Poly-Many Hardcore Bits for Any One-Way Function

Mihir Bellare[1] Igors Stepanovs[2] Stefano Tessaro[3]

February 2014

Abstract

We show how to extract an arbitrary polynomial number of simultaneously hardcore bits from any one-way function. In the case the one-way function is injective or has polynomially-bounded pre-image size, we assume the existence of indistinguishability obfuscation (iO). In the general case, we assume the existence of differing-input obfuscation (diO), but of a form weaker than full auxiliary-input diO. Our construction for injective one-way functions extends to extract hardcore bits on multiple, correlated inputs, yielding new D-PKE schemes.

[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 & Engineering, University of California San Diego, 9500 Gilman Drive, La Jolla, California 92093, USA. Email: `istepano@eng.ucsd.edu`.

[3] 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

OWF	Span	Assumption	Construction	See
injective	poly	iO	**HC1**	Corollary 4.2 of Theorem 4.1
poly pre-image size	poly	iO	**HC2**	Corollary 5.3 of Theorem 5.1
any	poly	diO^-	**HC2**	Corollary 5.2 of Theorem 5.1

Figure 1: **Our results:** We indicate the assumptions we make in order to construct hardcore functions with polynomial span. By diO^- we mean a weakening of diO that we define.

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 [16, 46, 30], hardcore functions have played a central role in the theory of cryptography. Historically, their motivating application was to achieve semantically-secure public-key encryption [30]. Here the public-key of the encryption scheme is a trapdoor, injective one-way function 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 both *injective* and *arbitrary* one-way functions, resolving a problem that had remained open, despite significant effort, since the 1980s. The tools we use are indistinguishability obfuscation (iO) [5, 42] and differing-input obfuscation (diO) [5, 19, 4]. See Fig. 1 for a summary of our results.

Prior work. Early results gave hardcore predicates (ie. span one) for specific one-way functions including discrete exponentiation modulo a prime, RSA and Rabin [16, 46, 43, 14, 35, 3, 39, 32, 24]. Eventually, in an impactful and influential work, Goldreich and Levin [29] gave a hardcore predicate for any one-way function. Extensions of these results are able to achieve logarithmic span [44, 38, 29, 1, 23].

Hardcore functions with polynomial span[2] have been provided for specific, algebraic functions including by Håstad, Schrift and Shamir [33] for discrete exponentiation modulo a composite, by Catalano, Gennaro and Howgrave-Graham [22] for the Paillier function [40], and by Akavia, Goldwasser and Vaikuntanathan [2] for certain LWE-based functions. Peikert and Waters showed that lossy trapdoor functions [41] achieve polynomial span, yielding further examples of specific one-way functions with polynomial span [41, 25].

In summary, this prior work has focused mostly on injective one-way functions, achieving polynomial span for particular functions but only logarithmic span for general injective one-way functions. For non-injective one-way functions also logarithmic span was the best achieved. The basic question that remained open, for both injective and non-injective one-way functions, was whether polynomial span is achievable. One answer was provided by [9], 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 arguably close to the desired conclusion.

Results. Injective one-way functions are the most important case for aplications. (Most applications are related to some form of encryption.) Accordingly we begin by focusing on the case where f is injective. Our construction (of a hardcore function h for f) is a natural one, namely to let h be an obfuscation of the circuit $\mathsf{G}(gk, \cdot)$ where G is a PRF [28] and gk is a random key for G. (The obfuscation is necessary because the

[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.

description of h must be public and revealing gk would violate security.) The difficulty is to show that this works assuming an achievable form of obfuscation. We prove that it does (as explained in more depth below) assuming the PRF is punctured [18, 36, 20] and the obfuscation is differing-input. Importantly, however, we are able to work with a form of diO that is weaker than full diO and shown by Boyle, Chung and Pass (BCP) [19] to be implied by iO. This yields our first result, a construction of a hardcore function with polynomial span for any injective one-way function assuming, beyond one-wayness of the original function, only the existence of an iO-secure obfuscator. (The punctured PRF is not an extra assumption since we are already assuming a one-way function.) We call this construction **HC1**.

The proof of security of **HC1** relies on the injectivity of f in a crucial way. For the case of an arbitrary (not necessarily injective) 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 $\mathsf{G}(gk, \cdot)$. We call this construction **HC2** and the proof is again expanded on below. This yields our second result, a construction of a hardcore function with polynomial span for any one-way function assuming, beyond one-wayness of the original function, only the existence of a diO-secure obfuscator. In the case the one-way function has polynomially-bounded pre-image size, BCP [19] can again be invoked to reduce the assumption to iO. While we do not know how to do this in the general case, we can reduce the assumption to a form of diO that is weaker than full auxiliary-input diO as defined in [19, 4].

In summary, in the important case of injective one-way functions, and even for one-way functions with polynomial pre-image size, we provide a hardcore function with polynomial span assuming only iO. For one-way functions with super-polynomial pre-image size, we provide a hardcore function with polynomial span assuming a weak form of diO that we denote by diO$^-$ in Fig. 1.

TECHNICAL APPROACH. We now take a closer look at the proofs to highlight the technical approach and novelties. Recall that the guarantee of iO [5, 42] 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 (diO) [5, 19, 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 3 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 [42] 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 [19] 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. The number of inputs on which the circuits involved differ is the number of possible pre-images of y^*. BCP [19] implies that iO suffices when this number is polynomial, but in general it could be exponential. In this case, diO suffices, but in fact a weaker version of it, that we define, does as well.

Extensions and applications. We show that our **HC1** hardcore function construction is able to extract random, independent bits even on inputs that are arbitrarily correlated, which is not true of most prior constructions and, combined with the fact that we get polynomially-many output bits for each input, yields new applications. In more detail, an injective function f is said to be one-way on a distribution $\mathcal{I}$ over vectors $\mathbf{x}$ if, given the result $f(\mathbf{x})$ of applying f to $\mathbf{x}$ component-wise, it is hard to recover any component of $\mathbf{x}$. We want a hardcore function such that the components of $h(\mathbf{x})$ look not only random but *independent* even given $f(\mathbf{x})$. If the components of $\mathbf{x}$ are independent, this is true for any hardcore function meeting the standard definition, and thus for ours, but if the components are correlated, standard hardcore functions give no such guarantee and may fail. As an example, many existing hardcore functions [16, 46, 43, 14, 35, 3, 39, 32, 24, 44, 38, 29, 1, 23, 33, 22, 40, 2] return certain specific bits of their input and will thus fail to be hardcore relative to the distribution in which $\mathbf{x}[2]$ is the bitwise complement of $\mathbf{x}[1]$, even if f is one-way on this distribution. We show however that **HC1** remains hardcore for f on *any* distribution $\mathcal{I}$ over which f is one-way, even when the entries of the vectors produced by $\mathcal{I}$ are arbitrarily correlated and even when $\mathcal{I}$ is not efficiently sampleable. This answers open questions from [31, 26].

Deterministic PKE (D-PKE) is useful for many applications, including efficient search on encrypted data [6] and providing resilience in the face of the low-quality randomness that pervades systems [7]. However, it cannot provide IND-CPA security. BBO [6] define what it means for a D-PKE scheme to provide PRIV-security over an input sampler $\mathcal{I}$, the latter returning vectors of *arbitrarily correlated* messages to be encrypted. We restrict attention to distributions that are admissible, meaning that there exists a family of injective, trapdoor functions that is one-way relative to $\mathcal{I}$. The basic question that emerges is, for which admissible distributions $\mathcal{I}$ does there exist a D-PKE scheme that is PRIV-secure over $\mathcal{I}$? We provide a full answer, showing that this is true for *all* admissible distributions, assuming only the existence of iO. We obtain this result using the security of **HC1** on correlated inputs and techniques from [26]. Previously, this was known only in the ROM [6], under the assumption that UCE-secure functions exist [10], for distributions with limited correlation between messages [8, 17] or assuming lossy trapdoor functions [26]. See Appendix B for definitions, a precise statement of the result, and proofs.

Parameterized diO. Of independent interest, we provide a definitional framework for diO in which security is parameterized by a class of circuit samplers. This allows us to unify and capture variant notions of iO and diO in the literature [5, 42, 19, 4]. The framework makes it easy to define further variants that are weaker than full diO, yielding a language in which one can state assumptions that are closer to those actually used in the proof rather than being overkill, particularly with regard to the type of auxiliary inputs used [13, 21, 27]. The weaker notion of diO noted above and denoted diO$^-$ in Fig. 1, is one such notion, requiring security only for “short” auxiliary inputs. See Section 3.

Discussion and related work. Random oracles (ROs) are “ideal” hardcore functions, able to provide polynomial span for any one-way function [11]. Our results, akin to [37, 9, 34], 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 PKE scheme [11] to obtain a standard-model IND-CPA scheme.

Interestingly, the hardcore function in our second construction is the reverse of the hash function used to instantiate FDH in HSW [34]: 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 [29, 23].

Given a one-way permutation f and a polynomial n it is possible to construct another one-way permutation g that has n hardcore bits. Namely let $g(x) = f^n(x)$ and let the hardcore function on x be the result of the Blum-Micali-Yao PRG [16, 46] on seed x. A similar transform is provided in [15].

GGHW [27] show that if a certain specific circuit can be obfuscated in a certain strong way, then full diO for all circuits is not possible. The assumption they need is however strong and novel and in any case the result does not rule out the assumptions we use, namely our weaker diO$^-$ assumption or iO. GGHW go on to show that if a certain specific Turing Machine can be obfuscated in a similar strong way then there is a one-way function with exponential pre-image size (in particular it is not injective) for which there is no hardcore predicate that is output-dependent. (That is, $f(x_1) = f(x_2)$ implies $h(x_1) = h(x_2)$, a property possessed by our second construction.) Again, however, the assumption they need is strong and novel, and it is not clear that it is more plausible than the existence of diO.

2 Preliminaries

We recall definitions for one-way functions, hardcore predicates and punctured PRFs.

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 C is a circuit then $|\mathrm{C}|$ denotes its size, and if s is an integer then $\mathsf{Pad}_s(\mathrm{C})$ denotes C padded to have size s. 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 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 result 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 [12]. (See Fig. 2 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

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

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

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

Figure 2: **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 $\mathcal{S}$.**

---

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 $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\lambda) = \Pr[\mathrm{OW}^{\mathcal{F}}_{\mathsf{F}}(\lambda)]$ and game $\mathrm{OW}^{\mathcal{F}}_{\mathsf{F}}(\lambda)$ is defined in Fig. 2. 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 $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\lambda) = 2\Pr[\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H}}(\lambda)] - 1$ and game $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H}}(\lambda)$ is defined in Fig. 2.

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. 2. 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 [18, 36, 20] who note that they can be built via the GGM construction [28]. 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.

3 Parameterized diO framework

We present a definitional framework for diO where security is parameterized by a class of circuit samplers. This is of conceptual value in enabling us to capture and unify existing forms of iO and diO. Further, it allows one to easily define new forms of diO that are *weaker* than the full auxiliary-input diO of [19, 4] and thus obtain results under weaker assumptions. Our parameterized language leads to sharper and more fine-grained security claims in which we can state assumptions that are closer to what is actually used by the proof rather than being overkill, in particular with regard to what types of auxiliary input are used [13, 21, 27]. Previous definitions did parameterize the definition by a class of circuits but this is different and in particular will not capture differences related to auxiliary inputs.

Circuit samplers. 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. 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. 2. 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.

iO-security. 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. If $\mathcal{S}$ is a circuit sampler and $\mathcal{O}$ is an adversary, we let $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S},\mathcal{O}}(\lambda) = 2\Pr[\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathcal{S}}(\lambda)] - 1$ where game $\mathrm{IO}^{\mathcal{O}}_{\mathsf{Obf},\mathcal{S}}(\lambda)$ is defined in Fig. 2. Now let $\boldsymbol{S}$ be a class (set) of circuit samplers. We say that Obf is $\boldsymbol{S}$-secure if $\mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S},\mathcal{O}}(\cdot)$ is negligible for every PT adversary $\mathcal{O}$ and every circuit sampler $\mathcal{S} \in \boldsymbol{S}$. This is our parameterized notion of security. The following obvious fact will often be useful:

Proposition 3.1 *Let $\boldsymbol{S}_1, \boldsymbol{S}_2$ be classes of circuit samplers and* Obf *an obfuscator. Suppose $\boldsymbol{S}_1 \subseteq \boldsymbol{S}_2$. Then if* Obf *is $\boldsymbol{S}_2$-secure it is also $\boldsymbol{S}_1$-secure.*

Capturing known notions. Different types of iO security in the literature can now be captured and unified by considering different classes of circuit samplers, as follows.

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 [19, 4].

Let $\boldsymbol{S}^{\overline{\mathrm{aux}}}$ be the class of circuit samplers $\mathcal{S}$ that do not have auxiliary inputs, meaning $aux = \varepsilon$ for all $\lambda \in \mathbb{N}$ and all $(\mathrm{C}_0, \mathrm{C}_1, aux) \in [\mathcal{S}(1^\lambda)]$. Let $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{diff}} = \boldsymbol{S}_{\mathrm{diff}} \cap \boldsymbol{S}^{\overline{\mathrm{aux}}} \subseteq \boldsymbol{S}_{\mathrm{diff}}$ be the class of all difference-secure circuit samplers that do not have auxiliary inputs. Then Obf being $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{diff}}$-secure means it is a differing-inputs obfuscator as per [5].

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 [42]. Let $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{eq}} = \boldsymbol{S}_{\mathrm{eq}} \cap \boldsymbol{S}^{\overline{\mathrm{aux}}}$ be the class of circuit samplers without auxiliary inputs that produce equivalent circuits. Then Obf being $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{eq}}$-secure means it is an indistinguishability obfuscator as per [5].

If $\mathcal{S}$ produces equivalent circuits it is certainly difference-secure. This means that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}$ and $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{eq}} \subseteq \boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{diff}}$. Hence Proposition 3.1 says that any $\boldsymbol{S}_{\mathrm{diff}}$-secure obfuscator is a $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator and any $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{diff}}$-secure obfuscator is a $\boldsymbol{S}^{\overline{\mathrm{aux}}}_{\mathrm{eq}}$-secure obfuscator. That is, diO implies iO, both for the case with auxiliary input and the case without, 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 $\boldsymbol{S}_{\mathrm{diff}}(d)$ be the class of all difference-secure circuit samplers that produce d-differing circuits, so that $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(d) \subseteq \boldsymbol{S}_{\mathrm{diff}}$. The interest of this definition is the following result of BCP [19]:

Proposition 3.2 *If d is a polynomial then any $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator is also a $\boldsymbol{S}_{\mathrm{diff}}(d)$-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.

New classes. Above, we have used our framework to express and capture existing variants of iO and diO. We now define a new variant, via a new class of samplers. Following the definition we will explain the motivation. We say that a circuit sampler $\mathcal{S}$ has short auxiliary inputs if $|aux| < |\mathrm{C}_b|$ for all $b \in \{0,1\}$, all $\lambda \in \mathbb{N}$ and all $(\mathrm{C}_0, \mathrm{C}_1, aux) \in [\mathcal{S}(1^\lambda)]$. We let $\boldsymbol{S}^{\mathrm{sh}}$ be the class of all circuit samplers that have short auxiliary inputs. The assumption made in Theorem 5.1 is a $\boldsymbol{S}$-secure obfuscator for a particular $\boldsymbol{S} \subseteq \boldsymbol{S}_{\mathrm{diff}} \cap \boldsymbol{S}^{\mathrm{sh}}$, meaning diO is only required relative to circuits samplers with short auxiliary inputs. This is a potentially *weaker* assumption than a $\boldsymbol{S}_{\rm diff}$-secure obfuscator.

4 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 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 G be a function family. Let Obf be an obfuscator and let $s\colon \mathbb{N} \to \mathbb{N}$. We define function family $\mathsf{H} = \mathbf{HC1}[\mathsf{G}, \mathsf{Obf}, s]$ as follows, with $\mathsf{H.il} = \mathsf{G.il}$ and $\mathsf{H.ol} = \mathsf{G.ol}$:

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

We give $\mathsf{H.Kg}$ two inputs because this is required by the syntax, but the second is ignored. Here $\mathsf{G.Ev}(gk, \cdot)$ represents the circuit that given x returns $\mathsf{G.Ev}(gk, x)$, and C is obtained by padding $\mathsf{G.Ev}(gk, \cdot)$ to size $s(\lambda)$. The padding length function s is a parameter of the construction that will depend on the one-way function F for which H will be hardcore. The description hk of the hardcore function is an obfuscation of circuit C. 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 $\boldsymbol{S}_{\rm diff}(1)$-secure obfuscator is weaker than a full $\boldsymbol{S}_{\rm diff}$-secure obfuscator since it is only required to work on circuits that differ on at most one (hard to find) input. We have:

Theorem 4.1 *Let* F *be an injective one-way function family. Let* G *be a punctured PRF with* $\mathsf{G.il} = \mathsf{F.il}$. *Then there is a polynomial* s *such that the following is true. Let* Obf *be any* $\boldsymbol{S}_{\rm diff}(1)$*-secure obfuscator. Then the function family* $\mathsf{H} = \mathbf{HC1}[\mathsf{G}, \mathsf{Obf}, s]$ *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 4.1 assumes a differing-inputs obfuscator only for circuits that differ on at most one input, which by Proposition 3.2 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 4.2 *Let* ℓ *be any polynomial. Let* F *be any injective one-way function. If there exists a* $\boldsymbol{S}_{\rm eq}$*-secure obfuscator then there exists a hardcore function* H *for* F *with* $\mathsf{H.ol} = \ell$.

Proof of Theorem 4.1: We define s as follows: For any $\lambda \in \mathbb{N}$ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathsf{G.Ev}(gk, \cdot)|, |\mathrm{C}^1_{gk^*,x^*,r^*}|, |\mathrm{C}^2_{fk,gk,y^*,r^*}|)$ where the last two circuits are in Fig. 3 and the maximum is over all $gk \in [\mathsf{G.Kg}(1^\lambda)]$, $x^* \in \{0,1\}^{\mathsf{G.il}(\lambda)}$, $gk^* \in [\mathsf{G.PKg}(1^\lambda, gk, x^*)]$, $fk \in [\mathsf{F.Kg}(1^\lambda)]$, $y^* \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$ and $r^* \in \{0,1\}^{\mathsf{G.ol}(\lambda)}$. Now let $\mathcal{H}$ be a PT adversary. Consider the games and associated circuits of Fig. 3. 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 $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H}}(\lambda)$ while G_4, similarly, is its $b = 0$ case, so

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

Games G_0–G_4

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

$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$ // G_0
$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,x^*,r^*})$ // G_1
$r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,x^*,r^*})$ // G_2
$r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,y^*,r^*})$ // G_3
$r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$ // 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 $\mathrm{C}^1_{gk^*,x^*,r^*}(x)$

If $x \neq x^*$ then return $\mathsf{G.PEv}(gk^*, x)$
Else return r^*

Circuit $\mathrm{C}^2_{fk,gk,y^*,r^*}(x)$

If $\mathsf{F.Ev}(fk, x) \neq y^*$ then return $\mathsf{G.Ev}(gk, x)$
Else return r^*

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

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 $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{H}}(\cdot)$ 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 [42], 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 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 $\mathcal{S}_1(1^\lambda)$, $\mathcal{S}_3(1^\lambda)$, $\mathcal{S}_4(1^\lambda)$

$fk \leftarrow\$ \mathsf{F.Kg}(1^\lambda)$; $gk \leftarrow\$ \mathsf{G.Kg}(1^\lambda)$
$x^* \leftarrow\$ \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$; $gk^* \leftarrow\$ \mathsf{G.PKg}(1^\lambda, gk, x^*)$
$r^* \leftarrow \mathsf{G.Ev}(gk, x^*)$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,x^*,r^*})$ // $\mathcal{S}_1$
$r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,x^*,r^*})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,y^*,r^*})$ // $\mathcal{S}_3$
$r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,y^*,r^*})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$ // $\mathcal{S}_4$
$aux \leftarrow (fk, y^*, r^*)$; Return $(\mathrm{C}_0, \mathrm{C}_1, aux)$

Adversary $\mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$

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

Now we have

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

We now make three claims: (1) $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$ (2) $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{eq}}$ (3) $\mathcal{S}_4 \in \boldsymbol{S}_{\mathrm{diff}}(1)$. Since $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(1)$ and Obf is assumed $\boldsymbol{S}_{\mathrm{diff}}(1)$-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 $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$.

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 $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{eq}}$.

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

Adversary $\mathcal{F}(1^\lambda, fk, y^*)$

$gk \leftarrow_{\$} \mathsf{G.Kg}(1^\lambda)$; $r^* \leftarrow_{\$} \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,y^*,r^*})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$
$aux \leftarrow (fk, y^*, r^*)$; $x \leftarrow_{\$} \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$; Return x

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 $\mathcal{S}_4 \in \boldsymbol{S}_{\mathrm{diff}}(1)$.

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

Adversary $\mathcal{G}^{\mathrm{CH}}(1^\lambda)$

$fk \leftarrow_{\$} \mathsf{F.Kg}(1^\lambda)$; $x^* \leftarrow_{\$} \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$; $(gk^*, r^*) \leftarrow_{\$} \mathrm{CH}(x^*)$
$\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,x^*,r^*})$; $hk \leftarrow_{\$} \mathsf{Obf}(1^\lambda, \mathrm{C})$; $b' \leftarrow_{\$} \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$; Return b'

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

5 Poly-many hardcore bits for any OWF

The proof of Theorem 4.1 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 and let $s\colon \mathbb{N} \to \mathbb{N}$. We define function family $\mathsf{H} = \mathbf{HC2}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}, s]$ as follows, with $\mathsf{H.il} = \mathsf{F.il}$ and $\mathsf{H.ol} = \mathsf{G.ol}$:

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

This time the result of the hardcore function output on input x is $\mathsf{G.Ev}(gk, y)$ where $y = \mathsf{F.Ev}(fk, x)$.

Results. The following says that $\boldsymbol{S}_{\mathrm{diff}}$-security of the obfuscator suffices for the security of the above hardcore function, but that in fact the assumption is weaker, the circuit samplers for which security is required being further restricted to have short auxiliary input and to produce circuits that differ at a number of points bounded by the pre-image size of the one-way function:

Theorem 5.1 *Let F be a one-way function family. Let G be a punctured PRF with $\mathsf{G.il} = \mathsf{F.ol}$. Let $d = \mathrm{PreImg}_{\mathsf{F}}$. Then there is a polynomial s such that the following is true. Let $\boldsymbol{S} = \boldsymbol{S}_{\mathrm{diff}}(d) \cap \boldsymbol{S}^{\mathrm{sh}}$. Let Obf be any $\boldsymbol{S}$-secure obfuscator. Then the function family $\mathsf{H} = \mathbf{HC2}[\mathsf{F}, \mathsf{G}, \mathsf{Obf}, s]$ defined above is hardcore for F.*

This means that in the most general case we have:

Corollary 5.2 *Let ℓ be any polynomial. Let F be any one-way function. If there exists a $(\boldsymbol{S}_{\mathrm{diff}} \cap \boldsymbol{S}^{\mathrm{sh}})$-secure obfuscator then there exists a hardcore function H for F with $\mathsf{H.ol} = \ell$.*

Games G_0–G_3

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

$r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ;\ \ \mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)))$ // G_0

$r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ;\ \ \mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{fk,gk^*,y^*,r^*})$ // G_1

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

$r^* \leftarrow_\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}\ ;\ \ \mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)))$ // G_3

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

Circuit $\mathrm{C}^1_{fk,gk^*,y^*,r^*}(x)$

$y \leftarrow \mathsf{F.Ev}(fk, x)$

If $y \neq y^*$ then return $\mathsf{G.PEv}(gk^*, y)$ else return r^*

Figure 4: **Games for proof of Theorem 5.1.**

When the pre-image size of F is polynomial, however, we can again exploit Proposition 3.2 to obtain our conclusion assuming nothing beyond iO:

Corollary 5.3 *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 5.1: We define s as follows: For any $\lambda \in \mathbb{N}$ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot))|, |\mathrm{C}^1_{fk,gk^*,y^*,r^*}|)$ where the second circuit is in Fig. 4 and the maximum is over all $gk \in [\mathsf{G.Kg}(1^\lambda)]$, $x^* \in \{0,1\}^{\mathsf{F.il}(\lambda)}$, $y^* \in \{0,1\}^{\mathsf{F.ol}(\lambda)}$, $gk^* \in [\mathsf{G.PKg}(1^\lambda, gk, y^*)]$, $fk \in [\mathsf{F.Kg}(1^\lambda)]$ and $r^* \in \{0,1\}^{\mathsf{G.ol}(\lambda)}$. Now let $\mathcal{H}$ be a PT adversary. Consider the games and associated circuits of Fig. 4. 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:

Circuit Samplers $\mathcal{S}_1(1^\lambda)$, $\mathcal{S}_3(1^\lambda)$

$fk \leftarrow_\$ \mathsf{F.Kg}(1^\lambda)\ ;\ gk \leftarrow_\$ \mathsf{G.Kg}(1^\lambda)$

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

$r^* \leftarrow \mathsf{G.Ev}(gk, y^*)\ ; \mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)))\ ; \mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{fk,gk^*,y^*,r^*})$ // $\mathcal{S}_1$

$r^* \leftarrow_\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}\ ;\ \mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk, \cdot)))\ ; \mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{fk,gk^*,y^*,r^*})$ // $\mathcal{S}_3$

$aux \leftarrow (fk, y^*, r^*)\ ;$ Return $(\mathrm{C}_0, \mathrm{C}_1, aux)$

Adversary $\mathcal{O}(1^\lambda, \overline{\mathrm{C}}, aux)$

$(fk, y^*, r^*) \leftarrow aux$

$hk \leftarrow \overline{\mathrm{C}}$

$b' \leftarrow_\$ \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$

Return b'

Now we have

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

We now make two claims: (1) $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}} \cap \boldsymbol{S}^{\mathrm{sh}}$ (2) $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{diff}}(d) \cap \boldsymbol{S}^{\mathrm{sh}}$. But $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(d)$. By Proposition 3.1 and the assumption that Obf is $(\boldsymbol{S}_{\mathrm{diff}}(d) \cap \boldsymbol{S}^{\mathrm{sh}})$-secure, the RHS of Equation (5) is negligible in both cases.

We now establish claim (1). If $\mathsf{F.Ev}(fk, x) \neq y^*$ then $\mathrm{C}^1_{fk,gk^*,y^*,r^*}(x) = \mathsf{G.PEv}(gk^*, y) = \mathsf{G.Ev}(gk, y)$ where $y = \mathsf{F.Ev}(fk, x)$. If $\mathsf{F.Ev}(fk, x) = y^*$ then $\mathrm{C}^1_{fk,gk^*,y^*,r^*}(x) = r^*$, but $\mathcal{S}_1$ sets $r^* = \mathsf{G.Ev}(gk, y^*)$. This means that $\mathcal{S}_1$ produces equivalent circuits, and hence $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$. Since fk, y^*, r^* are (explicitly, by construction) hardwired into $\mathrm{C}_0 = \mathrm{C}^1_{fk,gk^*,y^*,r^*}$, the size of this circuit is certainly larger than the length of $aux = (fk, y^*, r^*)$.

Since the two circuits output by the sampler are padded to the same size, the same is true for C_1. Thus $\mathcal{S}_1 \in \boldsymbol{S}^{\mathrm{sh}}$.

To establish claim (2), given any PT difference adversary $\mathcal{D}$ for $\mathcal{S}_3$ we build one-wayness adversary $\mathcal{F}$ that on input $1^\lambda, fk, y^*$ does the following:

$gk \leftarrow\$ \mathsf{G.Kg}(1^\lambda)$; $gk^* \leftarrow\$ \mathsf{G.PKg}(1^\lambda, gk, y^*)$; $r^* \leftarrow\$ \{0,1\}^{\mathsf{G.ol}(\lambda)}$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{fk,gk^*,y^*,r^*})$
$\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \mathsf{F.Ev}(fk,\cdot)))$; $aux \leftarrow (fk, y^*, r^*)$; $x \leftarrow\$ \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$; Return x

If $\mathrm{C}_1(x) \neq \mathrm{C}_0(x)$ then it must be that $\mathsf{F.Ev}(fk, x) = y^*$. Thus $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S}_3,\mathcal{D}}(\cdot) \leq \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{F}}(\cdot)$. 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 $\mathsf{F.Ev}(fk,\cdot)$, so $\mathcal{S}_4 \in \boldsymbol{S}_{\mathrm{diff}}(d)$. Also $\mathcal{S}_3 \in \boldsymbol{S}^{\mathrm{sh}}$ for the same reason as above.

One transition remains, namely that from G_1 to G_2. Here we have $\Pr[\mathrm{G}_1] - \Pr[\mathrm{G}_2] = \mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\lambda)$ where adversary $\mathcal{G}$, on input 1^λ and with access to oracle CH, returns b' computed via:

$fk \leftarrow\$ \mathsf{F.Kg}(1^\lambda)$; $x^* \leftarrow\$ \{0,1\}^{\mathsf{F.il}(\lambda)}$; $y^* \leftarrow \mathsf{F.Ev}(fk, x^*)$; $(gk^*, r^*) \leftarrow\$ \mathrm{CH}(y^*)$
$\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{fk,gk^*,y^*,r^*})$; $hk \leftarrow\$ \mathsf{Obf}(1^\lambda, \mathrm{C})$; $b' \leftarrow\$ \mathcal{H}(1^\lambda, fk, hk, y^*, r^*)$.

The assumption that G is a punctured PRF now concludes the proof. ▌

6 Hardcore functions for correlated inputs

We show that our hardcore functions are able to extract random bits even on sequences of inputs that are arbitrarily correlated. Somewhat more precisely, draw a vector $\mathbf{x}$ from an arbitrary distribution, in particuclar allowing its components to be arbitrarily correlated. Then applying our hardcore function componentwise to $\mathbf{x}$ results in a vector whose components look random *and independent* even given the result of applying f componentwise to $\mathbf{x}$, making only the necessary assumption that f remains one-way on the distribution from which $\mathbf{x}$ was selected. This is an unusual property, not possessed by all constructions of hardcore functions. The ability to extract polynomially-many bits on correlated inputs leads to new constructions of deterministic PKE schemes.

Notation. We denote vectors by boldface lowercase letters, for example $\mathbf{x}$. If $\mathbf{x}$ is a vector then $|\mathbf{x}|$ denotes the number of components of $\mathbf{x}$ and $\mathbf{x}[i]$ denotes the i-th component of $\mathbf{x}$, for any $1 \leq i \leq |\mathbf{x}|$. We write $x \in \mathbf{x}$ to mean that $x = \mathbf{x}[i]$ for some $1 \leq i \leq |\mathbf{x}|$. If F is a family of functions, $\mathbf{x}$ is a vector over $\{0,1\}^{\mathsf{F.il}(\lambda)}$ and $fk \in \{0,1\}^{\mathsf{F.kl}(\lambda)}$, then we let $\mathsf{F.Ev}(fk, \mathbf{x}) = (\mathsf{F.Ev}(fk, \mathbf{x}[1]), \ldots, \mathsf{F.Ev}(fk, \mathbf{x}[|\mathbf{x}|]))$. Let **Rand** denote the algorithm that on input a vector $\mathbf{x}$ and an integer ℓ returns a vector $\mathbf{r}$ of the same length as $\mathbf{x}$ whose entries are random ℓ-bit strings except that if two entries of $\mathbf{x}$ are the same, the same is true of the corresponding entries of $\mathbf{r}$. In detail, $\mathbf{Rand}(\mathbf{x}, \ell)$ creates table T via: For $i = 1, \ldots, |\mathbf{x}|$ do: If not $T[\mathbf{x}[i]]$ then $T[\mathbf{x}[i]] \leftarrow\$ \{0,1\}^\ell$. Then it sets $\mathbf{r}[i] \leftarrow T[\mathbf{x}[i]]$ for $i = 1, \ldots, |\mathbf{x}|$ and returns the vector $\mathbf{r}$.

Definitions. An input sampler is an algorithm $\mathcal{I}$ that on input 1^λ returns a $\mathcal{I}.\mathsf{vl}(\lambda)$-vector of strings over $\{0,1\}^{\mathcal{I}.\mathsf{il}(\lambda)}$, where the vector-length $\mathcal{I}.\mathsf{vl}\colon \mathbb{N} \to \mathbb{N}$ and the input length $\mathcal{I}.\mathsf{il}\colon \mathbb{N} \to \mathbb{N}$ are polynomials associated to $\mathcal{I}$. We say that a function family F is one-way with respect to input sampler $\mathcal{I}$ if $\mathsf{F.il} = \mathcal{I}.\mathsf{il}$ and $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{I},\mathcal{F}}(\cdot)$ is negligible for all PT adversaries $\mathcal{F}$, where $\mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{I},\mathcal{F}}(\lambda) = \Pr[\mathrm{OW}^{\mathcal{F}}_{\mathsf{F},\mathcal{I}}(\lambda) = 1]$ and game $\mathrm{OW}^{\mathcal{F}}_{\mathsf{F},\mathcal{I}}(\lambda)$ is defined in Fig. 5. We stress that for $\mathcal{F}$ to win in this game it needs to find the inverse under $\mathsf{F.Ev}(fk,\cdot)$ of *some* component of $\mathbf{y}^*$, not all components. Let H be a family of functions with $\mathsf{H.il} = \mathsf{F.il}$. We say that H is hardcore for F with respect to input sampler $\mathcal{I}$ if $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{I},\mathcal{H}}(\cdot)$ is negligible for all PT adversaries $\mathcal{H}$, where $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F},\mathsf{H},\mathcal{I},\mathcal{H}}(\lambda) = 2\Pr[\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H},\mathcal{I}}(\lambda)] - 1$ and game $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F},\mathsf{H},\mathcal{I}}(\lambda)$ is defined in Fig. 5.

We extend punctured PRFs as defined in Section 2 to allow puncturing at multiple points. In a punctured function family G, algorithm $\mathsf{G.PKg}$ now takes 1^λ, a key $gk \in [\mathsf{G.Kg}(1^\lambda)]$ and a vector $\mathbf{x}^*$ over $\{0,1\}^{\mathsf{G.il}(\lambda)}$ (the target inputs) to return a "punctured" key gk^* such that $\mathsf{G.PEv}(gk^*, x) = \mathsf{G.Ev}(gk, x)$ for all $x \in \{0,1\}^{\mathsf{G.il}(\lambda)}$ such that $x \notin \mathbf{x}^*$. 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. 5. Here $\mathcal{G}$ must make exactly one oracle query consisting of a vector over $\{0,1\}^{\mathsf{G.il}(\lambda)}$. Proposition 2.1 extends, and we exploit this below.

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

Figure 5: **Games defining one-wayness of F with respect to an input sampler $\mathcal{I}$, security of H as a hardcore function for F with respect to $\mathcal{I}$ and punctured-PRF security of G on multiple inputs.**

Results. The following says that our construction for injective functions F extractrs hardcore bits for any input sampler $\mathcal{I}$ with respect to which F is one way, meaning even for arbitrarily correlated inputs. The sampler need not even be PT. The proof is in Appendix A:

Theorem 6.1 *Let F be an injective function family. Let G be a punctured PRF with $\mathsf{G.il} = \mathsf{F.il}$. Let d: $\mathbb{N} \to \mathbb{N}$ be a polynomial. Then there is a polynomial s such that the following is true. Let $\mathcal{I}$ be any input sampler with $\mathcal{I}.\mathsf{vl} = d$ and $\mathcal{I}.\mathsf{il} = \mathsf{F.il}$. Let Obf be any $\boldsymbol{S}_{\mathrm{diff}}(d)$-secure obfuscator. Assume F is one-way with respect to $\mathcal{I}$. Then the function family $\mathsf{H} = \mathbf{HC1}[\mathsf{G}, \mathsf{Obf}, s]$ defined in Section 4 is hardcore for F with respect to $\mathcal{I}$.*

A subtle point here is that s depends on $d = \mathcal{I}.\mathsf{vl}$ in addition to F and G, which means that the size of the key hk describing the hardcore function grows with the number of correlated inputs d on which we want the function to be hardcore. This is expected and due to [45] may not be avoidable under falsifiable assumptions. Importantly for our applications, H does not depend on $\mathcal{I}$ beyond depending on $d = \mathcal{I}.\mathsf{vl}$ and $\mathsf{F.il} = \mathcal{I}.\mathsf{il}$.

Since d in Theorem 6.1 is a polynomial, we may apply Proposition 3.2 to obtain the analog of Corollary 4.2 for correlated inputs, namely that, assuming only a $\boldsymbol{S}_{\mathrm{eq}}$-secure obfuscator (i.e. iO), there exists, for any injective F and any input sampler $\mathcal{I}$, a function family that returns polynomially-many bits and is hardcore for F with respect to $\mathcal{I}$. In Appendix B we discuss the application of Theorem 6.1 to the design of new D-PKE schemes that are PRIV-secure for arbitrarily correlated messages.

Acknowledgments

We thank Daniel Wichs and Elette Boyle for comments and discussions.

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, 5

[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, 5

[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, 5

[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, 8

[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, 5, 8

[6] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In A. Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552, Santa Barbara, CA, USA, Aug. 19–23, 2007. Springer, Berlin, Germany. 5, 18, 19

[7] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249, Tokyo, Japan, Dec. 6–10, 2009. Springer, Berlin, Germany. 5

[8] M. Bellare, M. Fischlin, A. O'Neill, and T. Ristenpart. Deterministic encryption: Definitional equivalences and constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 360–378, Santa Barbara, CA, USA, Aug. 17–21, 2008. Springer, Berlin, Germany. 5, 19, 20

[9] 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, 6

[10] M. Bellare, V. T. Hoang, and S. Keelveedhi. Instantiating random oracles via UCEs. Cryptology ePrint Archive, Report 2013/424, 2013. Preliminary version appeared at *CRYPTO 2013*, pages 398–415, 2013. 5, 19

[11] 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. 6

[12] 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

[13] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. Cryptology ePrint Archive, Report 2013/641, 2013. 5, 8

[14] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal on computing*, 15(2):364–383, 1986. 3, 5

[15] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G. R. Blakley and D. Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 289–302, Santa Barbara, CA, USA, Aug. 19–23, 1984. Springer, Berlin, Germany. 6

[16] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM journal on Computing*, 13(4):850–864, 1984. 3, 5, 6

[17] A. Boldyreva, S. Fehr, and A. O'Neill. On notions of security for deterministic encryption, and efficient constructions without random oracles. In D. Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 335–359, Santa Barbara, CA, USA, Aug. 17–21, 2008. Springer, Berlin, Germany. 5, 19

[18] 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

[19] E. Boyle, K.-M. Chung, and R. Pass. On extractability obfuscation. Cryptology ePrint Archive, Report 2013/650, 2013. 3, 4, 5, 7, 8

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

[21] E. Boyle and R. Pass. Limits of extractability assumptions with distributional auxiliary input. Cryptology ePrint Archive, Report 2013/703, 2013. 5, 8

[22] 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, 5

[23] 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, 5, 6

[24] R. Fischlin and C.-P. Schnorr. Stronger security proofs for RSA and Rabin bits. *Journal of Cryptology*, 13(2):221–244, 2000. 3, 5

[25] 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

[26] B. Fuller, A. O'Neill, and L. Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 582–599, Taormina, Sicily, Italy, Mar. 19–21, 2012. Springer, Berlin, Germany. 5, 19, 20

[27] S. Garg, C. Gentry, S. Halevi, and D. Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. Cryptology ePrint Archive, Report 2013/860, 2013. 5, 6, 8

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

[29] 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, 6

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

[31] V. Goyal, A. O'Neill, and V. Rao. Correlated-input secure hash functions. In Y. Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 182–200, Providence, RI, USA, Mar. 28–30, 2011. Springer, Berlin, Germany. 5

[32] 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, 5

[33] 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, 5

[34] S. Hohenberger, A. Sahai, and B. Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. Cryptology ePrint Archive, Report 2013/509, 2013. 6

[35] 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, 5

[36] 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

[37] E. Kiltz, A. O'Neill, and A. Smith. Instantiability of RSA-OAEP under chosen-plaintext attack. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 295–313, Santa Barbara, CA, USA, Aug. 15–19, 2010. Springer, Berlin, Germany. 6

[38] D. L. Long and A. Wigderson. The discrete logarithm hides $O(\log n)$ bits. *SIAM journal on Computing*, 17(2):363–372, 1988. 3, 5

[39] 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, 5

[40] 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, 5

[41] 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

[42] A. Sahai and B. Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. Cryptology ePrint Archive, Report 2013/454, 2013. 3, 4, 5, 8, 10

[43] 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, 5

[44] 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, 5

Games G_0–G_4

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

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

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

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

$\mathbf{r}^* \leftarrow_{\$} \mathbf{Rand}(\mathbf{x}^*, \mathsf{G.ol}(\lambda))$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,\mathbf{y}^*,\mathbf{r}^*})$ // G_3

$\mathbf{r}^* \leftarrow_{\$} \mathbf{Rand}(\mathbf{x}^*, \mathsf{G.ol}(\lambda))$; $\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$ // G_4

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

Circuit $\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*}(x)$	Circuit $\mathrm{C}^2_{fk,gk,\mathbf{y}^*,\mathbf{r}^*}(x)$
If $\exists i : x = \mathbf{x}^*[i]$ then return $\mathbf{r}^*[i]$	If $\exists i$: $\mathsf{F.Ev}(fk, x) = \mathbf{y}^*[i]$ then return $\mathbf{r}^*[i]$
Else return $\mathsf{G.PEv}(gk^*, x)$	Else return $\mathsf{G.Ev}(gk, x)$

Figure 6: **Games for proof of Theorem 6.1.**

[45] D. Wichs. Barriers in cryptography with weak, correlated and leaky sources. In R. D. Kleinberg, editor, *ITCS 2013*, pages 111–126, Berkeley, CA, USA, Jan. 9–12, 2013. ACM. 14

[46] 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, 5, 6

A Proof of Theorem 6.1

Proof of Theorem 6.1: Theorem 4.1 is the special case of this theorem for the input sampler $\mathcal{I}$ with $\mathcal{I}.\mathsf{vl}(\lambda) = 1$ that outputs a uniformly random string in $\{0,1\}^{\mathsf{F.il}(\lambda)}$. The proof of Theorem 4.1 extends to this general case, but some care must be taken due to the fact that we are making no assumption on the sampler, and in particular some components of a sampled $\mathbf{x}^* \leftarrow_{\$} \mathcal{I}(1^\lambda)$ may collide. A complete description of the games and associated circuits for the proof are defined in Fig. 6.

In the following, recall that $d = \mathcal{I}.\mathsf{vl}$. We define s as follows: For any $\lambda \in \mathbb{N}$ let $s(\lambda)$ be a polynomial upper bound on $\max(|\mathsf{G.Ev}(gk, \cdot)|, |\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*}|, |\mathrm{C}^2_{fk,gk,\mathbf{y}^*,\mathbf{r}^*}|)$ where the last two circuits are in Fig. 6 and the maximum is over all $gk \in [\mathsf{G.Kg}(1^\lambda)]$, all d-vectors $\mathbf{x}^*$ over $\{0,1\}^{\mathsf{G.il}(\lambda)}$, $gk^* \in [\mathsf{G.PKg}(1^\lambda, gk, \mathbf{x}^*)]$, $fk \in [\mathsf{F.Kg}(1^\lambda)]$, as well as all d-vectors $\mathbf{y}^*$ and $\mathbf{r}^*$ over $\{0,1\}^{\mathsf{F.ol}(\lambda)}$ and $\{0,1\}^{\mathsf{G.ol}(\lambda)}$, respectively.

Now let $\mathcal{H}$ be a PT adversary. We consider the games and associated circuits of Fig. 6. (As usual, lines not annotated with comments are common to all five games.) The games are very similar to those in the proof of Theorem 4.1, with the exception that we now sample $\mathbf{x}^*$ using the input sampler $\mathcal{I}$ instead of sampling from $\{0,1\}^{\mathsf{F.il}(1^\lambda)}$, and then use the corresponding vector notation for defining circuits. Even more importantly, while the proof of Theorem 4.1 used a single-point punctured PRF and a $\boldsymbol{S}_{\mathrm{diff}}(1)$-secure obfuscator, now we require to use a punctured PRF with $\mathcal{I}.\mathsf{vl}(\lambda)$ simultaneously punctured target points (required for transition from G_1 to G_2) and a $\boldsymbol{S}_{\mathrm{diff}}(d)$-secure obfuscator for $d = \mathcal{I}.\mathsf{vl}$ (required for the circuit sampler $\mathcal{S}_4$ used below).

We now discuss the game transitions. Game G_0 does not use the punctured keys, and is equivalent to the $b = 1$ case of $\mathrm{HC}^{\mathcal{H}}_{\mathsf{F,H},\mathcal{I}}(\lambda)$ while G_4 corresponds to the $b = 0$ case, so

$$\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F,H},\mathcal{I},\mathcal{H}}(\lambda) = \Pr[G_0] - \Pr[G_4] . \tag{6}$$

We now show that $\Pr[G_{i-1}] - \Pr[G_i]$ is negligible for $i = 1, 2, 3, 4$, which by Equation (6) implies that $\mathsf{Adv}^{\mathsf{hc}}_{\mathsf{F,H},\mathcal{I},\mathcal{H}}(\cdot)$ is negligible and proves the theorem.

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

Now we can easily verify that

$$\Pr[\mathrm{G}_{i-1}] - \Pr[\mathrm{G}_i] = \mathsf{Adv}^{\mathsf{io}}_{\mathsf{Obf},\mathcal{S}_i,\mathcal{O}}(\lambda) \qquad \text{for } i \in \{1, 3, 4\} . \tag{7}$$

The only new subtle point that we have implicitly used above is that because F is injective, we obtain the same distribution for $\mathbf{r}^*$ if we sample it by inputting $\mathbf{y}^*$ to **Rand**, rather than $\mathbf{x}^*$. This issue did not appear in the single-input case considered in Theorem 4.1.

We now make three claims: (1) $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$ (2) $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{eq}}$ (3) $\mathcal{S}_4 \in \boldsymbol{S}_{\mathrm{diff}}(d)$. Since $\boldsymbol{S}_{\mathrm{eq}} \subseteq \boldsymbol{S}_{\mathrm{diff}}(d)$ and Obf is assumed $\boldsymbol{S}_{\mathrm{diff}}(d)$-secure, the RHS of Equation (7) is negligible in all three cases.

We now establish claim (1). If $x \notin \mathbf{x}^*$ then $\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*}(x) = \mathsf{G.PEv}(gk^*, x) = \mathsf{G.Ev}(gk, x)$. If $x = \mathbf{x}^*[i]$ then $\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*}(x) = \mathbf{r}^*[i]$, but $\mathcal{S}_1$ sets $\mathbf{r}^*[i] = \mathsf{G.Ev}(gk, \mathbf{x}^*[i])$. This means that $\mathcal{S}_1$ produces equivalent circuits, and hence $\mathcal{S}_1 \in \boldsymbol{S}_{\mathrm{eq}}$. Similarly, for claim (2), the assumed injectivity of F implies that circuits $\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*}$ and $\mathrm{C}^2_{fk,gk,\mathbf{y}^*,\mathbf{r}^*}$ are equivalent when $\mathbf{y}^* = \mathsf{F.Ev}(fk, \mathbf{x}^*)$, and hence $\mathcal{S}_3 \in \boldsymbol{S}_{\mathrm{eq}}$.

We now turn to verifying claim (3): Given any PT difference adversary $\mathcal{D}$ for $\mathcal{S}_4$, we build one-wayness adversary $\mathcal{F}$ via

Adversary $\mathcal{F}(1^\lambda, fk, \mathbf{y}^*)$

$gk \leftarrow_\$ \mathsf{G.Kg}(1^\lambda)$; $\mathbf{r}^* \leftarrow_\$ \mathbf{Rand}(\mathbf{y}^*, \mathsf{G.ol}(\lambda))$; $\mathrm{C}_1 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^2_{fk,gk,\mathbf{y}^*,\mathbf{r}^*})$; $\mathrm{C}_0 \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathsf{G.Ev}(gk, \cdot))$
$aux \leftarrow (fk, \mathbf{y}^*, \mathbf{r}^*)$; $x \leftarrow_\$ \mathcal{D}(\mathrm{C}_0, \mathrm{C}_1, aux)$; Return x

If $\mathrm{C}_1(x) \neq \mathrm{C}_0(x)$ then it must be that $\mathsf{F.Ev}(fk, x) \in \mathbf{y}^*$. Thus $\mathsf{Adv}^{\mathsf{diff}}_{\mathcal{S}_4,\mathcal{D}}(\cdot) \leq \mathsf{Adv}^{\mathsf{ow}}_{\mathsf{F},\mathcal{I},\mathcal{F}}(\cdot)$. The assumed one-wayness of F with respect to $\mathcal{I}$ 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 on d inputs, namely all $x' \in \mathbf{x}^*$. So $\mathcal{S}_4 \in \boldsymbol{S}_{\mathrm{diff}}(d)$.

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

$$\Pr[\mathrm{G}_1] - \Pr[\mathrm{G}_2] = \mathsf{Adv}^{\mathsf{pprf}}_{\mathsf{G},\mathcal{G}}(\lambda) \tag{8}$$

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

Adversary $\mathcal{G}^{\mathrm{CH}}(1^\lambda)$

$fk \leftarrow_\$ \mathsf{F.Kg}(1^\lambda)$; $\mathbf{x}^* \leftarrow_\$ \mathcal{I}(1^\lambda)$; $\mathbf{y}^* \leftarrow \mathsf{F.Ev}(fk, \mathbf{x}^*)$; $(gk^*, \mathbf{r}^*) \leftarrow_\$ \mathrm{CH}(\mathbf{x}^*)$
$\mathrm{C} \leftarrow \mathsf{Pad}_{s(\lambda)}(\mathrm{C}^1_{gk^*,\mathbf{x}^*,\mathbf{r}^*})$; $hk \leftarrow_\$ \mathsf{Obf}(1^\lambda, \mathrm{C})$; $b' \leftarrow_\$ \mathcal{H}(1^\lambda, fk, hk, \mathbf{y}^*, \mathbf{r}^*)$; Return b'

The RHS of Equation (8) is negligible by the assumption that G is a punctured PRF. This concludes the proof. ▮

B Application to D-PKE

A PKE scheme is deterministic if its encryption function is deterministic. Deterministic public-key encryption (D-PKE) is useful for many applications. However, it cannot provide IND-CPA security. BBO [6] define what it means for a D-PKE scheme to provide PRIV-security over an input sampler $\mathcal{I}$, the latter returning vectors of *arbitrarily correlated* messages to be encrypted. We restrict attention to distributions that are admissible, meaning that there exists a family of injective, trapdoor functions that is one-way relative to $\mathcal{I}$. The basic question that emerges is, for which admissible distributions $\mathcal{I}$ does there exist a D-PKE scheme that is PRIV-secure over $\mathcal{I}$? We provide a full answer, showing that this is true for *all* admissible distributions, assuming only the existence of iO. We obtain this result by combining Theorem 6.1 with techniques from [26]. Previously, this was known only in the ROM [6], under the assumption that UCE-secure functions exist [10], for distributions with limited correlation between messages [8, 17] or assuming lossy trapdoor functions [26].

Construction and result. We extend our definition of a family of functions from Section 2 to allow trapdoors by allowing $\mathsf{F.Kg}$ to return not just fk but also a trapdoor fk^{-1} that, via a PT algorithm $\mathsf{F.Ev}^{-1}$, defines a function $\mathsf{F.Ev}^{-1}(fk^{-1}, \cdot)$ that is the inverse of $\mathsf{F.Ev}(fk, \cdot)$. We refer to prior work [6, 8, 26] for definitions of D-PKE and the notion of a D-PKE scheme being PRIV-secure over an input sampler $\mathcal{I}$.

Generalizing prior constructions from [8, 6], Fuller, O'Neill and Reyzin [26] proposed a construction of a D-PKE scheme from an IND-CPA PKE scheme PKE and a family of injective, trapdoor one-way functions F. It uses a keyless hardcore function for F. We adapt it to allow a family of hardcore functions H for F that, as per our syntax and constructions, is keyed. The resulting D-PKE scheme $\mathsf{D\text{-}PKE} = \mathsf{EwHCore}[\mathsf{PKE}, \mathsf{F}, \mathsf{H}]$ associated to $\mathsf{PKE}, \mathsf{F}, \mathsf{H}$ is defined as follows:

$\underline{\mathsf{D\text{-}PKE.Kg}(1^\lambda)}$	$\underline{\mathsf{D\text{-}PKE.Enc}((pk, fk, hk), x)}$	$\underline{\mathsf{D\text{-}PKE.Dec}((sk, fk^{-1}), c)}$
$(pk, sk) \leftarrow\$ \mathsf{PKE.Kg}(1^\lambda)$	$y \leftarrow \mathsf{F.Ev}(fk, x)$	$y \leftarrow \mathsf{PKE.Dec}(sk, c)$
$(fk, fk^{-1}) \leftarrow\$ \mathsf{F.Kg}(1^\lambda)$	$r \leftarrow \mathsf{H.Ev}(hk, x)$	$x \leftarrow \mathsf{F.Ev}^{-1}(fk^{-1}, y)$
$hk \leftarrow\$ \mathsf{H.Kg}(1^\lambda, fk)$	$c \leftarrow \mathsf{PKE.Enc}(pk, y; r)$	Return x
Return $((pk, fk, hk), (sk, fk^{-1}))$	Return c	

We note that it is central for this construction that the hardcore function outputs polynomially many bits, since these bits are used as the coins for the PKE scheme. The following says that for any admissible $\mathcal{I}$ there is a D-PKE scheme that is PRIV-secure over $\mathcal{I}$:

Theorem B.1 *Let* PKE *be an IND-CPA secure PKE scheme. Let* F *be an injective trapdoor function family that is one-way with respect to input sampler* $\mathcal{I}$. *If there exists a* $\boldsymbol{S}_{\mathrm{eq}}$*-secure obfuscator then there exists a function family* H *such that* $\mathsf{D\text{-}PKE} = \mathsf{EwHCore}[\mathsf{PKE}, \mathsf{F}, \mathsf{H}]$ *is PRIV-secure on* $\mathcal{I}$.

Proof. The rest of this section is devoted to a proof of Theorem B.1. We will show that our constructions of hardcore functions meet the notion of a robust hardcore function from [26] and then apply results from the latter to obtain Theorem B.1.

We now define robustness. Let $\mathcal{I}, \mathcal{I}^*$ be input samplers such that $(\mathcal{I}.\mathsf{vl}, \mathcal{I}.\mathsf{il}) = (\mathcal{I}^*.\mathsf{vl}, \mathcal{I}^*.\mathsf{il})$ and also $[\mathcal{I}^*(1^\lambda)] \subseteq [\mathcal{I}(1^\lambda)]$ for all $\lambda \in \mathbb{N}$. For $\lambda \in \mathbb{N}$ let

$$P_{\mathcal{I}^*,\mathcal{I}}(\lambda) = \Pr\left[\, \mathbf{x} \in [\mathcal{I}^*(1^\lambda)] \;:\; \mathbf{x} \leftarrow\$ \mathcal{I}(1^\lambda) \,\right]$$

be the probability of the support of $\mathcal{I}^*$ under the distribution induced by $\mathcal{I}$. For $\alpha \in \mathbb{N}$ we say that $\mathcal{I}^*$ is an α-induced input sampler of $\mathcal{I}$ if the following are true. First,

$$\Pr\left[\, \mathbf{x} = \mathbf{x}^* \;:\; \mathbf{x} \leftarrow\$ \mathcal{I}^*(1^\lambda) \,\right] = \frac{\Pr\left[\, \mathbf{x} = \mathbf{x}^* \;:\; \mathbf{x} \leftarrow\$ \mathcal{I}(1^\lambda) \,\right]}{P_{\mathcal{I}^*,\mathcal{I}}(\lambda)}$$

for every $\mathbf{x}^* \in [\mathcal{I}^*(1^\lambda)]$ and every $\lambda \in \mathbb{N}$. Second, $P_{\mathcal{I}^*,\mathcal{I}}(\lambda) \geq 2^{-\alpha}$ for all $\lambda \in \mathbb{N}$.

Let F be a function family that is one-way with respect to input sampler $\mathcal{I}$. Let $\alpha \in \mathbb{N}$. We say that F is α-one-way with respect to $\mathcal{I}$ if for any α-induced input sampler $\mathcal{I}^*$ of $\mathcal{I}$, family F is one-way with respect to $\mathcal{I}^*$. Let H be a function family that is hardcore for F with respect to $\mathcal{I}$. We say that H is α-robust for F with respect to $\mathcal{I}$ if, for any α-induced input sampler $\mathcal{I}^*$ of $\mathcal{I}$, family H is hardcore for F with respect to $\mathcal{I}^*$. The following says that in the above construction $\mathsf{D\text{-}PKE} = \mathsf{EwHCore}[\mathsf{PKE}, \mathsf{F}, \mathsf{H}]$ of a D-PKE scheme, 2-robustness of the hardcore function family H for F with respect to $\mathcal{I}$ suffices for PRIV-security of $\mathsf{D\text{-}PKE}$ over $\mathcal{I}$:

Lemma B.2 ([26]) *Let* PKE *be an IND-CPA secure PKE scheme. Let* F *be an injective trapdoor function family that is one-way with respect to input sampler* $\mathcal{I}$*. Let* H *be a function family that is* 2*-robust for* F *with respect to* $\mathcal{I}$*. Then* $\mathsf{D\text{-}PKE} = \mathsf{EwHCore}[\mathsf{PKE}, \mathsf{F}, \mathsf{H}]$ *is PRIV-secure on* $\mathcal{I}$*.*

To prove Theorem B.1, it thus suffices to show that, under the conditions of the theorem, there exists a function family H that is 2-robust for F with respect to $\mathcal{I}$. The following says that our construction of Section 4 has the desired property, and is of independent interest:

Theorem B.3 *Let* F *be an injective function family. Let* G *be a punctured PRF with* $\mathsf{G.il} = \mathsf{F.il}$*. Let* $d\colon \mathbb{N} \to \mathbb{N}$ *be a polynomial. Then there is a polynomial* s *such that the following is true. Let* $\mathcal{I}$ *be any input sampler with* $\mathcal{I}.\mathsf{vl} = d$ *and* $\mathcal{I}.\mathsf{il} = \mathsf{F.il}$*. Let* Obf *be any* $\boldsymbol{S}_{\mathrm{diff}}(d)$*-secure obfuscator. Assume* F *is one-way with respect to* $\mathcal{I}$*. Let* $\alpha \in \mathbb{N}$*. Then the function family* $\mathsf{H} = \mathbf{HC1}[\mathsf{G}, \mathsf{Obf}, s]$ *defined in Section 4 is* α*-robust for* F *with respect to* $\mathcal{I}$*.*

Combining Theorem B.3 with Lemma B.2 proves Theorem B.1. We now proceed to prove Theorem B.3. Fuller, O'Neill and Reyzin [26] generalize a lemma in [8] to show the following:

Lemma B.4 ([26]) *Suppose function family* F *is one-way with respect to* $\mathcal{I}$*. Suppose* $\alpha \in \mathbb{N}$*. Then* F *is also* α*-one-way with respect to* $\mathcal{I}$*.*

One would like to make a similar claim for α-robustness, namely that if H is hardcore for F with respect to $\mathcal{I}$ then H is also α-robust for F with respect to $\mathcal{I}$, but this is not true in general. Crucial to the proof of Lemma B.4 is that one-wayness is a computational problem, meaning it is possible to verify a solution. But hardcore security is a decision problem. As a consequence, we cannot easily extend the proof. We can, however, exploit Theorem 6.1 in conjunction with Lemma B.4 to prove Theorem B.3, thereby concluding the proof of Theorem B.1.

Proof of Theorem B.3: The crucial point is that in Theorem 6.1, the polynomial s, and thus H, do not depend on anything about $\mathcal{I}$ beyond $\mathcal{I}.\mathsf{vl} = d$ and $\mathcal{I}.\mathsf{il} = \mathsf{F.il}$. Thus, the same family H is hardcore for F relative to *any* input sampler $\mathcal{I}$ relative to which F is one-way. But according to Lemma B.4, F continues to be one way relative to any α-induced input sampler $\mathcal{I}^*$ of $\mathcal{I}$, so H will be hardcore for F relative to $\mathcal{I}^*$.

In more detail, suppose $\mathcal{I}^*$ is an α-induced input sampler of $\mathcal{I}$. We need to show that H is hardcore for F with respect to $\mathcal{I}^*$. By Lemma B.4, F is one-way with respect to $\mathcal{I}^*$. Now we simply apply Theorem 6.1 with $\mathcal{I}$ set to $\mathcal{I}^*$. This says that H is hardcore for F with respect to $\mathcal{I}^*$ as desired. ▮