Indistinguishability Obfuscation with Non-trivial Efficiency*

Huijia Lin† Rafael Pass‡ Karn Seth§ Sidharth Telang¶

January 4, 2016

Abstract

It is well known that *inefficient* indistinguishability obfuscators (**iO**) with running time $\text{poly}(|C|, \lambda) \cdot 2^n$, where C is the circuit to be obfuscated, λ is the security parameter, and n is the input length of C, exists *unconditionally*: simply output the function table of C (i.e., the output of C on all possible inputs). Such inefficient obfuscators, however, are not useful for applications.

We here consider **iO** with a slightly "non-trivial" notion of efficiency: the running-time of the obfuscator may still be "trivial" (namely, $\text{poly}(|C|, \lambda) \cdot 2^n$), but we now require that the obfuscated code is just slightly smaller than the truth table of C (namely $\text{poly}(|C|, \lambda) \cdot 2^{n(1-\epsilon)}$, where $\epsilon > 0$); we refer to this notion as ***iO*** *with exponential efficiency*, or simply *exponentially-efficient* ***iO*** *(**XiO**)*. We show that, perhaps surprisingly, under the subexponential LWE assumption, subexponentially-secure **XiO** for polynomial-size circuits implies (polynomial-time computable) **iO** for all polynomial-size circuits.

*This paper appears in PKC 2016.

†University of California at Santa Barbara, Email: rachel.lin@cs.ucsb.edu. Work supported in part by NSF grants CNS-1528178 and CNS-1514526.

‡Cornell University, rafael@cs.cornell.edu. Work supported in part by a Microsoft Faculty Fellowship, Google Faculty Award, NSF Award CNS-1217821, NSF Award CCF-1214844, AFOSR Award FA9550-15-1-0262 and DARPA and AFRL under contract FA8750-11-2-0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

§Cornell University, Email: karn@cs.cornell.edu.

¶Cornell University, Email: sidtelang@cs.cornell.edu.

1 Introduction

The goal of *program obfuscation* is to "scramble" a computer program, hiding its implementation details (making it hard to "reverse-engineer"), while preserving the functionality (i.e, input/output behavior) of the program. In recent years, the notion of *indistinguishability obfuscation (**iO**)* [BGI+01, GGH+13b] has emerged as the central notion of obfuscation. Roughly speaking, this notion requires that obfuscations $\mathbf{iO}(C_1)$, $\mathbf{iO}(C_2)$ of any two *functionally equivalent* circuits C_1 and C_2 (i.e., whose outputs agree on all inputs) from some class $\mathcal{C}$ (of circuits of some bounded size) are computationally indistinguishable.

On the one hand, this notion of obfuscation is strong enough for a plethora of amazing applications (see e.g., [SW14, BCP14, BZ14, GGHR14, BGL+15, CHJV14, KLW14]); on the other hand, it may plausibly exist [GGH+13b, BGK+13, PST14a, GLSW14], whereas stronger notion of obfuscations have run into strong impossibility results, even in idealized models (see e.g., [BGI+01, GK05, CKP15, PS15, MMN15, LPST15])

However, despite all these amazing progress, to date, all candidate constructions of **iO** rely on candidate constructions of *multi-linear maps* [GGH13a, CLT13, GGH15, CLT15], all of which have non-trivial attacks [CHL+15, MF15], and it is not clear to what extent the security of the obfuscators that rely on them are affected.

Can *Inefficient* iO be Useful? Let us emphasize that for all known application of **iO**, it is important that the obfuscator is *efficient*—namely, polynomial-time. Indeed, as already observed by [BGI+01], it is "trivial" to provide an *inefficient* **iO** with running time $\mathrm{poly}(|C|, \lambda) \cdot 2^n$, where C is the circuit to be obfuscated, λ is the security parameter, and n is the input length of C, exists *unconditionally*: simply output the function table of C (i.e., the output of C on all possible inputs). Recall that, in contrast, for "standard" (efficient) **iO**, the running time and size of the obfuscator is required to be $\mathrm{poly}(|C|, \lambda)$—namely, *polylogarithmic* in the size of the truth table of C).

In this paper, we consider **iO** with just a slightly "non-trivial" notion of efficiency: the running-time of the obfuscator may still be "trivial" (namely, $\mathrm{poly}(|C|, \lambda) \cdot 2^n$), but we now require that the obfuscated code is just slightly smaller than the truth table of C (namely $\mathrm{poly}(|C|, \lambda) \cdot 2^{n(1-\epsilon)}$, where $\epsilon > 0$); we refer to this notion as ***iO** with exponential efficiency*, or simply *exponentially-efficient **iO** (**XiO**)*. The main question investigated in this paper is the following:

> *Can **iO** with just slightly non-trivial efficiency be useful for applications?*

Main Theorem Perhaps surprisingly, we show that in the regime of subexponential security, under the LWE assumption, **XiO** for $\mathsf{P/poly}$ implies (standard) **iO** for $\mathsf{P/poly}$.

Theorem 1. *Assume subexponential security of the LWE assumption, and the existence of subexponentially secure **XiO** for $\mathsf{P}^{\log}/\mathsf{poly}$. Then there exists subexponentially secure **iO** for $\mathsf{P/poly}$.*

Let us remark that in the proof of Theorem 1, we only employ the **XiO** on circuits that take inputs of length $O(\log \lambda)$ (it would be surprising if we didn't since we aim is to achieve an obfuscator with polynomial efficiency). As a consequence, the proof of Theorem 1 also shows that (under the subexponential LWE assumption), subexponentially secure **XiO** for circuits with such "short" inputs (i.e., inputs of length $O(\log \lambda)$)—we refer to this class of circuits as $\mathsf{P}^{\log}/\mathsf{poly}$—implies **iO** for all polynomial-size circuits (with"long" inputs).[1] We remark that in [BGL+15], the authors

[1] "Short-input" **iO** is more appealing than standard **iO** (for $\mathsf{P/poly}$) in the sense that it can be efficiently checked whether an attack on a candidate scheme succeeds [Nao03] (an attacker needs to come up with two circuits C_1, C_2 that

(implicitly) considered a notion of "short-input" **iO** (as opposed to **XiO**) and demonstrate that for *some* (but far from all) applications of **iO**, this weaker notion actually suffices. Our results show that in the regime of subexponential security, "short-input" **iO** (and in fact, even **XiO**) implies standard **iO** (and thus suffices for all applications of **iO**).

Techniques Our starting point are the recent beautiful works by Ananth and Jain [AJ15] and Bitansky and Vaikuntanathan [BV15] which show that the existence of subexponentially-secure *functional encryption with sublinearly compact ciphertexts (a.k.a. sublinear compact FE)* for $\mathsf{P/poly}$ implies **iO** for $\mathsf{P/poly}$. Roughly speaking, a (single-key) functional encryption scheme is a public-key encryption scheme for which it is possible to release a (single) functional secret-key sk_C (for circuit C of some a-priori bounded size S) such that knowledge of sk_C enables efficiently computing $C(m)$ given any encryption of the message m, (but nothing more); sublinear compactness means that the encryption time is sublinear in the upper bound S on the circuit-size.[2] We recently demonstrated in [LPST15] that assuming subexponential LWE, it in fact suffices to start off with an FE satisfying an even weaker notion of compactness—which we refer to as *weak sublinear compactness*—which simply requires that the *size* of the ciphertext (but not the encryption time) is sublinear in the circuit-size.

Our main technical contribution will be showing that **XiO** for $\mathsf{P}^{\log}\mathsf{/poly}$ implies weakly sublinear compact FE for $\mathsf{P/poly}$, which by the above-mentioned result implies our main theorem.

Theorem 2. *Assume the LWE assumption (resp. subexponential security of the LWE assumption) holds, and the existence of* ***XiO*** *for* $\mathsf{P}^{\log}\mathsf{/poly}$ *(resp. subexponentially-secure* ***XiO*** *for* $\mathsf{P}^{\log}\mathsf{/poly}$*). Then there exists weakly sublinear compact FE for* $\mathsf{P/poly}$ *(resp. subexponentially-secure weakly sublinear compact FE for* $\mathsf{P/poly}$*).*

Note that Theorem 2 is interesting in its own right as it applies also in the regime of polynomial security.[3]

The proof of Theorem 2 proceeds as follows. Following a proof template from [AJ15] (we discuss this result in more detail below), we start off with the result of Goldwasser et al [GKP+13] which shows that under the LWE assumption, there exists a functional encryption scheme for *boolean* functions (i.e., functions with 1-bit outputs) in NC^1 that has *logarithmic* compactness. Combined with the bootstrapping result of [ABSV14], this can be used to construct a functional encryption scheme for *boolean* functions in $\mathsf{P/poly}$ that still has logarithmic compactness. We next show how to use **XiO** for $\mathsf{P}^{\log}\mathsf{/poly}$ to extend any such compact FE scheme for boolean functions to one that handles arbitrary polynomial-sized circuits (with potentially long outputs). ([AJ15] provided a similar transformation assuming, so-called, *compact randomized encoding (for Turing machines)* instead of **XiO**.)

We now turn to describe our transformation from "single-bit compact FE" to "multi-bit weakly sublinear compact FE". As an initial approach, instead of simply encrypting a message m, encrypt the sequence $(m;1), (m;2), \ldots (m;\ell)$, where ℓ is the maximum output length of the class of functions

are functionally equivalent for which it can distinguish obfuscations; checking whether two circuits are functionally equivalent may be hard in general, but becomes efficient if the circuits are restricted to inputs of length $O(\log \lambda)$ by simply enumerating all inputs).

[2] More precisely, in a functional encryption scheme $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, Setup samples a public-key, secret-key pair (pk, msk), $\mathsf{KeyGen}(msk, C)$ generates the functional secret key sk_C; $\mathsf{Enc}(pk, m)$ outputs an encryption c of m, and $\mathsf{Dec}(sk_C, c)$ outputs $C(m)$ if c is an encryption of m.

[3] Furthermore, as we remark later on, weakly sublinear compact FE trivially implies a variant of **XiO** and this variant of **XiO** is also sufficient for our theorems. As such, by our results, **XiO** may be viewed as a new way to characterize the complexity of weakly sublinear compact FE.

we want to be able to evaluate. Then, instead of simply releasing a functional secret key for a circuit C, release a secret key for the function $C'(m;i) = C_i(m)$, where $C_i(m)$ denotes the ith output bit of $C(m)$. This approach clearly enables evaluating circuits with multi-bit outputs; but the encryption scheme is no longer (even weakly) compact! The length of the ciphertext grows *linearly* with the number of output bits. To retain compactness (or at least weakly sublinear compactness), we have the encryption algorithm release an obfuscation of a program Π that generates all the ℓ encryptions—more precisely, given an index i, it applies a PRF (with a hard-coded seed) to the index i to generate randomness r_i and then outputs an encryption of $(m;i)$. As long as obfuscation size is "just-slightly-compressing", the functional encryption will have weak sublinear compactness; furthermore, the program we obfuscate only needs to take inputs of length $O(\log \lambda)$. Thus, it suffices to assume the obfuscator satisfies **XiO** for $\mathsf{P}^{\log}/\mathsf{poly}$.

To prove security of the construction, we use the "one-input-at-a-time" technique from [BCP14, GLW14, PST14b, GLSW14, CLTV15], and the punctured program technique of Sahai and Waters [SW14]; the crucial point that enables us to keep the obfuscation small is that the output of the program Π on different inputs uses independent randomness (since they are independent encryptions) and thus in the hybrid arguments it suffices to puncture the PRF on a single point.

Let us end this section by briefly comparing our transformation to the above-mentioned transformation by Ananth and Jain [AJ15]; [AJ15] shows how to use, so-called, "compact randomized encoding" to transform single-bit compact FE for NC^1 into multi-bit compact FE for NC^1. As we explain in more detail in Remark 3, compact randomized encoding can be viewed as a special case of **XiO** for the class of *Turing machines* (as opposed to circuits) with short input. Turing machine obfuscation is a significantly more challenging task than circuit obfuscation. We provide a brief description of their transformation in Appendix A and explain why the transformation fails when using **XiO** (for circuits).

2 Preliminaries

Let $\mathcal{N}$ denote the set of positive integers, and $[n]$ denote the set $\{1, 2, \ldots, n\}$. We denote by PPT probabilistic polynomial time Turing machines, and by nuPPT non-uniform probabilistic polynomial time Turing machines. The term $\mathsf{negligible}$ is used for denoting functions that are (asymptotically) smaller than one over any polynomial. More precisely, a function $\nu(\cdot)$ from non-negative integers to reals is called $\mathsf{negligible}$ if for every constant $c > 0$ and all sufficiently large n, it holds that $\nu(n) < n^{-c}$. For any algorithm A and input x we denote by $\mathsf{outlen}_A(x)$, the output length of A when run with input x.

Definition 1. *We denote by* $\mathsf{P}^{\log}/\mathsf{poly}$ *the class of circuits* $\{\mathcal{C}_\lambda\}$ *where* $\mathcal{C}_\lambda$ *are* $\mathrm{poly}(\lambda)$*-size circuits that have input length* $c \log \lambda$ *for some constant* c*.*

2.1 Puncturable PRF

Puncturable PRFs defined by Sahai and Waters [SW14], are PRFs for which a key can be given out that allows evaluation of the PRF on all inputs, except for a designated polynomial-size set of inputs.

Definition 2 (Puncturable PRF [SW14])**.** *A puncturable pseudo-random function* F *is given by a triple of efficient algorithms* ($\mathsf{F.Key}$, $\mathsf{F.Punc}$, $\mathsf{F.Eval}$)*, and a pair of computable functions* $n(\cdot)$ *and* $m(\cdot)$*, satisfying the following conditions:*

- **Functionality preserved under puncturing**: *For every polynomial size set $S \subseteq \{0,1\}^{n(\lambda)}$ and for every $x \in \{0,1\}^{n(\lambda)} \setminus S$, we have that:*

$$\Pr[K \leftarrow \mathsf{F.Key}(1^\lambda), K_S = \mathsf{F.Punc}(K, S) \ : \ \mathsf{F.Eval}(K, x) = \mathsf{F.Eval}(K_S, x)] = 1$$

- **Pseudorandom at punctured points**: *For every polynomial size set $S \subseteq \{0,1\}^{n(\lambda)}$ and for every nuPPT adversary A we have that:*

$$|\Pr[A(K_S, \mathsf{F.Eval}(K, S)) = 1] - \Pr[A(K_S, U_{m(\lambda)\cdot|S|}) = 1]| = \mathrm{negl}(\lambda)$$

where $K \leftarrow \mathsf{F.Key}(1^\lambda)$ and $K_S = \mathsf{F.Punc}(K, S)$ and $\mathsf{F.Eval}(K, S)$ denotes the concatenation of $\mathsf{F.Eval}(K, x_1), \ldots \mathsf{F.Eval}(K, x_k)$ where $S = \{x_1, ..., x_k\}$ is the enumeration of the elements of S in lexicographic order, U_ℓ denotes the uniform distribution over ℓ bits.

The GGM tree-based construction of PRFs [GGM86] from one-way functions are easily seen to yield puncturable PRFs, as recently observed by [BW13, BGI14, KPTZ13]. Furthermore, it is easy to see that if the PRG underlying the GGM construction is sub-exponentially hard (and this can in turn be built from sub-exponentially hard OWFs), then the resulting puncturable PRF is sub-exponentially pseudorandom.

2.2 Functional Encryption

We recall the definition of public-key functional encryption (FE) with selective indistinguishability based security [BSW12, O'N10]. We note that in this work, we only need the security of the functional encryption scheme to hold with respect to statically chosen challenge messages and functions. We further consider FE schemes that only produce a single functional secret key for each public key.

Definition 3 (Functional Encryption [O'N10, BSW12])**.** *A public key functional encryption scheme for a class of circuits $\{\mathcal{C}_\lambda\}$ is a tuple of PPT algorithms* $(\mathsf{FE.Setup}, \mathsf{FE.KeyGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *that behave as follows:*

- $(msk, pk) \leftarrow \mathsf{FE.Setup}(1^\lambda)$: $\mathsf{FE.Setup}$ *takes as input the security parameter λ and outputs the master secret key msk and public key pk.*
- $sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C)$: $\mathsf{FE.KeyGen}$ *takes as input the master secret key and a circuit $C \in \mathcal{C}_\lambda$ and outputs the functional secret key sk_C.*
- $c \leftarrow \mathsf{FE.Enc}(pk, m)$: $\mathsf{FE.Enc}$ *takes as input the public key and message $m \in \{0,1\}^*$ and outputs the ciphertext c.*
- $y \leftarrow \mathsf{FE.Dec}(sk_C, c)$: $\mathsf{FE.Dec}$ *takes as input the functional secret key and ciphertext and outputs $y \in \{0,1\}^*$.*

We require the following conditions to hold:

- **Correctness**: *For every $\lambda \in \mathbb{N}$, $C \in \mathcal{C}_\lambda$ with input length n and message $m \in \{0,1\}^n$, we have that*

$$\Pr\left[\begin{array}{rl} (pk, msk) \leftarrow \mathsf{FE.Setup}(1^\lambda) & \\ sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C) & : C(m) = \mathsf{FE.Dec}(sk_C, c) \\ c \leftarrow \mathsf{FE.Enc}(pk, m) & \end{array}\right] = 1$$

- **Selective Security**: *For every nuPPT A there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$, every circuit $C \in \mathcal{C}_\lambda$ with input length n and pair of messages $m_0, m_1 \in \{0,1\}^n$ such that $C(m_0) = C(m_1)$ we have that $|\Pr[A(\mathcal{D}_0) = 1] - \Pr[A(\mathcal{D}_1) = 1]| \leq \mu(\lambda)$ where*

$$\mathcal{D}_b = Pr\left[\begin{array}{c} (pk, msk) \leftarrow \mathsf{FE.Setup}(1^\lambda) \\ sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C) \\ c_b \leftarrow \mathsf{FE.Enc}(pk, m_b) \end{array} : (pk, sk_C, c_b)\right]$$

We say the scheme has sub-exponential security *if there exists a constant ϵ such that for every λ, every 2^{λ^ϵ}-size adversary A, $|\Pr[A(\mathcal{D}_0) = 1] - \Pr[A(\mathcal{D}_1) = 1]| \leq 1/2^{\lambda^\epsilon}$ where $\mathcal{D}_b$ is defined above.*

We recall the definition of compactness and succinctness for functional encryption schemes, as defined in [BV15, AJ15].

Definition 4 (Compact Functional Encryption [BV15, AJ15])**.** *We say a functional encryption scheme for a class of circuits $\{\mathcal{C}_\lambda\}$ is* compact *if for every $\lambda \in \mathbb{N}$, $pk \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $m \in \{0,1\}^*$ we have that*

$$\mathsf{Time}(\mathsf{FE.Enc}(pk, m)) = \mathrm{poly}(\lambda, |m|, \log s)$$

where $s = \max_{C \in \mathcal{C}_\lambda} |C|$. We say the scheme has sub-linear compactness *if the running time of* $\mathsf{FE.Enc}$ *is bounded as*

$$\mathsf{Time}(\mathsf{FE.Enc}(pk, m)) = \mathrm{poly}(\lambda, |m|) \cdot s^{1-\epsilon}$$

where $\epsilon > 0$.

Definition 5 (Succinct Functional Encryption)**.** *A compact functional encryption scheme for a class of circuits that output only a* single bit *is called a* succinct *functional encryption scheme.*

Theorem 3 ([GKP+13])**.** *Assuming (sub-exponentially secure) LWE, there exists a (sub-exponentially secure) succinct functional encryption scheme for NC^1.*

We note that [GKP+13] do not explicitly consider sub-exponentially secure succinct functional encryption, but their construction satisfies it (assuming sub-exponentially secure LWE). Additionally, we have the following bootstrapping theorem:

Theorem 4 ([GHRW14, ABSV14, AJ15])**.** *Assuming the existence of symmetric-key encryption with decryption in NC^1 (resp. sub-exponentially secure) and succinct functional encryption for NC^1 (resp. sub-exponentially secure), there exists succinct functional encryption for $\mathsf{P/poly}$ (resp. sub-exponentially secure).*

Following [LPST15], we here also consider a weaker compactness notion, where only the ciphertext *size* (but not the encryption time) is sublinear in the output length of the function being evaluated.

Definition 6 (Weakly Sublinear Compact Functional Encryption [LPST15])**.** *We say a functional encryption scheme for a class of circuits $\{\mathcal{C}_\lambda\}$ is* weakly sublinear compact *if there exists $\epsilon > 0$ such that for every $\lambda \in \mathbb{N}$, $pk \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $m \in \{0,1\}^*$ we have that*

$$\mathsf{Time}_{\mathsf{FE.Enc}}(pk, m) = \mathrm{poly}(\lambda, |m|, s)$$
$$\mathsf{outlen}_{\mathsf{FE.Enc}}(pk, m) = s^{1-\epsilon} \cdot \mathrm{poly}(\lambda, |m|)$$

where $s = \max_{C \in \mathcal{C}_\lambda} |C|$.

2.3 Indistinguishability Obfuscation

We recall the notion of indistinguishability obfuscation (**iO**).

Definition 7 (Indistinguishability Obfuscator [BGI+01, GGH+13b])**.** *A PPT machine* iO *is an* indistinguishability obfuscator *(also referred to as* ***iO****) for a circuit class* $\{\mathcal{C}_\lambda\}_{\lambda\in\mathcal{N}}$ *if the following conditions are satisfied:*

- **Functionality:** *for all security parameters* $\lambda \in \mathbb{N}$*, for all* $C \in \mathcal{C}_\lambda$*, for all inputs* x*, we have that*
$$\Pr[C' \leftarrow \mathsf{iO}(C) \ : \ C'(x) = C(x)] = 1 \ .$$
- **Indistinguishability:** *for any polysize distinguisher* $\mathcal{D}$*, there exists a negligible function* μ *such that the following holds: For all security parameters* $\lambda \in \mathbb{N}$*, for all pairs of circuits* $C_0, C_1 \in \mathcal{C}_\lambda$ *of the same size, we have that if* $C_0(x) = C_1(x)$ *for all inputs* x*, then*
$$\left| \Pr\left[\mathcal{D}(\mathsf{iO}(C_0)) = 1\right] - \Pr\left[\mathcal{D}(\mathsf{iO}(C_1)) = 1\right]\right| \leq \mu(\lambda) \ .$$
We say the scheme has sub-exponential security *if there exists a constant* ϵ *such that for every* λ*, every* 2^{λ^ϵ}*-size adversary* $\mathcal{D}$*,* $|\Pr[\mathcal{D}(\mathsf{iO}(C_0)) = 1] - \Pr[\mathcal{D}(\mathsf{iO}(C_1)) = 1]| \leq 1/2^{\lambda^\epsilon}$*.*

The recent beautiful results of [AJ15], Bitansky and Vaikuntanathan [BV15] show that subexponentially secure sublinear compact functional encryption schemes implies **iO** for $\mathsf{P/poly}$. In an earlier work [LPST15], we demonstrated that (if we additionally assume subexponential LWE), it suffices to start off with just a *weakly* sublinear compact functional encryption scheme (recall that in such a scheme only the length of the ciphertext needs to be sublinear, but encryption time may be polynomial).

Theorem 5 ([LPST15])**.** *Assume the existence of sub-exponentially secure LWE. If there exists a weakly sublinear compact functional encryption scheme for* $\mathsf{P/poly}$ *with sub-exponential security, then there exists a sub-exponentially secure indistinguishability obfuscator for* $\mathsf{P/poly}$*.*

3 Exponentially-Efficient iO (XiO)

In this section, we define our new notion of exponentially-efficient indistinguishability obfuscation (**XiO**), which allows the obfuscator to have running time as long as a brute-force canonicalizer that outputs the entire truth table of the function, but requires the obfuscated program to be slightly smaller in size than a brute-force canonicalization.

Definition 8 (Exponentially-Efficient Indistinguishability Obfuscation (**XiO**))**.** *A machine* XiO *is an* exponentially-efficient indistinguishability obfuscator *(also referred to as* ***XiO****) for a circuit class* $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ *if it satisfies the same functionality and indistinguishability property of indistinguishability obfuscators as in Definition 7 and the following efficiency requirement.*

- **Non-trivial Efficiency**[4]*. There exists a constant* $\epsilon > 0$ *such that for any security parameter*

[4]Our notion of "trivial" running-time is even more relaxed than the notion used in the introduction. We here allow the running-time be polynomial in 2^n, and opposed to just linear (as we described it in the introduction). This even more relaxed notion of efficiency is useful in order to more cleanly compare **XiO** with the notion of compact FE; see Remark 2.

$\lambda \in \mathbb{N}$, *circuit* $C \in \mathcal{C}_\lambda$ *with input length* n *and* $C' \in \mathsf{XiO}(1^\lambda, C)$, *we have that*

$$\mathsf{Time}_{\mathsf{XiO}}(1^\lambda, C) = \mathrm{poly}(\lambda, |C|, 2^n)$$
$$\mathsf{outlen}_{\mathsf{XiO}}(1^\lambda, C) = \mathrm{poly}(\lambda, |C|) \cdot 2^{n(1-\epsilon)}$$

Remark 1. *(Circuits with logarithmic input length) Note that if we want the obfuscation to be efficient (i.e., polynomial-time in* λ *and the size of the circuit to be obfuscated), then the above definition is only meaningful when the class of circuits* $\mathcal{C}_\lambda$ *has input length* $O(\log \lambda)$*. Our results in this paper hold assuming* ***XiO*** *for* $\mathsf{P}^{\log}/\mathsf{poly}$*.*

Remark 2. *(****XiO*** *in the preprocessing model and comparison with Compact Functional Encryption) We can consider further a relaxation of the running-time requirement of the obfuscator. The obfuscator may first perform a* long *"pre-processing" step (without having seen the program to be obfuscated), taking time* $\mathrm{poly}(\lambda, s, 2^n)$ *(where* s *is the size bound on circuits to be obfuscated), and outputting a (potentially long) pre-processing public-key* O_{pk}*. The actual obfuscation then takes* O_{pk}*, and the circuit* C *as inputs, runs in time* $\mathrm{poly}(\lambda, s, 2^n)$ *and outputs an obfuscated program of size* $\mathrm{poly}(\lambda, s) \cdot 2^{n(1-\epsilon)}$*, and then the evaluation of the obfuscated program may finally also access the public-key* O_{pk}*. All our results also apply to this relaxed notion of* ***XiO****.*

Additionally, we note that weakly sublinear compact FE directly implies this notion as follows: pre-processing public key O_{pk} *(generated in the pre-processing step) is the public key pk for the FE and the functional secret key* sk_{FT} *corresponding to a* function table generator *program that takes as input a circuit and outputs the function table of it; the obfuscation of a circuit* C *is an encryption of the circuit* C *(w.r.t., the FE public key pk), and evaluation of the obfuscated code uses the functional secret key* sk_{FT} *inside* O_{pk} *to compute the function table of* C *and selects the appropriate output. Sub-linear compactness of the functional encryption scheme implies the obfuscator has exponential efficiency.*

Remark 3. *(Comparison with Compact Randomized Encoding for Turing machines) [AJ15] and [LPST15] study a notion of compact randomized encodings [IK02, AIK04]. Roughly speaking, a* randomized encoding (RE) *is a method for* encoding *a Turing Machine* Π*, an input* x *and a running-time bound* T*, into a randomized encoding* $\widehat{\Pi(x)}$ *from which* $\Pi(x)$ *can be efficiently* decoded*; furthermore the encodings does not leak anything more about* Π *and* x *than what can be (inefficiently) deduced from just the output* $\Pi(x)$ *(truncated at* T *steps).*[5] *A randomized encodings is compact (resp. sublinearly compact) if the encoding time is poly-logarithmic (resp sublinear) in* T *(and polynomial in the size of* Π *and* x*). We note that sublinear compact RE directly implies* ***XiO*** *as follows: to obfuscate a circuit* C*, compute an encoding* $\widehat{FT_C}$ *of the* function table generator *Turing machine* FT_C *that has the circuit* C *hardcoded (i.e.,* FT_C *takes no inputs and simply computes the function table of* C*); evaluation of the obfuscation on an input* i *simply decodes the encoding* $\widehat{FT_C}$ *and picks out the* i*th output. Sublinear compactness of the RE implies that the obfuscator is exponentially-efficient. In fact, this obfuscator has a stronger efficiency guarantee than* ***XiO****: the running time of the obfuscator is* $\mathrm{poly}(\lambda, |C|) \cdot 2^{n(1-\epsilon)}$ *whereas* ***XiO*** *allows for a longer running time.*

In fact, the above methods extend to show that (sublinearly) compact RE implies a notion of ***XiO*** *for* Turing machines*. We note that Turing machine obfuscation is a significantly harder task than circuit obfuscation (indeed, all known construction of Turing machine obfuscators first*

[5] Or equivalently, for any two programs Π_1, Π_2 and inputs x_1, x_2 such that $\Pi_1(x_1) = \Pi_2(x_2)$, a randomized encoding of Π_1, x_1 is indistinguishable from an encoding of Π_2, x_2.

go through circuit obfuscation). We also point out that whereas (subexponentially-secure) ***iO*** *for circuits is known to imply* ***iO*** *for Turing machine [BGL+15, CHJV14, KLW14], these techniques do not apply in the regime of programs with short input (and thus do not seem amenable in the regime of inefficient* ***iO*** *either).*

4 iO from XiO

In this section, we show how to achieve "standard" (polynomially-efficient) **iO** from **XiO**.

4.1 Weakly Sublinear Compact FE from Succinct FE and XiO

We first give our construction of weakly sublinear compact FE from succinct FE and **XiO** for circuits with input-size $O(\log(\lambda))$. At a high-level, our idea is to have the ciphertext for the FE scheme be **XiO** of a circuit that, on input i, generates a succinct FE encryption of (m, i). The secret key corresponding to C consists of a single key for the succinct FE scheme, that, given a ciphertext encrypting (m, i), computes the ith output bit of $C(m)$.

Let F be a puncturable pseudorandom function, XiO be an exponentially-efficient indistinguishability obfuscator for $\mathsf{P}^{\log}/\mathsf{poly}$ and sFE be a succinct functional encryption scheme (resp. with sub-exponential security) for an appropriate class of circuits $\{\mathcal{C}'_\lambda\}$ that includes C' defined below. We define a compact functional encryption scheme FE for a class of poly-size circuits $\{\mathcal{C}_\lambda\}$ as follows:

$(msk, pk) \leftarrow \mathsf{FE.Setup}(1^\lambda)$**:** $\mathsf{FE.Setup}$ is identical to $\mathsf{sFE.Setup}$ and has the same output.

$c \leftarrow \mathsf{FE.Enc}(pk, m)$**:** $\mathsf{FE.Enc}$ samples a puncturable PRF key $K \leftarrow \mathsf{F.Key}(1^\lambda)$ and outputs $\Pi \leftarrow \mathsf{XiO}(1^\lambda, G[pk, K, m])$ where $G[pk, K, m]$ is a circuit with input length $n = \log s$ where $s = \max_{C \in \mathcal{C}_\lambda} \mathsf{outlen}(C)$, defined as follows:

$$G[pk, K, m](i) = \mathsf{sFE.Enc}(pk, (m, i); \mathsf{F.Eval}(K, i))$$

G is padded to be the same size as circuits G' and G'' that we will define later in the security proof. All circuits G, G', and G'' will ultimately have size bounded by $S = \mathrm{poly}(\lambda, |m|, \log s)$ where $s = \max_{C \in \mathcal{C}_\lambda} |C|$, and are padded to size S.

$sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C)$**:** $\mathsf{FE.KeyGen}$ outputs $\mathsf{sFE.KeyGen}(msk, C')$ where C' on input (m, i) outputs the i^{th} bit of $C(m)$, or outputs $\perp$ if i is greater than the output length of C.

$y \leftarrow \mathsf{FE.Dec}(sk_C, \Pi)$**:** $\mathsf{FE.Dec}$ runs $c_i \leftarrow \Pi(i)$ and $y_i \leftarrow \mathsf{sFE.Dec}(sk_C, c_i)$ for every i and outputs $y_1, \ldots y_{2^n}$.

Let $\{\mathcal{C}'_\lambda\}$ be a class of circuits that includes C' as defined above for every $C \in \mathcal{C}_\lambda$.

Theorem 6. *Assuming* F *is a puncturable pseudorandom function (resp. with subexponential security),* XiO *is an exponentially efficient indistinguishability obfuscator for* $\mathsf{P}^{\log}/\mathsf{poly}$ *(resp. with subexponential security) and* sFE *is a succinct functional encryption scheme for* $\{\mathcal{C}'_\lambda\}$ *(resp. with subexponential security), we have that* FE *as defined above is a functional encryption scheme for* $\{\mathcal{C}_\lambda\}$ *with weakly sub-linear compactness (resp. and with subexponential security).*

Proof. We first show weak sublinear compactness of FE. Consider any λ, $C \in \mathcal{C}_\lambda$, message m, $pk \in \mathsf{FE.Setup}(1^\lambda)$ and puncturable PRF key $K \in \mathsf{F.Key}(1^\lambda)$. $\mathsf{Time}(\mathsf{FE.Enc}(pk, m))$ is the time XiO takes

to obfuscate the circuit $G[pk, K, m]$, which is of size $S = \text{poly}(\lambda, |m|, \log s)$ where $s = \max_{C \in \mathcal{C}_\lambda} |C|$. Hence we have that

$$\mathsf{Time}_{\mathsf{XiO}}(1^\lambda, G[pk, K, m]) = \text{poly}(\lambda, |m|, \log s, 2^n) \leq \text{poly}(\lambda, |m|, s)$$
$$\mathsf{outlen}_{\mathsf{XiO}}(1^\lambda, G[pk, K, m]) = \text{poly}(\lambda, |m|, \log s) \cdot 2^{n(1-\epsilon)} \leq \text{poly}(\lambda, |m|) \cdot s^{1-\epsilon'}$$

where ϵ' is a constant with $0 < \epsilon' < \epsilon$.

Next we show the selective security of FE. We proceed by using the "one-input-at-a-time" technique from [BCP14, GLW14, PST14b, GLSW14, CLTV15]. More precisely, we proceed by a hybrid argument where in each hybrid distribution, the circuit being obfuscated, on input i, produces ciphertexts of m_1 when i is less than a "threshold", and ciphertexts of m_0 otherwise. Indistinguishability of neighboring hybrids is shown using the "punctured programming" technique of [SW14], as was done in [CLTV15] for constructing **iO** for probabilistic functions. (This technique is also used extensively in other applications of **iO**, eg., [BGL$^+$15], [CHJV14], [KLW14] and more.)

Assume for contradiction there exists a nuPPT A and polynomial p such that for sufficiently large λ, circuit $C \in \mathcal{C}_\lambda$ and messages m_0, m_1 such that $C(m_0) = C(m_1)$, A distinguishes $\mathcal{D}_0$ and $\mathcal{D}_1$ with advantage $1/p(\lambda)$, where

$$\mathcal{D}_b = \left(\begin{array}{ll} (msk, pk) \leftarrow \mathsf{FE.Setup}(1^\lambda) & \\ K \leftarrow \mathsf{F.Key}(1^\lambda) & : pk, sk_C, \mathsf{XiO}(G[pk, K, m_b]) \\ sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C) & \end{array} \right)$$

For $j \in [\ell]$, we define the j^{th} hybrid distribution H_j as follows:

$$H_j = \left(\begin{array}{ll} (msk, pk) \leftarrow \mathsf{FE.Setup}(1^\lambda) & \\ K \leftarrow \mathsf{F.Key}(1^\lambda) & : pk, sk_C, \mathsf{XiO}(G'[pk, K, j, m_0, m_1]) \\ sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C) & \end{array} \right)$$

where $G'[pk, K, j, m_0, m_1]$, where G' is defined as follows

$$G'[pk, K, j, m_0, m_1](i) = \begin{cases} \mathsf{sFE.Enc}(pk, (m_0, i); \mathsf{F}(K, i)) & \text{if } i > j \\ \mathsf{sFE.Enc}(pk, (m_1, i); \mathsf{F}(K, i)) & \text{if } i \leq j \end{cases}$$

We also require G' to be padded to be of the same size S as $G[pk, K, m]$.

We consider the hybrid sequence $\mathcal{D}_0, H_1, \ldots, H_\ell, \mathcal{D}_1$. By a hybrid argument, there exists a pair of neighboring hybrids in this sequence such that A distinguishes the pair with probability $\frac{1}{p(\lambda) \cdot (\ell+2)} = \frac{1}{\text{poly}(\lambda)}$. We show a contradiction by proving that each pair of neighboring hybrids is computationally indistinguishable.

We first note that $\mathcal{D}_0$ is indistinguishable from H_0. This follows by observing that $G'[pk, K, 0, m_0, m_1]$ is functionally identical to $G[pk, K, m_0]$, and applying the security of XiO. The same argument also shows that H_ℓ is indistinguishable from $\mathcal{D}_1$.

Next, we show H_{j^*} and H_{j^*+1} are indistinguishable for each $j^* \in [\ell]$. Define hybrid distribution H'_0 which is identical to H_{j^*} except that XiO obfuscates a different circuit $G''[pk, K_{j^*}, j^*, m_0, m_1, c]$ where $K_{j^*} \leftarrow \mathsf{F.Punc}(\lambda, j^*)$ and $c \leftarrow \mathsf{sFE.Enc}(pk, (m_0, j^*); R)$ using uniformly sampled randomness R. G'' on input i has the same behavior as G' except $i = j^*$, where it outputs the hardcoded ciphertext c. By the "punctured programming" technique of Sahai-Waters [SW14], which relies on the security of the obfuscator XiO and puncturable PRF F, it follows that for sufficiently large λ, A distinguishes between H_{j^*} and H'_0 with negligible probability.

The puncturing programming technique itself works in two hybrid steps:

- First the circuit G' is replaced with circuit $G''[pk, K_{j^*}, j^*, m_0, m_1, c]$ where the hardwired ciphertext is $c = \mathsf{sFE.Enc}(pk, (m_0, j^*); \mathsf{F}(K, j^*))$, which is the same ciphertext G' previously computed. Since this doesn't change the functionality of the circuit, indistinguishability follows from the security of XiO.

- Second, the hardcoded ciphertext is modified to be generated from real randomness R, and indistinguishability follows from the security of the puncturable PRF.

Next, we define hybrid distribution H'_1 which is identical to H'_0 except that the hardcoded ciphertext c is generated as $\mathsf{sFE.Enc}(pk, (m_1, j^*); R)$ for uniformly sampled randomness R. Since $C(m_0)$ is identical to $C(m_1)$, from the security of sFE, A distinguishes H'_0 and H'_1 with negligible probability.

Finally, note that H'_1 and H_{j^*+1} differ in the same way H'_0 and H_{j^*} do, and are hence indistinguishable by a similar argument. Hence A distinguishes H_{j^*} and H_{j^*+1} with negligible probability and we have a contradiction. This completes the proof.

We note that the proof above is described in terms of computational indistinguishability, but in fact also can be applied to show that FE is subexponentially-secure, if both XiO and sFE are subexponentially secure. □

4.2 Putting Pieces Together

Theorem 7. *Assuming sub-exponentially hard LWE, if there exists a subexponentially secure exponentially efficient indistinguishability obfuscator for* $\mathsf{P}^{\log}/\mathsf{poly}$ *then there exists an indistinguishability obfuscator for* P/poly *with subexponential security.*

Proof. By Theorem 3 and Theorem 4, assuming subexponentially secure LWE, there exists a succinct functional encryption scheme for P/poly that is subexponentially secure. Using this with a subexponentially secure exponentially efficient indistinguishability obfuscator for $\mathsf{P}^{\log}/\mathsf{poly}$, by Theorem 6, we get weakly sublinear compact function encryption for P/poly with sub-exponential selective security. Together with Theorem 5, this gives us **iO** for P/poly. □

Remark 4. *(**XiO** for* NC^1 *suffices) We remark it in fact suffices to assume **XiO** for only* NC^1 *(instead of* P/poly*) if rely on the existence of puncturable PRFs in* NC^1*. Indeed, if encryption algorithm of the succinct FE scheme and the puncturable PRF are both in* NC^1*, then in our construction it suffices to obfuscate* NC^1 *circuits (we also need to verify that the "merged" circuit used in the hybrid argument is in NC1, which directly follows). By the result of [AIK04], assuming the existence of pseudorandom generators in* NC^1*, we can assume without loss of generality that the succinct FE encryption we rely on also has encryption in* NC^1 *(in fact even* NC^0*, but this will not be useful to us): the encryption algorithm for the new succinct FE scheme computes the "randomized encoding" of the original encryption function.*

Acknowledgments: We thank Vinod Vaikuntanathan for insightful discussions.

References

[ABSV14] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. The trojan method in functional encryption: From selective to adaptive security. Technical report, generically. Cryptology ePrint Archive, Report 2014/917, 2014.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. In *FOCS*, pages 166–175, 2004.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. *IACR Cryptology ePrint Archive*, 2015:173, 2015.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *TCC*, pages 52–73, 2014.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Advances in Cryptology CRYPTO 2001*, pages 1–18. Springer, 2001.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *PKC*, pages 501–519, 2014.

[BGK+13] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *EuroCrypt'14*, 2013.

[BGL+15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. *IACR Cryptology ePrint Archive*, 2015:356, 2015.

[BSW12] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *IACR Cryptology ePrint Archive*, 2015:163, 2015.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *ASIACRYPT (2)*, pages 280–300, 2013.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 480–499, 2014.

[CHJV14] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Indistinguishability obfuscation of iterated circuits and RAM programs. *IACR Cryptology ePrint Archive*, 2014:769, 2014.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 3–12, 2015.

[CKP15] Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On obfuscation with random oracles. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 456–467, 2015.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 476–493, 2013.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I*, pages 267–286, 2015.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and JesperBuus Nielsen, editors, *Theory of Cryptography*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer Berlin Heidelberg, 2015.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology–EUROCRYPT 2013*, pages 1–17. Springer, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *Proc. of FOCS 2013*, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 498–527, 2015.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, pages 74–94, 2014.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GHRW14] Gentry, Halevi, Raykova, and Wichs. Outsourcing private ram computation. *Proc. of FOCS 2014*, 2014.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 553–562, 2005.

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 555–564, 2013.

[GLSW14] Craig Gentry, Allison Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. Cryptology ePrint Archive, Report 2014/309, 2014.

[GLW14] Craig Gentry, Allison Lewko, and Brent Waters. Witness encryption from instance independent assumptions. In *Advances in Cryptology–CRYPTO 2014*, pages 426–443. Springer, 2014.

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP*, pages 244–256, 2002.

[KLW14] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. Technical report, Cryptology ePrint Archive, Report 2014/925, 201 4. http://eprint. iacr. org, 2014.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *CCS*, pages 669–684, 2013.

[LPST15] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. 2015.

[MF15] Brice Minaud and Pierre-Alain Fouque. Cryptanalysis of the new multilinear map over the integers. Cryptology ePrint Archive, Report 2015/941, 2015. `http://eprint.iacr.org/`.

[MMN15] Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. More on impossibility of virtual black-box obfuscation in idealized models. *IACR Cryptology ePrint Archive*, 2015:632, 2015.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2003.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/.

[PS15] Rafael Pass and Abhi Shelat. Impossibility of VBB obfuscation with ideal constant-degree graded encodings. *IACR Cryptology ePrint Archive*, 2015:383, 2015.

[PST14a] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *CRYPTO'14*, 2014.

[PST14b] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *Advances in Cryptology–CRYPTO 2014*, pages 500–517. Springer, 2014.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *Proc. of STOC 2014*, 2014.

A Comparison with [AJ15]

In this section we briefly describe the related result by [AJ15] and compare it with our result. [AJ15] show how to construct a compact functional encryption scheme from a succinct functional encryption scheme and "compact randomized encodings for Turing machines" (see Remark 3 for an informal description of randomized encodings). The rough idea is as follows: the compact functional

secret key for a function f is a sequence of ℓ independent succinct functional secret keys where ℓ is the output length of f. The i^{th} succinct functional secret key corresponds to the function that outputs the i^{th} bit of f. The compact functional ciphertext for a message m is the randomized encoding of a machine Π that takes no input and when run, outputs $\{Enc(pk_i, m)\}_{i\in[\ell]}$ where pk_i is the public key corresponding to the i^{th} instance of the succinct functional scheme (these instances are generated using a PRF, hence the description size of Π is independent of ℓ). The compactness of the functional encryption scheme follows from the compactness of the randomized encoding scheme.

Note that the above result necessarily requires the computation being encoded to be represented as a Turing machine, since the description size is required to be independent of the output length. As we explain in Remark 3, such a notion of randomized encodings for Turing machine does not seem useful for our purposes.