [19] .

Inversion in the standard definition of a one-way function is on images of random inputs. Our results apply also when the distribution of inputs on which the function is one-way is arbitrary.

2 Preliminaries

We recall definitions for one-way functions, hardcore predicates, punctured PRFs and relevant variants of indistinguishability obfuscation.

Notation. We denote by $\lambda \in \mathbb{N}$ the security parameter and by 1^λ its unary representation. We denote the size of a finite set X by $|X|$, and the length of a string $x \in \{0,1\}^*$ by $|x|$. We let ε denote the empty string. If X is a finite set, we let $x \leftarrow\$ X$ denote picking an element of X uniformly at random and assigning it to x. Algorithms may be randomized unless otherwise indicated. Running

Game $\mathrm{OW}_{\mathsf{F}}^{\mathcal{F}}(\lambda)$	Game $\mathrm{HC}_{\mathsf{F},\mathsf{H}}^{\mathcal{H}}(\lambda)$	Game $\mathrm{PPRF}_{\mathsf{G}}^{\mathcal{G}}(\lambda)$
$fk \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$	$b \leftarrow_{\$} \{0,1\}$	$b \leftarrow_{\$} \{0,1\}; gk \leftarrow_{\$} \mathsf{G.Kg}(1^\lambda); b' \leftarrow_{\$} \mathcal{G}^{\mathrm{CH}}(1^\lambda)$
$x^* \leftarrow_{\$} \{0,1\}^{\mathsf{F.il}(\lambda)}$	$fk \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$; $hk \leftarrow_{\$} \mathsf{H.Kg}(1^\lambda, fk)$	Return $(b = b')$
$y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$	$x^* \leftarrow_{\$} \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$	$\underline{\mathrm{CH}(x^*)}$
$x' \leftarrow_{\$} \mathcal{F}(1^\lambda, fk, y^*)$	If $b = 1$ then $r^* \leftarrow \mathsf{H.Ev}(hk, x^*)$	$gk^* \leftarrow_{\$} \mathsf{G.PKg}(1^\lambda, gk, x^*)$
Return $(x^* = x')$	Else $r^* \leftarrow_{\$} \{0,1\}^{\mathsf{H.ol}(\lambda)}$	If $b = 1$ then $r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$
	$b' \leftarrow_{\$} \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$	Else $r^* \leftarrow_{\$} \{0,1\}^{\mathsf{G.ol}(\lambda)}$
	Return $(b = b')$	Return (gk^*, r^*)

Game $\mathrm{DIFF}_{\mathcal{S}}^{\mathcal{D}}(\lambda)$	Game $\mathrm{IO}_{\mathsf{Obf},\mathcal{S}}^{\mathcal{O}}(\lambda)$
$b \leftarrow_{\$} \{0,1\}; (\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow_{\$} \mathcal{S}(1^\lambda)$	$b \leftarrow_{\$} \{0,1\}$; $(\mathrm{C}_0, \mathrm{C}_1, aux) \leftarrow_{\$} \mathcal{S}(1^\lambda)$
$x \leftarrow_{\$} \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$	$\overline{\mathrm{C}} \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathrm{C}_b)$; $b' \leftarrow_{\$} \mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$
Return $(\mathrm{C}_0(x) \neq \mathrm{C}_1(x))$	Return $(b = b')$

Figure 1: **Games defining one-wayness of F, security of H as a hardcore function for F, punctured-PRF security of G, difference-security of circuit sampler $\mathcal{S}$ and iO-security of obfuscator Obf relative to circuit sampler $\mathcal{S}$.**

---

time is worst case. "PT" stands for "polynomial-time," whether for randomized algorithms or deterministic ones. If A is an algorithm, we let $y \leftarrow A(x_1, \ldots; r)$ denote running A with random coins r on inputs $x_1, \ldots$ and assigning the output to y. We let $y \leftarrow_{\$} A(x_1, \ldots)$ be the resulting of picking r at random and letting $y \leftarrow A(x_1, \ldots; r)$. We let $[A(x_1, \ldots)]$ denote the set of all possible outputs of A when invoked with inputs $x_1, \ldots$. We say that $f : \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial p, there exists $n_p \in \mathbb{N}$ such that $f(n) < 1/p(n)$ for all $n > n_p$. We use the code based game playing framework of [8]. (See Fig. 1 for examples of games.) By $\mathrm{G}^{\mathcal{A}}(\lambda)$ we denote the event that the execution of game G with adversary $\mathcal{A}$ and security parameter λ results in the game returning true.

Function families. A family of functions F specifies the following. PT key generation algorithm $\mathsf{F.Kg}$ takes 1^λ and possibly another input to return a key $fk \in \{0,1\}^{\mathsf{F.kl}(\lambda)}$, where $\mathsf{F.kl}\colon \mathbb{N} \to \mathbb{N}$ is the key length function associated to F. The deterministic, PT evaluation algorithm $\mathsf{F.Ev}$ takes key fk and an input $x \in \{0,1\}^{\mathsf{F.il}(\lambda)}$ to return an output $\mathsf{F.Ev}(fk, x) \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$, where $\mathsf{F.il}, \mathsf{F.ol}\colon \mathbb{N} \to \mathbb{N}$ are the input and output length functions associated to F, respectively. The pre-image size of F is the function $\textsc{PreImg}_{\mathsf{F}}$ defined for $\lambda \in \mathbb{N}$ by

$$\textsc{PreImg}_{\mathsf{F}}(\lambda) = \max_{fk, x^*} \left| \{\, x \in \{0,1\}^{\mathsf{F.il}(\lambda)} \;:\; \mathsf{F.Ev}(fk, x) = \mathsf{F.Ev}(fk, x^*) \,\} \right|$$

where the maximum is over all $x^* \in \{0,1\}^{\mathsf{F.il}(\lambda)}$ and all keys fk. We say that F is injective if $\textsc{PreImg}_{\mathsf{F}}(\lambda) = 1$ for all $\lambda \in \mathbb{N}$, meaning $\mathsf{F.Ev}(fk, x_1) \neq \mathsf{F.Ev}(fk, x_2)$ for all distinct $x_1, x_2 \in \{0,1\}^{\mathsf{F.il}(\lambda)}$, all fk and all $\lambda \in \mathbb{N}$. We say that F has polynomial pre-image size if there is a polynomial p such that $\textsc{PreImg}_{\mathsf{F}}(\cdot) \leq p(\cdot)$.

One-wayness and hardcore functions. Function family F is one-way if $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\cdot)$ is negligible for all PT adversaries $\mathcal{F}$, where $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\lambda) = \Pr[\mathrm{OW}_{\mathsf{F}}^{\mathcal{F}}(\lambda)]$ and game $\mathrm{OW}_{\mathsf{F}}^{\mathcal{F}}(\lambda)$ is defined in Fig. 1. Let H be a family of functions with $\mathsf{H.il} = \mathsf{F.il}$. We say that H is hardcore for F if $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\cdot)$ is negligible for all PT adversaries $\mathcal{H}$, where $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\lambda) = 2\Pr[\mathrm{HC}_{\mathsf{F},\mathsf{H}}^{\mathcal{H}}(\lambda)] - 1$ and game $\mathrm{HC}_{\mathsf{F},\mathsf{H}}^{\mathcal{H}}(\lambda)$ is defined in Fig. 1.

PUNCTURED PRFS. A punctured function family G specifies (beyond the usual algorithms) additional PT algorithms $\mathsf{G.PKg}, \mathsf{G.PEv}$. On inputs 1^λ, a key $gk \in [\mathsf{G.Kg}(1^\lambda)]$ and target input $x^* \in \{0,1\}^{\mathsf{G.il}(\lambda)}$, algorithm $\mathsf{G.PKg}$ returns a "punctured" key gk^* such that $\mathsf{G.PEv}(gk^*, x) = \mathsf{G.Ev}(gk, x)$ for all $x \in \{0,1\}^{\mathsf{G.il}(\lambda)} \setminus \{x^*\}$. We say that G is a punctured PRF if $\mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\cdot)$ is negligible for all PT adversaries $\mathcal{G}$, where $\mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\lambda) = 2\Pr[\mathrm{PPRF}^{\mathcal{G}}_{\mathsf{G}}(\lambda)] - 1$ and game $\mathrm{PPRF}^{\mathcal{G}}_{\mathsf{G}}(\lambda)$ is defined in Fig. 1. Here $\mathcal{G}$ must make exactly one oracle query where it picks a target point x^* and gets back the corresponding punctured key together with a challenge for the value of $\mathsf{G.Ev}$ on the target point.

The concept of punctured PRFs is due to [11, 25, 13] who note that they can be built via the GGM construction [18]. This however yields a family G with $\mathsf{G.il} = \mathsf{G.ol}$. For our purposes, we need a stronger result, namely a punctured PRF with arbitrary polynomial output length:

Proposition 2.1 *Let ι, ℓ be polynomials and assume one-way functions exist. Then there is a punctured PRF G with $\mathsf{G.il} = \iota$ and $\mathsf{G.ol} = \ell$.*

The claimed punctured PRF G can be obtained by starting from a GGM-based punctured PRF $\overline{\mathsf{G}}$ with $\overline{\mathsf{G}}.\mathsf{il} = \overline{\mathsf{G}}.\mathsf{ol} = \iota$ and letting $\mathsf{G.Ev}(gk, x) = \mathsf{S.Ev}(\overline{\mathsf{G}}.\mathsf{Ev}(gk, x))$ where S is a PRG with input length ι and output length ℓ. We omit the details.

OBFUSCATION. A circuit sampler is a PT algorithm $\mathcal{S}$ that on input 1^λ returns a triple $(\mathrm{C}_0, \mathrm{C}_1, aux)$ where $\mathrm{C}_0, \mathrm{C}_1$ are circuits which have the same size, number of inputs and number of outputs, and aux is a string. An obfuscator is a PT algorithm Obf that on input 1^λ and a circuit C returns a circuit $\overline{\mathrm{C}}$ such that $\overline{\mathrm{C}}(x) = \mathrm{C}(x)$ for all x. We say that Obf is $\boldsymbol{S}$-secure, where $\boldsymbol{S}$ is a class (set) of circuit samplers, if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S},\mathcal{O}}(\cdot)$ is negligible for every PT adversary $\mathcal{O}$ and every circuit sampler $\mathcal{S} \in \boldsymbol{S}$, where $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathcal{S}}(\lambda)] - 1$ and game $\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathcal{S}}(\lambda)$ is defined in Fig. 1. Different types of iO security can now be captured by considering different classes of circuit samplers, as follows.

We say that a circuit sampler $\mathcal{S}$ is difference secure if $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S},\mathcal{D}}(\cdot)$ is negligible for every PT adversary $\mathcal{D}$, where $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S},\mathcal{D}}(\lambda) = 2\Pr[\mathrm{DIFF}^{\mathcal{D}}_{\mathcal{S}}(\lambda)] - 1$ and game $\mathrm{DIFF}^{\mathcal{D}}_{\mathcal{S}}(\lambda)$ is defined in Fig. 1. Difference security of $\mathcal{S}$ means that given $\mathrm{C}_0, \mathrm{C}_1, aux$ it is hard to find an input on which the circuits differ. Let $\boldsymbol{S}_{\mathrm{diff}}$ be the class of all difference-secure circuit samplers. Then Obf being $\boldsymbol{S}_{\mathrm{diff}}$-secure means it is a differing-inputs obfuscator as per [4].

We say that circuits $\mathrm{C}_0, \mathrm{C}_1$ are equivalent, written $\mathrm{C}_0 \equiv \mathrm{C}_1$, if they agree on all inputs. We say that circuit sampler $\mathcal{S}$ produces equivalent circuits if $\mathrm{C}_0 \equiv \mathrm{C}_1$ for all $\lambda \in \mathbb{N}$ and all $(\mathrm{C}_0, \mathrm{C}_1, aux) \in [\mathcal{S}(1^\lambda)]$. Let $\boldsymbol{S}_{\mathrm{eq}}$ be the class of all circuit samplers that produce equivalent circuits. Then Obf being $\boldsymbol{S}_{\mathrm{eq}}$-secure means it is an indistinguishability obfuscator as per [5, 30]. Note that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}$ (if $\mathcal{S}$ produces equivalent circuits it is certainly difference-secure) and hence any $\boldsymbol{S}_{\mathrm{diff}}$-secure obfuscator is a $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator. That is, diO implies iO, a fact we will often use.

We say that circuit sampler $\mathcal{S}$ produces d-differing circuits, where $d\colon \mathbb{N} \to \mathbb{N}$, if C_0 and C_1 differ on at most $d(\lambda)$ inputs for all $\lambda \in \mathbb{N}$ and all $(\mathrm{C}_0, \mathrm{C}_1, aux) \in [\mathcal{S}(1^\lambda)]$. Let $\boldsymbol{S}^d_{\mathrm{diff}}$ be the class of all difference-secure circuit samplers that produce d-differing circuits, so that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}^d_{\mathrm{diff}} \subseteq \boldsymbol{S}_{\mathrm{diff}}$. The interest of this definition is a result of BCP [12] showing that if d is a polynomial then any $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator is also a $\boldsymbol{S}^d_{\mathrm{diff}}$-secure obfuscator. We will exploit this to reduce our assumptions from $\boldsymbol{S}_{\mathrm{diff}}$-secure obfuscation to $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscation in some cases.

3 Poly-many hardcore bits for injective OWFs

In this section we consider the natural construction of a hardcore function with arbitrary span, namely an obfuscated PRF. We show that this works assuming the one-way function is injective and

Games G_0–G_4

$fk \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$; $gk \leftarrow_{\$} \mathsf{G.Kg}(1^\lambda)$; $x^* \leftarrow_{\$} \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$; $gk^* \leftarrow_{\$} \mathsf{G.PKg}(1^\lambda, gk, x^*)$

$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \cdot)$ // G_0

$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C} \leftarrow \mathrm{C}^1_{gk^*,x^*,r^*}$ // G_1

$r^* \leftarrow_{\$} \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathrm{C}^1_{gk^*,x^*,r^*}$ // G_2

$r^* \leftarrow_{\$} \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathrm{C}^2_{fk,gk,y^*,r^*}$ // G_3

$r^* \leftarrow_{\$} \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \cdot)$ // G_4

$\overline{\mathrm{C}} \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathrm{C})$; $hk \leftarrow \overline{\mathrm{C}}$; $b' \leftarrow \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$; Return $(b' = 1)$

Circuit $\mathrm{C}^1_{gk^*,x^*,r^*}(x)$	Circuit $\mathrm{C}^2_{fk,gk,y^*,r^*}(x)$
If $x \neq x^*$ then return $\mathsf{G.PEv}(gk^*, x)$	If $\mathsf{F.Ev}(fk, x) \neq y^*$ then return $\mathsf{G.Ev}(gk, x)$
Else return r^*	Else return r^*

Figure 2: **Games for proof of Theorem 3.1.**

the obfuscation is diO-secure, yielding our first result, namely a hardcore function with arbitrary polynomial span for any injective one-way function.

Construction. Let F, G be function families with $\mathsf{G.il} = \mathsf{F.il}$. Let Obf be an obfuscator. We define function family $\mathsf{H} = \mathbf{HC1}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}]$ as follows, with $\mathsf{H.il} = \mathsf{G.il}$ and $\mathsf{H.ol} = \mathsf{G.ol}$:

$\mathsf{H.Kg}(1^\lambda)$	$\mathsf{H.Ev}(hk, x)$
$gk \leftarrow_{\$} \mathsf{G.Kg}(1^\lambda)$; $\mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \cdot)$; $\overline{\mathrm{C}} \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathrm{C})$	$\overline{\mathrm{C}} \leftarrow hk$; $r \leftarrow \overline{\mathrm{C}}(x)$
$hk \leftarrow \overline{\mathrm{C}}$; Return hk	Return r

Here $\mathsf{G.Ev}(gk, \cdot)$ represents the circuit that given x returns $\mathsf{G.Ev}(gk, x)$. The description hk of the hardcore function is an obfuscation of this circuit. The hardcore function output is the result of this obfuscated circuit on x, which is simply $\mathsf{G.Ev}(gk, x)$, the result of the original circuit on x. The output of the hardcore function is thus the output of a PRF, the key for the latter embedded in an obfuscated circuit to prevent its being revealed.

Results. Recall that a $\boldsymbol{S}^1_{\mathrm{diff}}$-secure obfuscator is weaker than a full $\boldsymbol{S}_{\mathrm{diff}}$-secure obfuscator since it is only required to work on circuits that differ on at most one (hard to find) input. We have:

Theorem 3.1 *Let* F *be an injective one-way function family. Let* G *be a punctured PRF with* $\mathsf{G.il} = \mathsf{F.il}$. *Let* Obf *be a* $\boldsymbol{S}^1_{\mathrm{diff}}$*-secure obfuscator. Then the function family* $\mathsf{H} = \mathbf{HC1}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}]$ *defined above is hardcore for* F.

Proposition 2.1 yields punctured PRFs with as long an output as desired, so that the above allows extraction of an arbitrary polynomial number of hardcore bits. The one-way function assumption made in Proposition 2.1 is already implied by the assumption that F is one way. Theorem 3.1 assumes a differing-inputs obfuscator only for circuits that differ on at most one input, which by BCP13 [12] can be obtained from an indistinguishability obfuscator, making the latter the only assumption beyond one-wayness of F needed to extract polynomially-many hardcore bits. Formally, we have:

Corollary 3.2 *Let* ℓ *be any polynomial. Let* F *be any injective one-way function. If there exists a* $\boldsymbol{S}_{\mathrm{eq}}$*-secure obfuscator then there exists a hardcore function* H *for* F *with* $\mathsf{H.ol} = \ell$.

Proof of Theorem 3.1: Let $\mathcal{H}$ be a PT adversary. Consider the games and associated circuits of Fig. 2. Lines not annotated with comments are common to all five games. The games begin by picking keys fk, gk for the one-way function F and the punctured-PRF G, respectively. They pick the challenge input x^* and then create a punctured PRF key gk^* for it. Then the games differ in how they define r^* and the circuit C to be obuscated. These defined, the games re-unite to obfuscate the circuit and run $\mathcal{H}$.

Game G_0 does not use the punctured keys, and is equivalent to the $b = 1$ case of $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F,H}}(\lambda)$ while G_4, similarly, is its $b = 0$ case, so

$$\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F,H},\mathcal{H}}(\lambda) = \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_4] \ . \tag{1}$$

We now show that $\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i]$ is negligible for $i = 1, 2, 3, 4$, which by Equation (1) implies that $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F,H},\mathcal{H}}(\cdot)$ is negligible and proves the theorem. We begin with some intuition. In game G_1, we switch the circuit being obfuscated to one that uses the punctured key when $x \neq x^*$ and returns an embedded $r^* = \mathsf{G.Ev}(gk, x^*)$ otherwise. This circuit is equivalent to $\mathsf{G.Ev}(gk, \cdot)$ so iO-security, implied by diO, will tell us that the adversary $\mathcal{H}$ will hardly notice. This switch puts us in position to use the security of the punctured-PRF, based on which G_2 replaces r^* with a random value. These steps are direct applications of the Sahai-Waters technique [30], but now things get more difficult. Having made r^* random is not enough because we must now revert the circuit being obfuscated back to $\mathsf{G.Ev}(gk, \cdot)$. We also realize that we have not yet used the one-wayness of F, so this reversion must rely on it. But the circuit in G_2 embeds x^*, the point a one-wayness adversary would be trying to find, and it is not clear how we can simulate the construction of this circuit, required to run $\mathcal{H}$, in the design of an adversary violating one-wayness. To address this, instead of testing whether x equals x^*, the circuit in G_3 tests whether the values of $\mathsf{F.Ev}(fk, \cdot)$ on these points agree, which can be done given only $y^* = \mathsf{F.Ev}(fk, x^*)$ and avoids putting x^* in the circuit. But we need this to not alter the functionality of the circuit. This is where we make crucial use of the assumption that $\mathsf{F.Ev}(gk, \cdot)$ is injective. Finally, in game G_4 we revert the circuit back to $\mathsf{G.Ev}(gk, \cdot)$. But the circuits in $\mathrm{G}_3, \mathrm{G}_4$ are now manifestedly *not* equivalent. However, the input on which they differ is x^*. We show that the one-wayness of F implies that this point is hard to find, whence diO (here iO is not enough) allows us to conclude. We now proceed to the details.

Below, on the left we (simultaneously) define three circuit samplers that differ at the commented lines and have the uncommented lines in common. On the right, we define an iO-adversary:

Circuit Samplers $\mathcal{S}_1(1^\lambda)$, $\mathcal{S}_3(1^\lambda)$, $\mathcal{S}_4(1^\lambda)$	Adversary $\mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$
$fk \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$; $gk \leftarrow_{\$} \mathsf{G.Kg}(1^\lambda)$	$(fk, y^*, r^*) \leftarrow aux$
$x^* \leftarrow_{\$} \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$; $gk^* \leftarrow_{\$} \mathsf{G.PKg}(1^\lambda, gk, x^*)$	$hk \leftarrow \overline{\mathrm{C}}$
$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C}_1 \leftarrow \mathsf{G.Ev}(gk, \cdot)$; $\mathrm{C}_0 \leftarrow \mathrm{C}^1_{gk^*,x^*,r^*}$ // $\mathcal{S}_1$	$b' \leftarrow \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$
$r^* \leftarrow \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathrm{C}^1_{gk^*,x^*,r^*}$; $\mathrm{C}_0 \leftarrow \mathrm{C}^2_{fk,gk,y^*,r^*}$ // $\mathcal{S}_3$	Return b'
$r^* \leftarrow \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathrm{C}^2_{fk,gk,y^*,r^*}$; $\mathrm{C}_0 \leftarrow \mathsf{G.Ev}(gk, \cdot)$ // $\mathcal{S}_4$	
$aux \leftarrow (fk, y^*, r^*)$; Return $(\mathrm{C}_0, \mathrm{C}_1, aux)$	

Above, it is understood that $\mathrm{C}_0, \mathrm{C}_1$ are suitably padded to have the same size. Now we have

$$\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i] = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S}_i,\mathcal{O}}(\lambda) \qquad \text{for } i = 1, 3, 4 \ . \tag{2}$$

We now make three claims: (1) $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$ (2) $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{eq}}$ (3) $\mathcal{S}_4 \in \boldsymbol{S}^1_{\mathrm{diff}}$. Since $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}^1_{\mathrm{diff}}$ and Obf is assumed $\boldsymbol{S}^1_{\mathrm{diff}}$-secure, the RHS of Equation (2) is negligible in all three cases.

We now establish claim (1). If $x \neq x^*$ then $\mathrm{C}^1_{gk^*,x^*,r^*}(x) = \mathsf{G.PEv}(gk^*, x) = \mathsf{G.Ev}(gk, x)$. If $x = x^*$ then $\mathrm{C}^1_{gk^*,x^*,r^*}(x) = r^*$, but $\mathcal{S}_1$ sets $r^* = \mathsf{G.Ev}(gk, x^*)$. This means that $\mathcal{S}_1$ produces equivalent circuits, and hence $\mathcal{S}_1 \in \boldsymbol{S}_{\text{eq}}$.

Next we establish claim (2). The assumed injectivity of F implies that circuits $\mathrm{C}^1_{gk^*,x^*,r^*}$ and $\mathrm{C}^2_{fk,gk,y^*,r^*}$ are equivalent when $y^* = \mathsf{F.Ev}(fk, x^*)$, and hence $\mathcal{S}_3 \in \boldsymbol{S}_{\text{eq}}$.

Now we establish claim (3). Given any PT difference adversary $\mathcal{D}$ for $\mathcal{S}_4$, we build one-wayness adversary $\mathcal{F}$ via

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{F}(1^\lambda, fk, y^*)}\\ gk \leftarrow\!\!\$\; \mathsf{G.Kg}(1^\lambda)\,;\; r^* \leftarrow \{0,1\}^{\mathsf{G.ol}(\lambda)}\,;\; \mathrm{C}_1 \leftarrow \mathrm{C}^2_{fk,gk,y^*,r^*}\,;\; \mathrm{C}_0 \leftarrow \mathsf{G.Ev}(gk,\cdot)\\ aux \leftarrow (fk, y^*, r^*)\,;\; x \leftarrow\!\!\$\; \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)\,;\; \text{Return } x\end{array}$$

If $\mathrm{C}_1(x) \neq \mathrm{C}_0(x)$ then it must be that $\mathsf{F.Ev}(fk, x) = y^*$. Thus $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S}_4,\mathcal{D}}(\cdot) \leq \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\cdot)$. The assumed one-wayness of F thus means that $\mathcal{S}_4$ is difference-secure. But we also observe that, due to the injectivity of F, circuits $\mathrm{C}_0, \mathrm{C}_1$ differ on only one input, namely x^*. So $\mathcal{S}_4 \in \boldsymbol{S}^1_{\text{diff}}$.

One transition remains, namely that from G_1 to G_2. Here we have

$$\Pr[\mathrm{G}_1] - \Pr[\mathrm{G}_2] = \mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\lambda) \tag{3}$$

where adversary $\mathcal{G}$ is defined via

$$\begin{array}{l}\underline{\text{Adversary } \mathcal{G}^{\mathrm{CH}}(1^\lambda)}\\ fk \leftarrow\!\!\$\; \mathsf{F.Kg}(1^\lambda)\,;\; x^* \leftarrow\!\!\$\; \{0,1\}^{\mathsf{F.il}(\lambda)}\,;\; y^* \leftarrow \mathsf{F.Ev}(fk, x^*)\,;\; (gk^*, r^*) \leftarrow\!\!\$\; \mathrm{CH}(x^*)\\ \mathrm{C} \leftarrow \mathrm{C}^1_{gk^*,x^*,r^*}\,;\; hk \leftarrow \mathsf{Obf}(1^\lambda, \mathrm{C})\,;\; b' \leftarrow \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)\,;\; \text{Return } b'\end{array}$$

The RHS of Equation (3) is negligible by the assumption that G is a punctured PRF. This concludes the proof. ▌

4 Poly-many hardcore bits for any OWF

The proof of Theorem 3.1 makes crucial use of the assumed injectivity of F. To remove this assumption, we modify the construction so that the obfuscated PRF is applied, not to x, but to the result of the one-way function on x.

Construction. Let F, G be function families with $\mathsf{G.il} = \mathsf{F.ol}$. Let Obf be an obfuscator. We define function family $\mathsf{H} = \mathbf{HC2}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}]$ as follows, with $\mathsf{H.il} = \mathsf{G.il}$ and $\mathsf{H.ol} = \mathsf{G.ol}$:

$$\begin{array}{l|l}\underline{\mathsf{H.Kg}(1^\lambda, fk)} & \underline{\mathsf{H.Ev}(hk, x)}\\ gk \leftarrow\!\!\$\; \mathsf{G.Kg}(1^\lambda)\,;\; \mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk,\cdot))\,;\; \overline{\mathrm{C}} \leftarrow\!\!\$\; \mathsf{Obf}(1^\lambda, \mathrm{C}) & \overline{\mathrm{C}} \leftarrow hk\,;\; r \leftarrow \overline{\mathrm{C}}(x)\\ hk \leftarrow \overline{\mathrm{C}}\,;\; \text{Return } hk & \text{Return } r\end{array}$$

On input x, the circuit C computes $y = \mathsf{F.Ev}(fk, x)$ and returns $\mathsf{G.Ev}(gk, y)$. The description hk of the hardcore function is an obfuscation of this circuit. The hardcore function output on input x is thus $\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, x))$.

Results. The following says that diO-security of the obfuscator suffices for the security of the above hardcore function, and that this may further be restricted to circuits that differ at a number of points determined by the pre-image size of the one-way function:

Theorem 4.1 *Let F be a one-way function family. Let G be a punctured PRF with $\mathsf{G.il} = \mathsf{F.ol}$. Let $d = \textsc{PreImg}_{\mathsf{F}}$ and let Obf be a $\boldsymbol{S}^d_{\text{diff}}$-secure obfuscator. Then the function family $\mathsf{H} = \mathbf{HC2}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}]$ defined above is hardcore for F.*

$$
\begin{array}{l}
\underline{\text{Games } \mathrm{G}_0\text{–}\mathrm{G}_3} \\
fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)\ ;\ gk \leftarrow\!\!\$\ \mathsf{G.Kg}(1^\lambda)\ ;\ x^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}\ ;\ y^* \leftarrow \mathsf{F.Ev}(fk, x^*)\ ;\ gk^* \leftarrow\!\!\$\ \mathsf{G.PKg}(1^\lambda, gk, y^*) \\
\quad r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ;\ \ \mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)) \quad /\!/\ \mathrm{G}_0 \\
\quad r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ;\ \ \mathrm{C} \leftarrow \mathrm{C}^1_{fk,gk^*,y^*,r^*} \quad /\!/\ \mathrm{G}_1 \\
\quad r^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{G.ol}(\lambda)}\ ;\ \ \mathrm{C} \leftarrow \mathrm{C}^1_{fk,gk^*,y^*,r^*} \quad /\!/\ \mathrm{G}_2 \\
\quad r^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{G.ol}(\lambda)}\ ;\ \ \mathrm{C} \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)) \quad /\!/\ \mathrm{G}_3 \\
\overline{\mathrm{C}} \leftarrow\!\!\$\ \mathsf{Obf}(1^\lambda, \mathrm{C})\ ;\ hk \leftarrow \overline{\mathrm{C}}\ ;\ b' \leftarrow \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)\ ;\ \text{Return } (b' = 1) \\
\hline
\underline{\text{Circuit } \mathrm{C}^1_{fk,gk^*,y^*,r^*}(x)} \\
y \leftarrow \mathsf{F.Ev}(fk, x) \\
\text{If } y \neq y^* \text{ then return } \mathsf{G.PEv}(gk^*, y) \text{ else return } r^*
\end{array}
$$

Figure 3: **Games for proof of Theorem 4.1.**

This means that in the most general case, a differing-inputs obfuscator will suffice:

Corollary 4.2 *Let ℓ be any polynomial. Let F be any one-way function. If there exists a $\boldsymbol{S}_{\mathrm{diff}}$-secure obfuscator then there exists a hardcore function H for F with $\mathsf{H.ol} = \ell$.*

When the pre-image size of F is polynomial, however, we can again exploit BCP [12] to obtain our conclusion assuming nothing beyond iO:

Corollary 4.3 *Let ℓ be any polynomial. Let F be any one-way function with polynomially-bounded pre-image size. If there exists a $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator then there exists a hardcore function H for F with $\mathsf{H.ol} = \ell$.*

Proof of Theorem 4.1: Let $\mathcal{H}$ be a PT adversary. Consider the games and associated circuits of Fig. 3. Game G_0 does not use the punctured keys, and is equivalent to the $b = 1$ case of $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H}}(\lambda)$ while G_3, similarly, is its $b = 0$ case, so

$$\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\lambda) = \Pr[\mathrm{G}_0] - \Pr[\mathrm{G}_3]\ . \tag{4}$$

We now show that $\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i]$ is negligible for $i = 1, 2, 3$, which by Equation (4) implies that $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\cdot)$ is negligible and proves the theorem.

Below, on the left we (simultaneously) define two circuit samplers. On the right we define an iO-adversary:

$$
\begin{array}{l|l}
\underline{\text{Circuit Samplers } \mathcal{S}_1(1^\lambda),\ \mathcal{S}_3(1^\lambda)} & \underline{\text{Adversary } \mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)} \\
fk \leftarrow\!\!\$\ \mathsf{F.Kg}(1^\lambda)\ ;\ gk \leftarrow\!\!\$\ \mathsf{G.Kg}(1^\lambda) & (fk, y^*, r^*) \leftarrow aux \\
x^* \leftarrow\!\!\$\ \{0,1\}^{\mathsf{F.il}(\lambda)}\ ;\ y^* \leftarrow \mathsf{F.Ev}(fk, x^*)\ ;\ gk^* \leftarrow\!\!\$\ \mathsf{G.PKg}(1^\lambda, gk, y^*) & hk \leftarrow \overline{\mathrm{C}} \\
r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ;\mathrm{C}_1 \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot))\ ;\mathrm{C}_0 \leftarrow \mathrm{C}^1_{fk,gk^*,y^*,r^*} \ /\!/\ \mathcal{S}_1 & b' \leftarrow \mathcal{H}(1^\lambda, fk, hk, y^*, r^*) \\
r^* \leftarrow \{0,1\}^{\mathsf{G.ol}(\lambda)}\ ;\ \ \mathrm{C}_0 \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot))\ ;\mathrm{C}_1 \leftarrow \mathrm{C}^1_{fk,gk^*,y^*,r^*} \ /\!/\ \mathcal{S}_3 & \text{Return } b' \\
aux \leftarrow (fk, y^*, r^*)\ ;\ \text{Return } (\mathrm{C}_0, \mathrm{C}_1, aux) &
\end{array}
$$

Above, it is understood that $\mathrm{C}_0, \mathrm{C}_1$ are suitably padded to have the same size. Now we have

$$\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i] = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S}_i,\mathcal{O}}(\lambda) \qquad \text{for } i = 1, 3\ . \tag{5}$$

We now make two claims: (1) $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$ (2) $\mathcal{S}_3 \in \boldsymbol{S}^d_{\mathrm{diff}}$. Since $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}^d_{\mathrm{diff}}$ and Obf is assumed $\boldsymbol{S}^d_{\mathrm{diff}}$-secure, the RHS of Equation (5) is negligible in both cases.

We now establish claim (1). If $\mathsf{F.Ev}(\mathit{fk}, x) \neq y^*$ then $\mathrm{C}^1_{\mathit{fk},gk^*,y^*,r^*}(x) = \mathsf{G.PEv}(gk^*, y) = \mathsf{G.Ev}(gk, y)$. If $\mathsf{F.Ev}(\mathit{fk}, x) = y^*$ then $\mathrm{C}^1_{\mathit{fk},gk^*,y^*,r^*}(x) = r^*$, but $\mathcal{S}_1$ sets $r^* = \mathsf{G.Ev}(gk, y^*)$. This means that $\mathcal{S}_1$ produces equivalent circuits, and hence $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$.

Next we establish claim (2). Given any PT difference adversary $\mathcal{D}$ for $\mathcal{S}_3$ we build one-wayness adversary $\mathcal{F}$ via

$$
\begin{array}{l}
\underline{\text{Adversary } \mathcal{F}(1^\lambda, \mathit{fk}, y^*)} \\
gk \leftarrow\!\!\$\; \mathsf{G.Kg}(1^\lambda)\,;\; r^* \leftarrow \{0,1\}^{\mathsf{G.ol}(\lambda)}\,;\; \mathrm{C}_1 \leftarrow \mathrm{C}^1_{\mathit{fk},gk^*,y^*,r^*}\,;\; \mathrm{C}_0 \leftarrow \mathsf{G.Ev}(gk, \mathsf{F.Ev}(\mathit{fk}, \cdot)) \\
\mathit{aux} \leftarrow (\mathit{fk}, y^*, r^*)\,;\; x \leftarrow\!\!\$\; \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, \mathit{aux})\,;\; \text{Return } x
\end{array}
$$

If $\mathrm{C}_1(x) \neq \mathrm{C}_0(x)$ then it must be that $\mathsf{F.Ev}(\mathit{fk}, x) = y^*$. Thus $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S}_3,\mathcal{D}}(\cdot) \leq \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\cdot)$. The assumed one-wayness of F thus means that $\mathcal{S}_3$ is difference-secure. But we also observe that circuits $\mathrm{C}_0, \mathrm{C}_1$ differ exactly on pre-images of y^* under $\mathsf{F.Ev}(\mathit{fk}, \cdot)$, so $\mathcal{S}_4 \in \boldsymbol{S}^d_{\mathrm{diff}}$.

One transition remains, namely that from G_1 to G_2. Here we have

$$
\Pr[\mathrm{G}_1] - \Pr[\mathrm{G}_2] = \mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\lambda) \tag{6}
$$

where adversary $\mathcal{G}$ is defined via

$$
\begin{array}{l}
\underline{\text{Adversary } \mathcal{G}^{\mathrm{CH}}(1^\lambda)} \\
\mathit{fk} \leftarrow\!\!\$\; \mathsf{F.Kg}(1^\lambda)\,;\; x^* \leftarrow\!\!\$\; \{0,1\}^{\mathsf{F.il}(\lambda)}\,;\; y^* \leftarrow \mathsf{F.Ev}(\mathit{fk}, x^*)\,;\; (gk^*, r^*) \leftarrow\!\!\$\; \mathrm{CH}(y^*) \\
\mathrm{C} \leftarrow \mathrm{C}^1_{\mathit{fk},gk^*,y^*,r^*}\,;\; hk \leftarrow \mathsf{Obf}(1^\lambda, \mathrm{C})\,;\; b' \leftarrow \mathcal{H}(1^\lambda, \mathit{fk}, hk, y^*, r^*)\,;\; \text{Return } b'
\end{array}
$$

The RHS of Equation (6) is negligible by the assumption that G is a punctured PRF. This concludes the proof. ▮

References

[1] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *44th FOCS*, pages 146–159, Cambridge, Massachusetts, USA, Oct. 11–14, 2003. IEEE Computer Society Press. 3

[2] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In O. Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 474–495. Springer, Berlin, Germany, Mar. 15–17, 2009. 3

[3] W. Alexi, B. Chor, O. Goldreich, and C. P. Schnorr. RSA and Rabin functions: Certain parts are as hard as the whole. *SIAM Journal on Computing*, 17(2):194–209, 1988. 3

[4] P. Ananth, D. Boneh, S. Garg, A. Sahai, and M. Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013. 3, 4, 5, 7

[5] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18, Santa Barbara, CA, USA, Aug. 19–23, 2001. Springer, Berlin, Germany. 3, 4, 7

[6] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version in CRYPTO 2013. 3, 5

[7] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73, Fairfax, Virginia, USA, Nov. 3–5, 1993. ACM Press. 5

[8] M. Bellare and P. Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In S. Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 409–426, St. Petersburg, Russia, May 28 – June 1, 2006. Springer, Berlin, Germany. 6

[9] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on computing*, 15(2):364–383, 1986. 3

[10] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM journal on Computing*, 13(4):850–864, 1984. 3

[11] D. Boneh and B. Waters. Constrained pseudorandom functions and their applications. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300, Bengalore, India, Dec. 1–5, 2013. Springer, Berlin, Germany. 4, 7

[12] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. Cryptology ePrint Archive, Report 2013/650, 2013. 4, 5, 7, 8, 11

[13] E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. Cryptology ePrint Archive, Report 2013/401, 2013. 4, 7

[14] D. Catalano, R. Gennaro, and N. Howgrave-Graham. The bit security of Paillier's encryption scheme and its applications. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 229–243, Innsbruck, Austria, May 6–10, 2001. Springer, Berlin, Germany. 3

[15] Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 361–381, Zurich, Switzerland, Feb. 9–11, 2010. Springer, Berlin, Germany. 3

[16] R. Fischlin and C.-P. Schnorr. Stronger security proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244, 2000. 3

[17] D. M. Freeman, O. Goldreich, E. Kiltz, A. Rosen, and G. Segev. More constructions of lossy and correlation-secure trapdoor functions. *Journal of Cryptology*, 26(1):39–74, Jan. 2013. 3

[18] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986. 3, 7

[19] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *21st ACM STOC*, pages 25–32, Seattle, Washington, USA, May 15–17, 1989. ACM Press. 3, 5

[20] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984. 3

[21] J. Håstad and M. Näslund. The security of individual RSA bits. In *39th FOCS*, pages 510–521, Palo Alto, California, USA, Nov. 8–11, 1998. IEEE Computer Society Press. 3

[22] J. Håstad, A. Schrift, and A. Shamir. The discrete logarithm modulo a composite hides $O(n)$ bits. *Journal of Computer and System Sciences*, 47(3):376–404, 1993. 3

[23] S. Hohenberger, A. Sahai, and B. Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/509, 2013. 5

[24] B. S. Kaliski Jr. A pseudo-random bit generator based on elliptic logarithms. In A. M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 84–103, Santa Barbara, CA, USA, Aug. 1986. Springer, Berlin, Germany. 3

[25] A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias. Delegatable pseudorandom functions and applications. In A.-R. Sadeghi, V. D. Gligor, and M. Yung, editors, *ACM CCS 13*, pages 669–684, Berlin, Germany, Nov. 4–8, 2013. ACM Press. 4, 7

[26] D. L. Long and A. Wigderson. The discrete logarithm hides $O(\log n)$ bits. *SIAM journal on Computing*, 17(2):363–372, 1988. 3

[27] M. Näslund. All bits of $ax + b \bmod p$ are hard. In N. Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 114–128, Santa Barbara, CA, USA, Aug. 18–22, 1996. Springer, Berlin, Germany. 3

[28] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer, Berlin, Germany. 3

[29] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 187–196, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press. 3

[30] A. Sahai and B. Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Report 2013/454, 2013. 4, 7, 9

[31] C.-P. Schnorr and W. Alexi. RSA-bits are 0.5 + epsilon secure. In T. Beth, N. Cot, and I. Ingemarsson, editors, *EUROCRYPT'84*, volume 209 of *LNCS*, pages 113–126, Paris, France, Apr. 9–11, 1984. Springer, Berlin, Germany. 3

[32] U. V. Vazirani and V. V. Vazirani. Efficient and secure pseudo-random number generation. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 193–202, Santa Barbara, CA, USA, Aug. 19–23, 1984. Springer, Berlin, Germany. 3

[33] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91, Chicago, Illinois, Nov. 3–5, 1982. IEEE Computer Society Press. 3