Output-Compressing Randomized Encodings and Applications

Huijia Lin[1]*, Rafael Pass[2]**, Karn Seth[2], and Sidharth Telang[2].

[1] University of California at Santa Barbara
rachel.lin@cs.ucsb.edu
[2] Cornell University
{rafael,karn,sidtelang}@cs.cornell.edu

Abstract. We consider *randomized encodings (RE)* that enable encoding a Turing machine Π and input x into its "randomized encoding" $\hat{\Pi}(x)$ in sublinear, or even polylogarithmic, time in the running-time of $\Pi(x)$, *independent of its output length*. We refer to the former as *sublinear RE* and the latter as *compact RE*. For such efficient RE, the standard simulation-based notion of security is impossible, and we thus consider a weaker (distributional) indistinguishability-based notion of security: Roughly speaking, we require indistinguishability of $\hat{\Pi}_0(x_0)$ and $\hat{\Pi}_0(x_1)$ as long as Π_0, x_0 and Π_1, x_1 are sampled from some distributions such that $\Pi_0(x_0), \mathsf{Time}(\Pi_0(x_0))$ and $\Pi_1(x_1), \mathsf{Time}(\Pi_1(x_1))$ are indistinguishable.

We show the following:

- **Impossibility in the Plain Model:** Assuming the existence of subexponentially secure one-way functions, subexponentially-secure sublinear RE does not exists. (If additionally assuming subexponentially-secure **iO** for circuits we can also rule out polynomially-secure sublinear RE.) As a consequence, we rule out also puncturable **iO** for Turing machines (even those without inputs).
- **Feasibility in the CRS model and Applications to iO for circuits:** Subexponentially-secure sublinear RE in the CRS model and one-way functions imply **iO** for circuits through a simple construction generalizing GGM's PRF construction. Additionally, any compact (even with sublinear compactness) functional encryption essentially directly yields a sublinear RE in the CRS model, and as such we get an alternative, modular, and simpler proof of the results of [AJ15,BV15] showing that subexponentially-secure sublinearly compact FE implies **iO**. We further show other ways of instantiating sublinear RE in the CRS model (and thus also **iO**): under the

* Work supported in part by a NSF award CNS-1514526.

** Work supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF Award CNS-1217821, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, AFOSR YIP Award FA9550-10-1-0093, 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.

subexponential LWE assumption, it suffices to have a subexponentially secure FE schemes with just *sublinear ciphertext* (as opposed to having sublinear encryption time).

– **Applications to iO for Unbounded-input Turing machines:** Subexponentially-secure compact RE for natural *restricted* classes of distributions over programs and inputs (which are not ruled out by our impossibility result, and for which we can give candidate constructions) imply **iO** for *unbounded-input* Turing machines. This yields the first construction of **iO** for unbounded-input Turing machines that does not rely on (public-coin) differing-input obfuscation.

1 Introduction

The beautiful notion of a *randomized encoding (RE)*, introduced by Ishai and Kushilevitz [IK00], aims to trade the computation of a "complex" (deterministic) function Π on a given input x for the computation of a "simpler" randomized function—the "encoding algorithm"—whose output distribution $\hat{\Pi}(x)$ encodes $\Pi(x)$ (from which $\Pi(x)$ can be efficiently decoded, or "evaluated"). Furthermore, the encoding $\hat{\Pi}(x)$ should not reveal anything beyond $\Pi(x)$; this is referred to as the *privacy*, or *security*, property of randomized encodings and is typically defined through the simulation paradigm [GMR89].

Most previous work have focused on randomized encodings where encodings can be computed in lower parallel-time complexity than what is required for computing the original function Π. For instance, all log-space computations have *perfectly-secure* randomized encodings in $\mathbf{NC}^0$ [IK00,IK02a,AIK04], and assuming low-depth pseudo-random generators, this extends to all polynomial-time computations (with computational security) [AIK06,Yao82]. Such randomized encodings have been shown to have various applications to parallel cryptography, secure computation, verifiable delegation, etc. (see [App11] for a survey).

Bitansky, Garg, Lin, Pass and Telang [BGL$^+$15] recently initiated a study of *succinct randomized encodings* where we require that the *time* required to compute $\hat{\Pi}(x)$ is smaller than the time required to compute $\Pi(x)$; their study focused on functions Π that have *single-bit* outputs. [BGL$^+$15,CHJV14,KLW14] show that subexponentially-secure indistinguishability obfuscators (**iO**) [BGI$^+$01,GGH$^+$13] and one-way functions[3] imply the existence of such succinct randomized encodings for all polynomial-time Turing machines that output just a single bit.

We here further the study of such objects, focusing on functions Π with *long* outputs. Given a description of a Turing machine Π and an input x, we consider two notions of efficiency for randomized encodings $\hat{\Pi}(x)$ of $\Pi(x)$ with running time T.

– *compact RE*: Encoding time (and thus also size of the encodings) is $\text{poly}(|\Pi|, |x|, \log T)$

[3] The one-way function assumption can be weakened to assume just that $\mathsf{NP} \not\subseteq io\mathsf{BPP}$ [KMN$^+$14].

- *sublinear RE*: Encoding time (and thus also size) is bounded by $\text{poly}(|\mathit{\Pi}|, |x|) * T^{1-\epsilon}$, for some $\epsilon > 0$.

We assume without loss of generality that the randomized encoding $\hat{\mathit{\Pi}}(x)$ of $\mathit{\Pi}, x$ itself is a program, and that the decoding/evaluation algorithm simply executes $\hat{\mathit{\Pi}}(x)$.

It is easy to see that for such notions of efficiency, the standard simulation-based notion of security is impossible to achieve—roughly speaking, the simulator given just $\mathit{\Pi}(x)$ needs to output a "compressed" version of it, which is impossible if $\mathit{\Pi}(x)$ has high pseudo-Kolmogorov complexity (e.g., if $\mathit{\Pi}$ is a PRG); we formalize this argument in Theorem 14 in Section 6. Consequently, we consider weaker indistinguishability-based notions of privacy. One natural indistinguishability based notion of privacy simply requires that encoding $\hat{\mathit{\Pi}}_0(x_0)$ and $\hat{\mathit{\Pi}}_1(x_1)$ are indistinguishable as long as $\mathit{\Pi}_0(x_0) = \mathit{\Pi}_1(x_1)$ and $\mathsf{Time}(\mathit{\Pi}_0(x_0)) = \mathsf{Time}(\mathit{\Pi}_1(x_1))$, where $\mathsf{Time}(\mathit{\Pi}(x))$ is the running-time of $\mathit{\Pi}(x)$; such a notion was recently considered by Ananth and Jain [AJ15]. In this work, we consider a stronger notion which requires indistinguishability of $\hat{\mathit{\Pi}}_0(x_0)$ and $\hat{\mathit{\Pi}}_0(x_1)$ as long as $\mathit{\Pi}_0, x_0$ and $\mathit{\Pi}_1, x_1$ are sampled from some distributions such that $\mathit{\Pi}_0(x_0), \mathsf{Time}(\mathit{\Pi}_0(x_0))$ and $\mathit{\Pi}_1(x_1), \mathsf{Time}(\mathit{\Pi}_1(x_1))$ are indistinguishable. We refer to this notion as *distributional indistinguishability security*, and note that it easily follows that the standard simulation-based security implies distributional indistinguishability security.

The goal of this paper is to investigate compact and sublinear RE satisfying the above-mentioned distributional indistinguishability notion. For the remainder of the introduction, we refer to randomized encodings satisfying distributional indistinguishability security as simply *RE*. For comparison, we refer to randomized encodings with the weaker (non-distributional) indistinguishability security as *weak RE*.

Compact RE v.s. Obfuscation Before turning to describe our results, let us point out that RE can be viewed as (a degenerate form) of obfuscation for special classes of programs.

Recall that an indistinguishability obfuscator (**iO**) [BGI+01,GGH+13] is a method $\mathcal{O}$ for "scrambling" a program $\mathit{\Pi}$ into $\mathcal{O}(\mathit{\Pi})$ such that for any two functionally equivalent programs $\mathit{\Pi}_0, \mathit{\Pi}_1$ (that is, their outputs and run-time are the same on all inputs,) $\mathcal{O}(\mathit{\Pi}_0)$ is indistinguishable from $\mathcal{O}(\mathit{\Pi}_1)$. **iO** for Turing machines [BGI+01,BCP14,ABG+13] additionally requires that the size of the obfuscated code does not grow (more than polylogarithmically) with the running-time of the Turing machine.

We may also consider a useful strengthening of this notion—which we call "puncturable **iO**"—which, roughly speaking, requires indistinguishability of $\mathcal{O}(\mathit{\Pi}_0)$ and $\mathcal{O}(\mathit{\Pi}_1)$ as long as $\mathit{\Pi}_0$ and $\mathit{\Pi}_1$ differ *on at most one input* x^* and their outputs on input x^* are indistinguishable. More precisely, we say that a distribution D is *admissible* if there exists some x^* such that a) for every triple $(\mathit{\Pi}_0, \mathit{\Pi}_1, \mathit{\Pi})$ in the support of D, and every $x \neq x^*$, it holds that $\mathit{\Pi}_0(x) = \mathit{\Pi}_1(x) = \mathit{\Pi}(x)$, and b) $(\mathit{\Pi}, \mathit{\Pi}_0(x^*))$ and $(\mathit{\Pi}, \mathit{\Pi}_1(x^*))$ are computationally indistinguishable when $(\mathit{\Pi}_0, \mathit{\Pi}_1, \mathit{\Pi})$ are sampled randomly from D. Puncturable **iO** requires indistinguishability of $\mathcal{O}(\Pi_0)$ and $iO(\Pi_1)$ for Π_0, Π_1 sampled from any admissible distribution. Interestingly, for the case of *circuits*, puncturable **iO** is equivalent to (standard) **iO**.[4] Indeed, such a notion is implicitly used in the beautiful and powerful punctured-program paradigm by Sahai and Waters [SW14], and all its applications. (In this context, think of Π as the "punctured" version of the programs Π_0, Π_1.)

In the case of Turing machines, when restricting to the degenerate case of Turing machines with no inputs (or more precisely, we only consider the execution of $\Pi()$ on the "empty" input), the notion of **iO** for Turing machines is equivalent to the notion of a compact *weak* RE. Compact RE, on the other hand, is equivalent to puncturable **iO** for Turing machines (without inputs). (Jumping ahead, as we shall see, for the case of Turing machines it is unlikely that puncturable **iO** is equivalent to standard **iO**.)

1.1 Our results

<u>iO from sublinear RE</u> We start by showing that sublinear RE is an extremely useful primitive: Subexponentially-secure sublinear RE implies indistinguishability obfuscators for all polynomial-size circuits.

Theorem 1. *The existence of subexponentially-secure sublinear RE and one-way functions implies the existence of subexponentially-secure **iO** for circuits.*

Before continuing, let us mention that Theorem 1 is related to a recent beautiful result by Ananth and Jain [AJ15] which shows that *under the LWE assumption*, subexponentially-secure compact RE (satisfying only the weak indistinguishability security) implies **iO** for circuits. Their construction goes from RE to *functional encryption* (FE) [BSW11], and then from FE to **iO**; (the first step relies on previous constructions of FE [GKP$^+$13a,GVW13], while the second step relies on a sequence of complex transformations and analysis). In contrast, the proof of Theorem 1 directly constructs **iO** from RE in a surprisingly simple way: We essentially use the GGM construction [GGM86] that builds a PRF from a PRG using a tree, but replace the PRG with a RE. Let us explain in more details below.

Consider a program Π taking n-bit inputs. We consider a binary tree where the leaves are randomized encodings of the function applied to all possible inputs, and each node in the tree is a randomized encoding that generates its two children. More precisely, given a sequence of bits $x_1, \cdots, x_i$, let $\tilde{\Pi}_{R,x_1,\cdots,x_i}$ denote an (input-less) program that

- if $i = n$ simply outputs a RE of the program Π and input $(x_1, \cdots, x_n)$ using R as randomness, and

[4] To see this, consider a hybrid program $\Pi^y(x)$ that runs $\Pi(x)$ if $x \neq x^*$ and otherwise (i.e., if $x = x^*$ outputs y). By the **iO** property we have that for every Π, Π_0, Π_1 in the support of D, $\mathcal{O}(\Pi^{\Pi_b(x^*)})$ is indistinguishable from $\mathcal{O}(\Pi_b)$. Thus, if $\mathcal{O}(\Pi_0)$, $\mathcal{O}(\Pi_1)$ are distinguishable, so are $\mathcal{O}(\Pi^{\Pi_0(x^*)})$, $\mathcal{O}(\Pi^{\Pi_1(x^*)})$, which contradicts indistinguishability of $(\Pi, \Pi_0(x^*))$ and $(\Pi, \Pi_1(x^*))$.

- otherwise, after expanding R_0, R_1, R_2, R_3 from R using a PRG, outputs randomized encodings of (input-less) programs $\tilde{\Pi}_{R_0,x_1,\cdots,x_i,0}$ and $\tilde{\Pi}_{R_1,x_1,\cdots,x_i,1}$ using respectively R_2, R_3 as randomness.

We associate each node in the binary tree that has index $x_1, \cdots, x_i$ with a randomized encoding of the program $\tilde{\Pi}_{R,x_1,\cdots,x_i}$, denoted as $\hat{\Pi}_{R,x_1,\cdots,x_i}$. In particular, the root of the tree is associated with a randomized encoding $\hat{\Pi}$ of the (initial) program $\tilde{\Pi}_R$ hardwired with a randomly chosen R.

The obfuscation of Π is now a program with the "root" $\hat{\Pi}$ hardcoded, and given an input x, computes the path from the root to the leaf x – by recursively evaluating the randomized encodings associated with nodes on the path – and finally outputs the evaluation of the leaf. More precisely, on input x, evaluate $\hat{\Pi}$ to obtain $\hat{\Pi}_0, \hat{\Pi}_1$, next evaluate $\hat{\Pi}_{x_1}$ to obtain $\hat{\Pi}_{x_1,0}, \hat{\Pi}_{x_1,1}$, so on and so forth until $\hat{\Pi}_{x_1,\cdots,x_n}$ is evaluated, yielding the output $\Pi(x_1, \cdots, x_n)$.

Note that for any two functionally equivalent programs, the randomized encodings associated with individual leaf node are computationally indistinguishable by the indistinguishability security property (the non-distributional version suffices here). Then, by the distributional indistinguishability security, the randomized encodings associated with tree nodes one layer above are also indistinguishable. Thus, by induction, it follows that the roots are indistinguishable, which implies that obfuscations of functionally equivalent programs are indistinguishable. Let us note that the reason that subexponential security is needed is that each time we go up one level in the tree (in the inductive argument), we lose at least a factor 2 in the indistinguishability gap (as each node generates two randomized encodings, its children). Hence, we need to ensure that encodings are at least poly(2^n)-indistinguishable, which can be done by scaling up the security parameter.

ON THE EXISTENCE OF COMPACT AND SUBLINEAR RE We next turn to investigating the existence of compact and sublinear RE. We show—assuming just the existence of subexponentially-secure one-way functions—*impossibility* of subexponentially-secure sublinear (and thus also compact) RE.[5]

Theorem 2. *Assume the existence of subexponentially secure one-way functions. Then, there do not exists subexponentially-secure sublinear RE.*

As observed above, compact RE can be interpreted as a stronger notion of **iO** (which we referred to as *puncturable* ***iO***) for "degenerate" input-less Turing machines, and as such Theorem 2 rules out (assuming just one-way functions) such a natural strengthening of **iO** for (input-less) Turing machines. We note that this impossibility stands in contrast with the case of *circuits* where puncturable **iO** is equivalent to **iO**.

[5] This result was established after hearing that Bitansky and Paneth had ruled out compact RE assuming public-coin differing-input obfuscation for Turing Machines and collision-resistant hashfunctions. We are very grateful to them for informing us of their result.

We remark that although it may seem like Theorem 2 makes Theorem 1 pointless, it turns out that Theorem 1 plays a crucial role in the proof of Theorem 2: Theorem 2 is proven by first ruling out sublinear (even just polynomially-secure) RE *assuming* **iO** and one-way functions. Next, by using Theorem 1, the **iO** assumption comes for free if considering subexponentially-secure RE. That is, assuming one-way functions, we have the following paradigm:

$$\text{sub-exp secure sublinear RE} \overset{\text{Theorem 1}}{\Longrightarrow} \textbf{iO} \Longrightarrow \text{impossibility of (poly secure) sublinear RE}$$

Let us now briefly sketch how to rule out sublinear RE assuming **iO** and one-way functions (as mentioned, Theorem 2 is then deduced by relying on Theorem 1). The idea is somewhat similar to the non-black-box zero-knowledge protocol of Barak [Bar01].

Let $\Pi^b_{s,u}$ be a program that takes no input and outputs a sufficiently long pseudo-random string $y = \mathsf{PRG}(s)$ and an indistinguishability obfuscation $\tilde{R}^b_y$ (generated using pseudo-random coins $\mathsf{PRG}(u)$) of the program R^b_y. The program R^b_y takes input Σ of length $|y|/2$, and outputs b iff Σ, when interpreted as an input-less Turing machine, generates y; in all other cases, it outputs $\perp$.[6] We note that the size of the program $\Pi^b_{s,u}$ is linear in the security parameter λ, whereas the pseudo-random string y it generates could have length $|y| = \lambda^\alpha$ for any sufficiently large constant α.

Consider the pair of distributions $\Pi^0_{U_\lambda,U_\lambda}$ and $\Pi^1_{U_\lambda,U_\lambda}$ that samples respectively programs $\Pi^0_{s,u}$ and $\Pi^1_{s,u}$ as described above with random s and u. We first argue that their outputs are computationally indistinguishable. Recall that the output of $\Pi^b_{s,u}$ is a pair $(y, \tilde{R}^b_y)$. By the pseudorandomness of PRG, this output distribution is indistinguishable from $(X, \tilde{R}^b_X)$ where X a uniformly distributed random variable over λ^α bit strings. With overwhelming probability X has high Kolmogorov complexity, and when this happens R^b_X is functionally equivalent to the program $R_\perp$ that always outputs $\perp$. Therefore, by the security of the **iO**, the output of programs sampled from $\Pi^b_{U_\lambda,U_\lambda}$ is computationally indistinguishable to $(X, \tilde{R}_\perp)$, and hence outputs of $\Pi^0_{U_\lambda,U_\lambda}$ and $\Pi^1_{U_\lambda,U_\lambda}$ are indistinguishable.

Let us now turn to showing that randomized encodings of $\Pi^0_{U_\lambda,U_\lambda}$ and $\Pi^1_{U_\lambda,U_\lambda}$ can be distinguished. Recall that a randomized encoding $\hat{\Pi}^b$ of $\Pi^b_{U_\lambda,U_\lambda}$ itself can be viewed as a (input-less) program that outputs $(y, \tilde{R}^b_y)$. Given $\hat{\Pi}^b$, the distinguisher can thus first evaluate $\hat{\Pi}^b$ to obtain $(y, \tilde{R}^b_y)$ and next evaluate $\tilde{R}^b_y(\hat{\Pi}^b)$ to attempt to recover b. Note that $\hat{\Pi}^b$ clearly is a program that generates y (as its first input); furthermore, if the RE scheme is compact, the length of the program $|\hat{\Pi}^b|$ is bounded by $\mathrm{poly}(\lambda, \log \lambda^\alpha)$, which is far smaller than $|y|/2 = \lambda^\alpha/2$ when α is sufficiently large. Therefore, $\Sigma = \hat{\Pi}^b$ is indeed an input that makes $\tilde{R}^b_y$ output b, enabling the distinguisher to distinguish $\hat{\Pi}^0$ and $\hat{\Pi}^1$ with probability close to 1!

[6] To enable this, we require **iO** for bounded-input Turing machines, whereas Theorem 1 only gives us **iO** for circuits. However, by the results of [BGL$^+$15,CHJV14,KLW14] we can go from **iO** for circuits to **iO** for bounded-inputs Turing machines.

Finally, if the RE is only sublinear, the length of the encoding $|\hat{\Pi}^b|$ is only sublinear in the output length, in particular, bounded by $\text{poly}(\lambda)(\lambda^\alpha)^{1-\epsilon}$ for some constant $\epsilon > 0$. If $\alpha > 1/(1-\epsilon)$ (which clearly happens if ϵ is sufficiently small), then we do not get enough "compression" for the above proof to go through. We circumvent this problem by composing a sublinear RE with itself a sufficient (constant) number of times—to compose once, consider creating randomized encoding of the randomized encoding of a function, instead of the function itself; each time of composition reduces the size of the encoding to be w.r.t. a smaller exponent $1 - \epsilon'$. Therefore, it is without loss of generality to assume that ϵ is any sufficiently *big* constant satisfying $\alpha << 1/(1-\epsilon)$; so the desired compression occurs.

Sublinear RE in the CRS model from sublinear FE Despite Theorem 2, not all is lost. We remark that any sublinear functional encryption scheme (FE) [AJ15,BV15] almost directly yields a sublinear RE in the *Common Reference String (CRS)* model; roughly speaking, an FE scheme is called sublinear if the encryption time is sublinear in the size of the circuit that can be evaluated on the encrypted message.

Theorem 3. *Assume the existence of subexponentially-secure sublinear (resp. compact) FE. Then there exists a subexponentially-secures sublinear (resp. compact) RE in the CRS model.*

Furthermore, Theorem 1 straightforwardly extends also to RE in the CRS model. Taken together, these result provide an alternative, modular, simpler proof of the recent results of Ananth and Jain [AJ15] and Bitansky and Vaikuntanathan [BV15] showing that subexponentially-secure sublinear FE implies subexponentially-secure **iO**. (All these approaches, including a related work by Brakerski, Komargodski and Segev [BKS15] have one thing in common though: they all proceed by processing inputs one bit at a time, and hard-coding parts of input to the program.)

Theorem 4 (informal, alternative proof of [BV15,AJ15]). *Assume the existence of subexponentially-secure sublinear FE. Then there exists a subexponentially-secure* ***iO*** *for circuits.*

But there are also other ways to instantiate sublinear RE in the CRS model. We show that under the *subexponential LWE assumption* (relying on [GKP$^+$13b,ABSV14,GVW13]) sublinear RE in the CRS model can be based on a significantly weaker notion of sublinear FE—namely FE schemes where the encryption time may be fully polynomial (in the size of the circuit to be evaluated) but only the *size of the ciphertext* is sublinear in the circuit size—we refer to this notion as a *FE with sublinear ciphertexts*. Roughly speaking, we show this by 1) transforming the "succinct" FE (i.e. compact FE for 1-bit outputs) of [GKP$^+$13b,ABSV14] into an RE which depends linearly on the output length but only polylogarithmically on the running time, 2) transforming an FE with sublinear ciphertext into an RE with "large" running-time but short output, and 3) finally composing the two randomized encodings (i.e, computing the step 1 RE of the step 2 RE).

Combining this result with (the CRS-extended version of) Theorem 1, we get:

Theorem 5 (informal). *Assume the existence of subexponentially-secure FE with sublinear ciphertexts and the subexponential LWE assumption. Then there exists a subexponentially-secure* ***iO*** *for circuits.*

Toward Turing Machine Obfuscation with Unbounded Inputs We finally address the question of constructing indistinguishability obfuscators for Turing machines with *unbounded* inputs. (For the case of Turing machine obfuscation with unbounded-length inputs, the same obfuscated code needs to work for every input-length, and in particular, the size of the obfuscated code cannot grow with it.) Although it is known that subexponentially secure **iO** for circuits implies **iO** for Turing machines with *bounded inputs lengths* [BGL+15,CHJV14,KLW14] , the only known construction of **iO** for Turing machines with unbounded inputs relies on (public-coin) differing-input obfuscation for circuits and (public-coin) SNARKs [BCP14,ABG+13,IPS15]—these are strong "extractability" assumptions (and variants of them are known to be implausible [BCPR13,GGHW13,BP15]).

We note that the construction from Theorem 1 easily extends to show that subexponentially-secure *compact* RE implies **iO** for Turing machines with unbounded input: instead of having a binary tree, we have a ternary tree where the "third" child of a node is always a leaf; that is, for a tree node corresponding to $x_1, \cdots, x_i$, its third child is associated with a randomized encoding of program Π, and input $(x_1, \cdots, x_i)$, which can be evaluated to obtain output $\Pi(x_1, \cdots x_i)$. Then, by using a tree of *super-polynomial* depth, we can handle any polynomial-length input. Note that since obfuscating a program only involves computing the root RE (as before), the obfuscation is still efficient. Moreover, for any input, we still compute the output of the program in time polynomial in the length of the input by evaluating the "third" child of the node when all input bits have been processed.[7]

But as shown in Theorem 2, compact RE cannot exist (assuming one-way functions)! However, just as for the case of differing-inputs obfuscation and SNARKs, we may assume the existence of compact RE for *restricted* types of "nice" distributions (over programs and inputs), for which impossibility does not hold, yet the construction in Theorem 1 still works. We formalize one natural class of such distributions, and may assume that the **iO** for bounded-input Turing machines construction of [KLW14] (based on **iO** for circuits) yields such a compact RE (for the restricted class of distributions). This yields a new candidate construction of unbounded input Turing machines (based on a very different type of assumption than known constructions).

[7] Proving security becomes slightly more problematic since there is no longer a polynomial bound on the depth of the tree (recall that we required poly(2^n)-indistinguishable RE to deal with inputs of length n). Thus issue, however, can be dealt with by using larger and larger security parameters for RE that are deeper down in the tree.

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. The term 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 negligible if for every constant $c > 0$ and all sufficiently large n, it holds that $\nu(n) < n^{-c}$.

Turing machine notation For any Turing machine Π, input $x \in \{0,1\}^*$ and time bound $T \in \mathbb{N}$, we denote by $\Pi^T(x)$ the output of Π on x when run for T steps. We refer to $\{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ as a class of Turing machines. One particular class we will consider is the class of Turing machines that have 1-bit output. We call such a machine a Boolean Turing machine. Throughout this paper, by *Turing machine* we refer to a machine with *multi-bit* output unless we explicitly mention it to be a Boolean Turing machine.

2.1 Concrete Security

Definition 1 ($(\lambda_0, S(\cdot))$-indistinguishability). *A pair of distributions X, Y are S-indistinguishable for some $S \in \mathbb{N}$ if every S-size distinguisher D it holds that*

$$|\Pr[x \xleftarrow{\$} X : D(x) = 1] - \Pr[y \xleftarrow{\$} Y : D(y) = 1]| \leq \frac{1}{S}$$

A pair of ensembles $\{X_\lambda\}$, $\{Y_\lambda\}$ are $(\lambda_0, S(\cdot))$-indistinguishable for some $\lambda_0 \in \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$ if for every security parameter $\lambda > \lambda_0$, the distributions X_λ and Y_λ are $S(\lambda)$ indistinguishable.

Discussion on $(\lambda_0, S(\cdot))$-indistinguishability: We remark that the above definition requires that there is a universal λ_0 that works for all distinguisher D. A seemingly weaker variant could switch the order of quantifiers and only require that for every distinguisher D there is a λ_0. We show that the above definition is w.l.o.g, since it is implied by the following standard definition with auxiliary inputs in the weaker fashion.

Let U be a universal TM that on an input x and a circuit C computes $C(x)$. Let $S' : N \to N$ denote the run time $S'(S)$ of U on input a size S circuit.

Definition 2. *A pair of ensembles $\{X_\lambda\}$, $\{Y_\lambda\}$ are $S(\cdot)$-indistinguishable if for every $S' \circ S(\cdot)$-time uniform TM distinguisher D, there exists a $\lambda_0 \in N$, such that, for every security parameter $\lambda > \lambda_0$, and every auxiliary input $z = z_\lambda \in \{0,1\}^*$,*

$$|\Pr[x \xleftarrow{\$} X_\lambda : D(1^\lambda, x, z) = 1] - \Pr[y \xleftarrow{\$} Y_\lambda : D(1^\lambda, y, z) = 1]| \leq \frac{1}{S(\lambda)}$$

This definition implies $(\lambda_0, S(\cdot))$-indistinguishability. Consider a distinguisher D that on input $(1^\lambda, x, z)$ runs the universal TM $U(x, z)$, and let λ_U be the constant associated with it. For any $\lambda > \lambda_U$, and every $S(\lambda)$-size circuit C, by setting the auxiliary input $z = C$, the above definition implies that the distinguishing gap by C is at most $1/S(\lambda)$. Therefore, λ_U is effectively the universal constant that works for all (circuit) distinguisher.

Below, we state definitions of cryptographic primitives using $(\lambda_0, S(\cdot))$ indistinguishability. Traditional polynomial or sub-exponential security can be directly derived from such more concrete definitions as follows:

Definition 3 (Polynomial Indistinguishability). *A pair of ensembles $\{X_\lambda\}$, $\{Y_\lambda\}$ are polynomially indistinguishable if for every polynomial $p(\cdot)$, there is a constant $\lambda_p \in N$, such that, the two ensembles are $(\lambda_p, p(\cdot))$-indistinguishable.*

Definition 4 (Sub-exponential Indistinguishability). *A pair of ensembles $\{X_\lambda\}$, $\{Y_\lambda\}$ are sub-exponentially indistinguishable, if there is a sub-exponential function $S(\lambda) = 2^{\lambda^\varepsilon}$ with $\varepsilon \in (0, 1)$ and a constant $\lambda_0 \in N$, such that, the two ensembles are $(\lambda_0, S(\cdot))$-indistinguishable.*

2.2 Standard cryptographic primitives

Definition 5 (Pseudorandom Generator). *A deterministic PT uniform machine* PRG *is a pseudorandom generator if the following conditions are satisfied:*

Syntax *For every $\lambda, \lambda' \in \mathbb{N}$ and every $r \in \{0,1\}^\lambda$, $\mathsf{PRG}(r, \lambda')$ outputs $r' \in \{0,1\}^{\lambda'}$*

$(\lambda_0, S(\cdot))$**-Security** *For every function $p(\cdot)$, such that, $p(\lambda) \leq S(\lambda)$ for all λ, the following ensembles are $(\lambda_0, S(\cdot))$ indistinguishable*

$$\left\{r \xleftarrow{\$} \{0,1\}^\lambda : \mathsf{PRG}(r, p(\lambda))\right\} \; \left\{r' \xleftarrow{\$} \{0,1\}^{p(\lambda)}\right\}$$

2.3 Indistinguishability Obfuscation

In this section, we recall the definition of indistinguishability obfuscation for Turing machines from [BGI$^+$01,BCP14,ABG$^+$13]. Following [BCP14], we considers two notions of obfuscation for Turing machines. The first definition, called *bounded-input* indistinguishability obfuscation, only requires the obfuscated program to work for inputs of *bounded length* and furthermore the size of the obfuscated program may depend polynomially on this input length bound. (This is the notion achieved in [BGL$^+$15,CHJV14,KLW14] assuming subexponentially-secure **iO** for circuits and one-way functions.)

The second notion considered in [BCP14] is stronger and requires the obfuscated program to work on any arbitrary polynomial length input (and the size of the obfuscated machine thus only depends on the program size and security parameter). We refer to this notion as *unbounded-input* indistinguishability obfuscation. (This stronger notion of unbounded-input indistinguishability obfuscator for Turing machines is only known to be achievable based on strong "extractability assumptions"—namely, (public-coin) differing-input obfuscation for circuits and (public-coin) SNARKs [BCP14,ABG$^+$13,IPS15], variants of which are known to be implausible [BCPR13,GGHW13,BP15]).

Definition 6 (Indistinguishability Obfuscator ($i\mathcal{O}$) for a class of Turing machines). *An indistinguishability obfuscator for a class of Turing machines $\{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is a uniform machine that behaves as follows:*

$\hat{\Pi} \leftarrow i\mathcal{O}(1^\lambda, \Pi, T)$*: $i\mathcal{O}$ takes as input a security parameter 1^λ, the Turing machine to obfuscate $\Pi \in \mathcal{M}_\lambda$ and a time bound T for Π. It outputs a Turing machine $\hat{\Pi}$.*

We require the following conditions to hold.

Correctness: *For every $\lambda \in N$, $\Pi_\lambda \in \mathcal{M}_\lambda$, input x_λ and time bound T_λ,*

$$\Pr[(\tilde{\Pi} \xleftarrow{\$} i\mathcal{O}(1^\lambda, \Pi_\lambda, T_\lambda) : \ \tilde{\Pi}(x_\lambda) = \Pi^T(x_\lambda)] = 1 \ .$$

Efficiency: *The running times of $i\mathcal{O}$ and $\hat{\Pi}$ are bounded as follows:*
There exists polynomial p such that for every security parameter λ, Turing machine $\Pi \in \mathcal{M}_\lambda$, time bound T and every obfuscated machine $\hat{\Pi} \leftarrow i\mathcal{O}(1^\lambda, \Pi, T)$ and input x, we have that

$$\mathsf{Time}_{i\mathcal{O}}(1^\lambda, \Pi, T) \leq p(\lambda, |\Pi|, \log T)$$
$$\mathsf{Time}_{\hat{\Pi}}(x) \leq p(\lambda, |\Pi|, |x|, T)$$

$(\lambda_0, S(\cdot))$**-Security:** *For every ensemble of pairs of Turing machines and time bounds $\{\Pi_{0,\lambda}, \Pi_{1,\lambda}, T_\lambda\}$ where for every $\lambda \in \mathbb{N}$, $\Pi_0 = \Pi_{0,\lambda}$, $\Pi_1 = \Pi_{1,\lambda}$, $T = T_\lambda$, satisfying the following*

$$\Pi_0, \Pi_1 \in \mathcal{M}_\lambda \quad |\Pi_0| = |\Pi_1| \leq \mathrm{poly}(\lambda) \quad T \leq \mathrm{poly}(\lambda)$$
$$\forall x, \Pi_0^T(x) = \Pi_1^T(x) \ ,$$

the following ensembles are $(\lambda_0, S(\cdot))$-indistinguishable

$$\left\{i\mathcal{O}(1^\lambda, \Pi_{0,\lambda}, T_\lambda)\right\} \ \left\{i\mathcal{O}(1^\lambda, \Pi_{1,\lambda}, T_\lambda)\right\} \ .$$

Definition 7 (Unbounded-input indistinguishability obfuscator for Turing machines). *An unbounded-input indistinguishability obfuscator for Turing machines $i\mathcal{O}(\cdot,\cdot,\cdot)$ is simply an indistinguishability obfuscator for the class of all Boolean Turing machines.*

Remark 1 (Obfuscation for Boolean Turing machines is without loss of generality) *The above definition is equivalent to one that considers the class of all Turing machines. Any Turing machine with output length m can be represented as a Boolean Turing machine that takes in an additional input $i \in [m]$ and returns the i^{th} bit of the m-bit long output.*

Definition 8 (Bounded-input indistinguishability obfuscator for Turing machines). *A bounded-input indistinguishability obfuscator for Turing machines $i\mathcal{O}(\cdot,\cdot,\cdot,\cdot)$ is a uniform machine such that for every polynomial p, $i\mathcal{O}(p,\cdot,\cdot,\cdot)$ is an indistinguishability obfuscator for the class of Turing machines $\{\mathcal{M}_\lambda\}$ where $\mathcal{M}_\lambda$ are machines that accept only inputs of length $p(\lambda)$. Additionally, $i\mathcal{O}(p, 1^\lambda, \Pi, T)$ is allowed to run in time* $\mathrm{poly}(p(\lambda) + \lambda + |\Pi| + \log T)$.

2.4 Functional Encryption

Definition 9 (Selectively-secure Single-Query Public-key Functional Encryption). *A tuple of PPT algorithms* $(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a selectively-secure functional encryption scheme for a class of circuits* $\{\mathcal{C}_\lambda\}$ *if it satisfies the following properties.*

Completeness *For every* $\lambda \in \mathbb{N}$, $C \in \mathcal{C}_\lambda$ *and message* $m \in \{0,1\}^*$,

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

$(\lambda_0, S(\cdot))$**-Selective-security** *For every ensemble of circuits and pair of messages* $\{C_\lambda, m_{0,\lambda}, m_{1,\lambda}\}$ *where* $C_\lambda \in \mathcal{C}_\lambda$, $|C_\lambda|, |m_{0,\lambda}|, |m_{1,\lambda}| \leq \mathrm{poly}(\lambda)$, *and* $C_\lambda(m_{0,\lambda}) = C_\lambda(m_{1,\lambda})$, *the following ensembles of distributions* $\{D_{0,\lambda}\}$ *and* $\{D_{1,\lambda}\}$ *are* $(\lambda_0, S(\cdot))$*-indistinguishable.*

$$D_{b,\lambda} = \left(\begin{array}{c} (mpk, msk) \leftarrow \mathsf{FE.Setup}(1^\lambda) \\ c \leftarrow \mathsf{FE.Enc}(1^\lambda, m_{b,\lambda}) \\ sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C_\lambda) \end{array} : mpk, c, sk_C\right)$$

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.

Definition 10 (Compact Functional Encryption). *We say a functional encryption scheme is* compact *if it additionally satisfies the following requirement:*

Compactness *The running time of* $\mathsf{FE.Enc}$ *is bounded as follows.*

There exists a polynomial p *such that for every security parameter* $\lambda \in \mathbb{N}$ *and message* $m \in \{0,1\}^*$, $\mathsf{Time}_{\mathsf{FE.Enc}}(1^\lambda, m) \leq p(\lambda, |m|, polylog(s))$, *where* $s = \max_{C \in \mathcal{C}_\lambda} |C|$.

Furthermore, we say the functional encryption scheme has sub-linear compactness *if there exists a polynomial* p *and constant* $\epsilon > 0$ *such that for every security parameter* $\lambda \in \mathbb{N}$ *and message* $m \in \{0,1\}^*$, $\mathsf{Time}_{\mathsf{FE.Enc}}(1^\lambda, m) \leq p(\lambda, |m|)s^{1-\epsilon}$.

We also define a notion of succinctness, as follows:

Definition 11 (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 6 ([GKP$^+$13b]). *Assuming (sub-exponentially secure) LWE, there exists a (sub-exponentially secure) succinct functional encryption scheme for* NC^1.

We note that [GKP$^+$13b] do not explicitly consider sub-exponentially secure succinct functional encryption, but their construction satisfies it (assuming sub-exponentially secure LWE).

Theorem 7 ([GKP$^+$13b,ABSV14]). *Assuming the existence of symmetric-key encryption with decryption in* NC^1 *(resp. sub-exponentially secure) and succinct FE for* NC^1 *(resp. sub-exponentially secure), there exists succinct FE for* $\mathsf{P/poly}$ *(resp. sub-exponentially secure).*

We also consider an even weaker notion of sublinear-compactness, where only the ciphertext size is sublinear in the size bound s of the function being evaluation, but the encryption time can depend polynomially on s.

Definition 12 (Weakly Sublinear Compact Functional Encryption). *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 $s = \max_{C \in \mathcal{C}_\lambda} |C|$.

3 Randomized Encoding Schemes

Roughly speaking, randomized encoding schemes encodes a computation of a program Π on an input x, into an encoded computation $(\hat{\Pi}, \hat{x})$, with the following two properties: First, the encoded computation evaluates to the same output $\Pi(x)$, while leaking no other information about Π and x. Second, the encoding is "simpler" to compute than the original computation. In the literature, different measures of simplicity have been considered. For instance, in the original works by [IK02a,AIK06], the depth of computation is used and it was shown that any computation in P can be encoded in NC_1 using Yao's garbled circuits [Yao82]. A recent line of works [BGL$^+$15,CHJV14,KLW14] uses the time-complexity as the measure and show that any *Boolean* Turing machine computation can be encoded in time poly-logarithmic in the run-time of the computation.

Traditionally, the security of randomized encoding schemes are capture via simulation. In this work, we consider a new *distributional* indistinguishability-based security notion, and show that it is implied by the transitional simulation security. Additionally, we further explore how compact the encoded computation can be: Similar to the recent works [BGL$^+$15,CHJV14,KLW14], we consider encoding whose size depends poly-logarithmically on the run-time of the encoded computation; but differently, we directly consider Turing machines with arbitrary length outputs, and require the size of the encoding to be independent of the output length. Such scheme is called a compact randomized encoding scheme.

3.1 Distributional Indistinguishability Security

In this paper, we study randomized encoding for all Turing machine computation, whose encoding size is independent of the output length of the computation—we say such randomized encoding schemes are **compact**. Towards this, we must consider weaker security notions than simulation security, and indistinguishability-based security notions are natural candidates. One weaker notion that has been considered in the literature requires encoding of two computation, (Π_1, x_1) and (Π_2, x_2) with the same output $\Pi_1(x_1) = \Pi_2(x_2)$, to be indistinguishable. In this work, we generalize this notion to, what called *distributional* indistinguishability security—this notion requires encoding of computations sampled from two distributions, $(\Pi_1, x_1) \xleftarrow{\$} D_1$ and $(\Pi_2, x_2) \xleftarrow{\$} D_2$, to be indistinguishable, provided that their outputs are indistinguishable.

Definition 13 (Randomized Encoding Scheme for a Class of Turing Machines). *A Randomized Encoding scheme* RE *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *consists of two algorithms,*

- $(\hat{\Pi}, \hat{x}) \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \Pi, x, T)$*: On input a security parameter* 1^λ*, Turing machine* $\Pi \in \mathcal{M}_\lambda$*, input* x *and time bound* T*,* Enc *generates an encoded machine* $\hat{\Pi}$ *and encoded input* $\hat{x}$*.*
- $y = \mathsf{Eval}(\hat{\Pi}, \hat{x})$*: On input* $(\hat{\Pi}, \hat{x})$ *produced by* $\mathsf{Enc}, \mathsf{Eval}$ *outputs* y*.*

Correctness: *The two algorithms* Enc *and* Eval *satisfy the following correctness condition: For all security parameters* $\lambda \in \mathbb{N}$*, Turing machines* $\Pi \in \mathcal{M}_\lambda$*, inputs* x *and time bounds* T*, it holds that,*

$$\Pr[(\hat{\Pi}, \hat{x}) \xleftarrow{\$} \mathsf{Enc}(1^\lambda, \Pi, x, T) : \ \mathsf{Eval}(\hat{\Pi}, \hat{x}) = \Pi^T(x))] = 1$$

Definition 14 (Distributional $(\lambda_0, S(\cdot))$-Indistinguishability Security). *A randomized encoding scheme* RE *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *satisfies* distributional $(\lambda_0, S(\cdot))$-indistinguishability security*, (or* $(\lambda_0, S(\cdot))$*-ind-security for short) if the following is true w.r.t. some constant* $c > 0$*:*

For every ensembles of distributions $\{D_{0,\lambda}\}$ *and* $\{D_{1,\lambda}\}$ *with the following property:*

1. *there exists a polynomial* B*, such that, for every* $b \in \{0, 1\}$*,* $D_{b,\lambda}$ *is a distribution over tuples of the form* (Π_b, x_b, T_b)*, where* Π_b *is a Turing machine,* x_b *is an input and* T_b *is a time bound, and* $\lambda, |\Pi_b|, |x_b|, T_b \leq B(\lambda)$*.*
2. *there exist an integer* $\lambda'_0 \geq \lambda_0$*, and a function* S' *with* $S'(\lambda) \leq S(\lambda)$ *for all* λ*, such that, the following ensembles of output distributions are* $(\lambda'_0, S'(\cdot))$*-indistinguishable,*

$$\left\{(\Pi_0, x_0, T_0) \xleftarrow{\$} \mathcal{D}_{0,\lambda} : \Pi_0^{T_0}(x_0), T_0, |\Pi_0|, |x_0|\right\}$$
$$\left\{(\Pi_1, x_1, T_1) \xleftarrow{\$} \mathcal{D}_{1,\lambda} : \Pi_1^{T_1}(x_1), T_1, |\Pi_1|, |x_1|\right\}$$

the following ensembles of encoding is $(\lambda'_0, S''(\cdot))$*-indistinguishable, where* $S''(\lambda) = \frac{S'(\lambda)}{\lambda^c} - B(\lambda)^c$*.*

$$\left\{(\Pi_0, x_0, T_0) \xleftarrow{\$} \mathcal{D}_{0,\lambda} : \mathsf{Enc}(1^\lambda, \Pi_0, x_0, T_0)\right\}$$
$$\left\{(\Pi_1, x_1, T_1) \xleftarrow{\$} \mathcal{D}_{1,\lambda} : \mathsf{Enc}(1^\lambda, \Pi_1, x_1, T_1)\right\}$$

For convenience, in the rest of the paper, we directly refer to distributional indistinguishability security as indistinguishability security. The above concrete security directly gives the standard polynomial and sub-exponential security.

Definition 15 (Polynomial and Sub-exponential Indistinguishability Security). *A randomized encoding scheme* RE *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *satisfies* polynomial ind-security *, if it satisfies* $(\lambda_p, p(\cdot))$*-indistinguishability security for every polynomial* p *and some* $\lambda_p \in N$*. Furthermore, it satisfies sub-exponential ind-security if it satisfies* $(\lambda_0, S(\cdot))$*-indistinguishability security for* $S(\lambda) = 2^{\lambda^\varepsilon}$ *with some* $\varepsilon \in (0,1)$*.*

We note that, by definition, it holds that any randomized encoding scheme that is $(\lambda_0, S(\cdot))$-ind-secure, is also $(\lambda'_0, S'(\cdot))$-ind-secure for any $\lambda'_0 \geq \lambda_0$ and S' s.t. $S'(\lambda) \leq S(\lambda)$ for every λ. Therefore, naturally, sub-exponential ind-security is stronger than polynomial ind-security.

In the full version, we show that RE schemes with ind-security are composable just as RE schemes with simulation security are.

3.2 Compactness and Sublinear Compactness

With indistinguishability-security, we now define compact randomized encoding schemes for all Turing machines, whose time-complexity of encoding is independent of the output length.

Definition 16 (Compact Randomized Encoding for Turing machines). *A* $(\lambda_0, S(\cdot))$*-ind-secure* compact randomized encoding *scheme for Turing machines, is a randomized encoding scheme with* $(\lambda_0, S(\cdot))$*-indistinguishability security for the class of all Turing machines, with the following efficiency:*

- *For every security parameter* λ*, Turing machine* Π*, input* x*, time bound* T *and every encoded pair* $(\hat{\Pi}, \hat{x}) \leftarrow \mathsf{Enc}(1^\lambda, \Pi, x, T)$*, it holds*

$$\mathsf{Time}_{\mathsf{Enc}}(1^\lambda, \Pi, x, T) = \mathrm{poly}(\lambda, |\Pi|, |x|, \log T)$$
$$\mathsf{Time}_{\mathsf{Eval}}(\hat{\Pi}, \hat{x}) = \mathrm{poly}(\lambda, |\Pi|, |x|, T)$$

In this work, we also consider a weaker variant of the above compactness requirement, where the encoding time is sub-linear (instead of poly-logarithmic) in the computation time. For our results a compact randomized encoding scheme with sub-linear efficiency will suffice.

Definition 17 (Sub-linear Compactness of Randomized Encoding schemes). *We say a randomized encoding scheme* $\mathsf{RE} = (\mathsf{Enc}, \mathsf{Eval})$ *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *has sub-linear compactness if the efficiency requirement on* Enc *in Definition 16 is relaxed to: For some constant* $\varepsilon \in (0,1)$*,*

$$\mathsf{Time}_{\mathsf{Enc}}(1^\lambda, \Pi, x, T) \leq \mathrm{poly}(\lambda, |\Pi|, |x|) \cdot T^{1-\epsilon}$$

4 Unbounded-Input IO from Compact RE

In this section, we define our succinct indistinguishability obfuscator for Turing machines. Let $\mathsf{RE} = (\mathsf{Enc}, \mathsf{Eval})$ be a compact randomized encoding scheme for Turing machines with sub-exponential indistinguishability security. Let c be the constant for the security loss associated with the indistinguishability security of RE. We assume without loss of generality that $\mathsf{Enc}(1^\lambda, \cdot, \cdot)$ requires a random tape of length λ. Let PRG be a sub-exponentially secure pseudorandom generator and let ϵ be the constant associated with the sub-exponential security of PRG.

For every $\lambda \in \mathbb{N}$, $D \leq 2^\lambda$, define

$$l(\lambda, -1) = \lambda$$

$$l(\lambda, D) = l(\lambda, D-1) + (2d\lambda)^{1/\epsilon}$$

where $d > 0$ is any constant strictly greater than c.

Construction 1 *Consider a Turing machine $\mathit{\Pi}$, security parameter $\lambda \in \mathbb{N}$, and time bound T of $\mathit{\Pi}$. For every partial input $s \in \{0,1\}^*$ with $|s| \leq 2^\lambda$ and $R \in \{0,1\}^{2l(\lambda,|s|)}$, we recursively define a Turing machine $\widetilde{\mathit{\Pi}}_{s,R}$ to be as follows:*

When $|s| < 2^\lambda$:

On the empty input, $\widetilde{\mathit{\Pi}}_{s,R}$ outputs:

$$\mathsf{Enc}(1^{l(\lambda,|s|+1)}, \widetilde{\mathit{\Pi}}_{s0,R_0}, T'(\lambda, |s|+1, |\mathit{\Pi}|, \log(T)); R_1)$$
$$\mathsf{Enc}(1^{l(\lambda,|s|+2)}, \widetilde{\mathit{\Pi}}_{s1,R_2}, T'(\lambda, |s|+1, |\mathit{\Pi}|, \log(T)); R_3)$$
$$\mathsf{Enc}(1^{l(\lambda,|s|+1)}, \mathit{\Pi}, s, T; R_4)$$

where $(R_0, R_1, R_2, R_3, R_4) \leftarrow \mathsf{PRG}(R, 5 \cdot 2l(\lambda, |s|+1))$ and T' is some fixed polynomial in $\lambda, |s|+1, |\mathit{\Pi}|$ and $\log(T)$. In the special case when $|s| = 2^\lambda - 1$, the time bound used in the first two encodings is set to T.

On all other inputs, $\widetilde{\mathit{\Pi}}_{s,R}$ outputs $\perp$.

When $|s| = 2^\lambda$:

On the empty input, $\widetilde{\mathit{\Pi}}_{s,R}$ outputs $\mathsf{Enc}(1^{l(\lambda,|s|+1)}, \mathit{\Pi}, s, T; R)$. On all other inputs, $\widetilde{\mathit{\Pi}}_{s,R}$ outputs $\perp$.

We define $T'(\cdot,\cdot,\cdot,\cdot)$ (corresponding to the bound placed on the running time of $\widetilde{\mathit{\Pi}}_{s,R}$) to be the smallest polynomial such that for all λ, $s \in \{0,1\}^{\leq 2^\lambda}$, $R \in \{0,1\}^{2l(\lambda,|s|)}$, $\mathit{\Pi}$ and T,

$$\begin{aligned} T'(\lambda, |s|, |\mathit{\Pi}|, \log(T)) \geq\ & p(\lambda_{|s|+1}, |\widetilde{\mathit{\Pi}}_{s0,R}|, 0, \log(T'_{|s|+1})) \\ & + p(\lambda_{|s|+1}, |\widetilde{\mathit{\Pi}}_{s1,R}|, 0, \log(T'_{|s|+1})) \\ & + p(\lambda_{|s|+1}, |\mathit{\Pi}|, |s|, \log(T)) \\ & + \mathsf{Time}_{\mathsf{PRG}}(R, 5 \cdot 2l(\lambda, |s|+1)) \end{aligned}$$

where $\lambda_{|s|+1} = l(\lambda, |s|+1)$*,* $T'_{|s|+1} = T'(\lambda, |s|+1, |\Pi|, \log(T))$ *(corresponding to the security parameter and time bound used for each of* $\widetilde{\Pi}_{s0,R_0}$ *and* $\widetilde{\Pi}_{s1,R_1}$*),* $\mathsf{Time}_{\mathsf{PRG}}$ *is the bound on the running time of the PRG, and* $p(\cdot,\cdot,\cdot,\cdot)$ *is the bound on* $\mathsf{Time}_{\mathsf{Enc}}$ *from the compactness of* RE*. We note that the polynomial* T' *exists because* p *is a polynomial, each of* $\lambda_{|s|+1}$ *and* $|\widetilde{\Pi}_{s,R}|$ *are of size polynomial in* $\lambda, |s|$ *and* $|\Pi|$*, and the self-dependence of* $T'(\lambda, |s|, |\Pi|, \log(T))$ *on* $T'_{|s|+1}$ *is only poly-logarithmic.*

Remark: *We note that* $|\widetilde{\Pi}_{s,R}|$ *is always* $poly(\lambda, |\Pi|, |s|, \log(T))$*. This is because* $\widetilde{\Pi}_{s,R}$ *is fully described by* λ*,* Π*,* s*,* R *and* T*, and the size of each of these is bounded by* $poly(\lambda, |\Pi|, |s|, \log(T))$*.*

Given this definition of $\widetilde{\Pi}_{s,R}$, we define our indistinguishability obfuscator as follows:

Construction 2 (Indistinguishability Obfuscator) *On input* $\lambda \in \mathbb{N}$*, Turing machine* Π *and time bound* T*, define* $\widetilde{\Pi}$*, the indistinguishability obfuscation of* Π*, to be*

$$\widetilde{\Pi} = \boldsymbol{iO}(1^\lambda, \Pi, T) = \mathsf{Enc}(1^{l(\lambda,0)}, \widetilde{\Pi}_{\epsilon,R}, T'(\lambda, 0, |\Pi|, \log(T)))$$

Where ϵ *is the empty string, and* $R \xleftarrow{\$} \{0,1\}^{2l(\lambda,0)}$ *and* T' *a fixed polynomial in* $\lambda, |\Pi|$ *and* $\log(T)$*, as described above.*

Evaluation: The algorithm to evaluate $\widetilde{\Pi}$ on input $x \in \{0,1\}^d, d < 2^\lambda$ proceeds as follows:

1. For every $0 \le i \le d$, compute encodings of $\widetilde{\Pi}_{x_{\le i},R}$ successively, starting with $\widetilde{\Pi}$, an encoding of $\widetilde{\Pi}_{\epsilon,R}$, and subsequently, for every $0 < i \le d$, computing the encoding of $\widetilde{\Pi}_{x_{\le i},R}$ by evaluating the encoding of $\widetilde{\Pi}_{x_{<i},R}$, and selecting the encoding of $\widetilde{\Pi}_{x_{\le i},R}$ from its output.
2. Evaluate the encoding of $\widetilde{\Pi}_{x,R} = \widetilde{\Pi}_{x_{\le d},R}$ and obtain from its output $(\hat{\Pi}, \hat{x}) = \mathsf{Enc}(1^{l(\lambda,|x|+1)}, \Pi, x, T; R_4)$.
3. Run $\mathsf{Eval}(\hat{\Pi}, \hat{x})$ to obtain $\Pi(x)$.

We defer analysis of the correctness, running time, and compactness of our **iO** construction to the full version of our paper [LPST15]

4.1 Security Proof

Theorem 8. *Let* $(\mathsf{Enc}, \mathsf{Eval})$ *be a sub-exponentially-indistinguishability-secure, compact randomized encoding scheme and let* PRG *be a sub-exponentially-secure pseudorandom generator. Then the indistinguishability obfuscator defined in Construction 2 is subexponentially-secure.*

Proof. Consider any pair of ensembles of Turing machines and time bounds $\left\{\Pi^0_\lambda, \Pi^1_\lambda, T_\lambda\right\}$ where for every $\lambda \in \mathbb{N}$, $\Pi^0 = \Pi^0_\lambda$, $\Pi^1 = \Pi^1_\lambda$, $T = T_\lambda$,

$$|\Pi^0| = |\Pi^1| \leq \text{poly}(\lambda) \quad |T| \leq \text{poly}(\lambda)$$
$$\forall x, \Pi^{0,T}(x) = \Pi^{1,T}(x)$$

We first introduce some notation to describe the distributions of randomized encodings generated by $i\mathcal{O}(1^\lambda, \Pi^0_\lambda, T_\lambda)$ and $i\mathcal{O}(1^\lambda, \Pi^1_\lambda, T_\lambda)$. For $\lambda \in \mathbb{N}$, $s \in \{0,1\}^*, |s| \leq 2^\lambda$, we define the following distributions

$$D_{\lambda,0,s} = \mathsf{Enc}(1^{l(\lambda,|s|)}, \widetilde{\Pi}^0_{s,R}, T')$$
$$D_{\lambda,1,s} = \mathsf{Enc}(1^{l(\lambda,|s|)}, \widetilde{\Pi}^1_{s,R}, T')$$

where R is uniformly random, T' is as described in Construction 1 and $\widetilde{\Pi}^b_{s,R}$ is defined for the Turing machine Π^b_λ, security parameter λ and time bound T_λ. We will show something stronger than the theorem statement. In particular, we have the following claim.

Claim. There exists $\lambda_0, \epsilon \in \mathbb{N}$ such that for every $\lambda > \lambda_0$, for every $s \in \{0,1\}^*, |s| \leq 2^\lambda$ we have that the distributions $D_{\lambda,0,s}$ and $D_{\lambda,1,s}$ are $S(\lambda)$ indistinguishable where $S(\lambda) \geq 10 \cdot 2^{l(\lambda,|s|-1)^\epsilon}$.

Using the above claim with s as the empty string and recalling $l(\lambda, 0) = \lambda$, the theorem statement follows. Therefore, in the remainder of the proof, we prove the above claim.

PROOF OF CLAIM Let ϵ be the larger of the constants associated with the subexponential security of the pseudorandom generator PRG and the indistinguishability security of the encoding scheme $(\mathsf{Enc}, \mathsf{Eval})$ (these constants are also named ϵ in their respective security definitions). Similarly, We consider λ_0 to be large enough so that the security of the encoding scheme $(\mathsf{Enc}, \mathsf{Eval})$ and the pseudorandom generator PRG is applicable. We will actually require a larger λ_0 so that certain asymptotic conditions (depending only on the polynomial size bounds of Π^0_λ, Π^1_λ and T_λ) hold, which we make explicit in the remainder of the proof. For every $\lambda > \lambda_0$, we prove the claim by induction on $|s|$. Our base case will be when $|s| = 2^\lambda$ and in the inductive step we show the claim holds for all s of a particular length d, if it holds for all s of length $d+1$.

INDUCTION STATEMENT, FOR A FIXED $\lambda > \lambda_0$: For every $s \in \{0,1\}^{\leq 2^\lambda}$, the distributions $D_{\lambda,0,s}$ and $D_{\lambda,1,s}$ are $10 \cdot 2^{l(\lambda,|s|-1)^\epsilon}$ indistinguishable.

BASE CASE: $|s| = 2^\lambda$.

In this case, recall that the output of $\widetilde{\Pi}^b_{s,R}$ is simply $(\hat{\Pi}^{b,T}_\lambda, \hat{s})$. We first claim that, for all s, $(\hat{\Pi}^{0,T}_\lambda \hat{s})$ and $(\hat{\Pi}^{1,T}_\lambda, \hat{s})$ are $2^{\lambda'^\epsilon}$ indistinguishable where $\lambda' = l(\lambda, |s|)$, as follows.

Recall that the output of evaluating $\hat{\Pi}^{b,T}_\lambda, \hat{s}$ is simply $\Pi^{b,T}_\lambda(s)$. Since we have that $\Pi^{0,T}_\lambda(s) = \Pi^{1,T}_\lambda(s)$ for all s, we can apply the security of the randomized encoding scheme. More concretely, since the output (point) distributions are identical, they are $10 \cdot 2^{\lambda'^{\epsilon}}$-indistinguishable where $\lambda' = l(\lambda, |s|+1)$. Let $B(\cdot)$ be a polynomial such that $B(\lambda')$ bounds from above $|\mathit{\Pi}^b|, |s|$ and T. By the security of the encoding scheme, the encodings $(\hat{\mathit{\Pi}}_\lambda^{0,T} \hat{s})$ and $(\hat{\mathit{\Pi}}_\lambda^{1,T} \hat{s})$ are S' indistinguishable where

$$S' \geq \frac{10 \cdot 2^{l(\lambda,|s|+1)^{\epsilon}}}{l(\lambda,|s|+1)^c} - B(l(\lambda,|s|+1))^c \geq \frac{10 \cdot 2^{l(\lambda,|s|+1)^{\epsilon}}}{l(\lambda,|s|+1)^d} \geq 10 \cdot 2^{l(\lambda,|s|)^{\epsilon}}$$

where the first inequality holds for sufficiently large λ and in the second inequality, we use the fact that $l(\lambda,|s|+1) = l(\lambda,|s|) + \lambda^{d/\epsilon}$. Thus $(\hat{\mathit{\Pi}}_\lambda^{0,T}, \hat{s})$ and $(\hat{\mathit{\Pi}}_\lambda^{1,T}, \hat{s})$ are $10 \cdot 2^{l(\lambda,|s_\lambda|)^{\epsilon}}$-indistinguishable.

Now, recall that the output of $\widetilde{\mathit{\Pi}}_{s,R}^b$ is simply $(\hat{\mathit{\Pi}}_\lambda^{b,T}, \hat{s})$. By the above argument, we have that, for all s, $(\hat{\mathit{\Pi}}_\lambda^{0,T} \hat{s})$ and $(\hat{\mathit{\Pi}}_\lambda^{1,T}, \hat{s})$ are $2^{\lambda'^{\epsilon}}$-indistinguishable where $\lambda' = l(\lambda, |s|)$. Let B' be the polynomial such that $B'(l(\lambda,|s|))$ bounds $|\widetilde{\mathit{\Pi}}_{s,R}^b|$ and the running time of $\widetilde{\mathit{\Pi}}_{s,R}^b$. The encodings $D_{\lambda,0,s}$ and $D_{\lambda,1,s}$ are S' indistinguishable where

$$S' \geq \frac{10 \cdot 2^{l(\lambda,|s|)^{\epsilon}}}{l(\lambda,|s|)^c} - B'(l(\lambda,|s|))^c \geq \frac{10 \cdot 2^{l(\lambda,|s|+1)^{\epsilon}}}{l(\lambda,|s|)^d} \geq 10 \cdot 2^{l(\lambda,|s|-1)^{\epsilon}}$$

where, as before, the first inequality holds for sufficiently large λ and in the second inequality, we use the fact that $l(\lambda,|s|+1) = l(\lambda,|s|) + \lambda^{d/\epsilon}$. Hence the claim holds for $|s| = 2^\lambda$.

INDUCTIVE STEP: $|s| < 2^\lambda$. By the induction hypothesis, we assume the claim holds for all s' such that $|s'| = |s|+1$. Recall that the output of $\widetilde{\mathit{\Pi}}_{s,R}^b$ (where $R \xleftarrow{\$} \{0,1\}^{2l(\lambda,|s|)}$) is

$$\begin{gathered}\mathsf{Enc}(1^{l(\lambda,|s|+1)}, \widetilde{\mathit{\Pi}}_{s0,R_0}^b, T'; R_1)\\ \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \widetilde{\mathit{\Pi}}_{s1,R_2}^b, T'; R_3)\\ \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \mathit{\Pi}_\lambda^b, s, T; R_4)\end{gathered}$$

where $(R_0, R_1, R_2, R_3, R_4) \leftarrow \mathsf{PRG}(R, 5 \cdot 2l(\lambda,|s|+1))$. Let H^b denote the above output distribution. We will show H^0 and H^1 are indistinguishable by a hybrid argument as follows.

- Let G_1 be a hybrid distribution exactly as H^0 except that $(R_0, R_1, R_2, R_3, R_4) \xleftarrow{\$} \{0,1\}^{5 \cdot 2l(\lambda,|s|+1)}$. We claim that for both the distributions H^0 and G_1 are $5 \cdot 2^{\lambda'^{\epsilon}}$ indistinguishable where $\lambda' = l(\lambda, |s|)$.
  This follows from the PRG security as follows: any size $5 \cdot 2^{\lambda'^{\epsilon}}$ adversary A that distinguishes H^0 and G_1 can be turned into an adversary A' that can break the PRG security with seed length $2\lambda'$ with the same advantage. A' has $\mathit{\Pi}_\lambda^0$, $\mathit{\Pi}_\lambda^1$, T_λ and s hardcoded in it. Hence, the size of A' is

$$5 \cdot 2^{\lambda'^{\epsilon}} + \mathrm{poly}(\lambda) + \mathrm{poly}(|s|) \leq 5 \cdot 2^{\lambda'^{\epsilon}} + \mathrm{poly}(\lambda') \leq 2^{(2\lambda')^{\epsilon}}$$

where the last inequality holds when λ is sufficiently large. Hence, A' breaks the $2^{(2\lambda')^{\epsilon}}$-security of PRG and we have a contradiction.

Writing out the components of G_1, we have that it is identical to

$$G_1 \equiv D_{\lambda,0,s0}, D_{\lambda,0,s1}, \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \Pi_{\lambda}^{0}, s, T_{\lambda}; R)$$

- Let G_2 be a hybrid distribution obtained by modifying the first component of G_1 as follows.

  $$G_2 \equiv D_{\lambda,1,s0}, D_{\lambda,0,s1}, \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \Pi_{\lambda}^{0}, s, T_{\lambda}; R)$$

  We show that G_1 and G_2 are $5 \cdot 2^{{\lambda'}^{\epsilon}}$ indistinguishable. This follows from the induction hypothesis as follows: any size $5 \cdot 2^{{\lambda'}^{\epsilon}}$ adversary A that distinguishes G_1 and G_2 with advantage better than $1/(5 \cdot 2^{{\lambda'}^{\epsilon}})$ can be turned into an adversary A' that can distinguish $D_{\lambda,0,s0}$ and $D_{\lambda,1,s0}$ with the same advantage. As before, A' has Π_{λ}^{0}, Π_{λ}^{1}, T_{λ} and s hardcoded in it, and therefore the size of A' is at most $5 \cdot 2^{{\lambda'}^{\epsilon}} + \mathrm{poly}(\lambda') \leq 10 \cdot 2^{{\lambda'}^{\epsilon}}$. Hence, A' breaks the induction hypothesis that says $D_{\lambda,0,s0}$ and $D_{\lambda,1,s0}$ are $10 \cdot 2^{{\lambda'}^{\epsilon}}$-indistinguishable.
- Similarly, let G_3 be a hybrid distribution obtained by modifying the second component of G_2 as follows.

  $$G_3 \equiv D_{\lambda,1,s0}, D_{\lambda,1,s1}, \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \Pi_{\lambda}^{0}, s, T_{\lambda}; R)$$

  Similarly as above, we have that G_2 and G_3 are $5 \cdot 2^{{\lambda'}^{\epsilon}}$-indistinguishable.
- Let G_4 be a hybrid distribution obtained by modifying the third component of G_3 as follows.

  $$G_4 \equiv D_{\lambda,1,s0}, D_{\lambda,1,s1}, \mathsf{Enc}(1^{l(\lambda,|s|+1)}, \Pi_{\lambda}^{1}, s, T_{\lambda}; R)$$

  We show G_3 and G_4 are $5 \cdot 2^{{\lambda'}^{\epsilon}}$-indistinguishable. First, since $\Pi_{\lambda}^{0,T}(s) = \Pi_{\lambda}^{1,T}(s)$, by the security of the encoding scheme, we have that the encodings that form the third component of G_3 and G_4 are S' indistinguishable where, similar to the base case, $B(l(\lambda,|s|))$ bounds from above $|\Pi_{\lambda}^{b}|$, $|s|$ and T

  $$S' \geq \frac{10 \cdot 2^{l(\lambda,|s|)^{\epsilon}}}{l(\lambda,|s|)^{c}} - B(l(\lambda,|s|))^{c} \geq \frac{10 \cdot 2^{l(\lambda,|s|)^{\epsilon}}}{l(\lambda,|s|)^{d}} \geq 10 \cdot 2^{l(\lambda,|s|-1)^{\epsilon}}$$

  Hence by a similar argument as before, the hybrid distributions are $5 \cdot 2^{{\lambda'}^{\epsilon}}$-indistinguishable.
- Finally we observe that G_4 and H^1 are $5 \cdot 2^{{\lambda'}^{\epsilon}}$-indistinguishable just as G_1 and H^0 were. By a simple hybrid argument, we have that H^0 and H^1 are $2^{{\lambda'}^{\epsilon}}$-indistinguishable.

Recall that H^0 and H^1 are the distributions of outputs of $\widetilde{\Pi}_{s,R}^{0}$ and $\widetilde{\Pi}_{s,R}^{1}$ respectively. By the security of the randomized encoding scheme, the encodings of these machines, *i.e.* $D_{\lambda,0,s}$ and $D_{\lambda,1,s}$ are $S'(\lambda)$-indistinguishable where

$$S'(\lambda) \geq \frac{2^{l(\lambda,|s|)^{\epsilon}}}{l(\lambda,|s|)^{c}} - B'(l(\lambda,|s|)^{c} \geq \frac{2^{l(\lambda,|s|)^{\epsilon}}}{l(\lambda,|s|)^{d}} \geq \frac{2^{l(\lambda,|s|-1)^{\epsilon}} \cdot 2^{(2d\lambda)}}{2^{d\lambda} \cdot (2d\lambda)^{d/\epsilon}} \geq 10 \cdot 2^{l(\lambda,|s|-1)^{\epsilon}}$$

where $B'(l(\lambda, |s|))$ bounds from above $|\Pi^b_{s,R}|$ and T'. The second inequality holds for sufficiently large λ. In the third inequality, we use the fact that $l(\lambda, |s|) \leq |s|(2d\lambda)^{1/\epsilon} \leq 2^\lambda(2d\lambda)^{1/\epsilon}$ and the last inequality holds for sufficiently large λ.

4.2 Nice Distributions

Later in Section 6, we show that compact RE does not exist for general distributions in the plain model. However, here we observe that the above construction of unbounded input IO relies only on compact RE for certain "special purpose" distributions that is not ruled out by the impossibility result in Section 6. We now abstract out the structure of these special purpose distributions. Let $\mathsf{RE} = (\mathsf{Enc}, \mathsf{Dec})$ be a randomized encoding scheme; we define "nice" distributions w.r.t. RE.

0-nice distributions: We say that a pair of distribution ensembles $\{\mathcal{D}_{0,\lambda}\}$ and $\{D_{1,\lambda}\}$ are *0-nice* if $D_{0,\lambda}$ always outputs a fixed tuple (Π_0, x, T) while $D_{1,\lambda}$ always outputs a fixed tuple (Π_1, x, T), satisfying that $\Pi_0^T(x) = \Pi_1^T(x)$.

k-nice distributions: We say that a pair of distribution ensembles $\{\mathcal{D}_{0,\lambda}\}$ and $\{\mathcal{D}_{1,\lambda}\}$ are *k-nice* if there exist some $\ell = \text{poly}(\lambda)$ pairs of distributions $(\{\mathcal{E}^i_{0,\lambda}\}, \{\mathcal{E}^i_{1,\lambda}\})_{i\in[\ell]}$, where the i^{th} pair is k^i-nice with $k^i \leq k-1$, such that, $\mathcal{D}_{b,\lambda}$ samples tuple (Π_b, x_b, T_b) satisfying the following:

- For each $i \in [\ell]$, sample $(\Lambda^i_b, z^i_b, T^i_b) \xleftarrow{\$} \mathcal{E}^i_{b,\lambda}$.
- The output of $\Pi_b(x_b)$ consists of ℓ randomized encodings, where the i^{th} encoding is in the support of $\mathsf{Enc}(1^{\lambda'}, \Lambda^i_b, z^i_b, T^i_b)$, for some $\lambda' = \text{poly}(\lambda)$.

Finally, we say that a pair of distribution ensembles $\{\mathcal{D}_{0,\lambda}\}$ and $\{\mathcal{D}_{1,\lambda}\}$ are *nice* w.r.t. RE if they are *k-nice* w.r.t. RE for some integer k.

Our construction of unbounded input IO and its analysis in previous sections relies only on compact RE for nice distribution ensembles. Hence we can refine Theorem 8 to the following:

Proposition 1. *Assume the existence of a compact randomized encoding scheme* RE *which is sub-exponentially-indistinguishability-secure for every pair of distribution ensemble that are nice w.r.t.* RE*; assume further the existence of sub-exponentially secure one-way functions. Then, there is an unbounded-input indistinguishability obfuscator for Turing machines.*

We stress again that compact RE for nice distributions is not ruled out by the impossibility result in Section 6. Hence, we obtain unbounded input IO from a new assumption different from the extactability assumptions used in previous work [BCP14,ABG+13,IPS15].

CANDIDATE CONSTRUCTION: Finally, we describe a candidate construction of compact RE for nice distributions using the KLW indistinguishability obfuscator for bounded-input Boolean Turing machines: Given input $(1^\lambda, M, x, T)$, the encoding is an obfuscation, using the KLW scheme, of the program $\Pi_{M,x}$ that on input $i \in [T]$ outputs the i^{th} bit of the output $M^T(x)$. Since $\Pi_{M,x}$ is Boolean, the KLW obfuscator can be applied, and the encoding time is $\text{poly}(\lambda, |M|, |x|, \log T)$ (hence compact). By the security of indistinguishability obfuscation, for any M_1, x_1 and M_2, x_2 with identical outputs, their encodings are indistinguishable, and thus this construction is a weak compact RE. We here consider it also a candidate construction for compact RE with distributional indistinguishability.

Bounded-Input IO from Sublinear RE: We note that relying on a very similar construction as above, a randomized encoding scheme with only sublinear compactness (as opposed to full compactness) can be used to construct a bounded-input indistinguishability obfuscator for Turing machines. We refer the reader to the full version of this paper [LPST15] for more details.

5 Bounded-Input IO from Compact RE in the CRS Model

In this section we consider compact RE schemes for Turing machines in the *common reference string* (CRS) model. We show that (1) such encoding schemes can be constructed from compact functional encryption for circuits, and that (2) such encoding schemes suffice to get IO for circuits, which then by [KLW14] suffices to get bounded-input IO for Turing machines.

5.1 Randomized Encoding Schemes in the CRS model

We first formally define a randomized encoding scheme for a class of Turing machines in the CRS model. In this model, a one-time setup is performed which takes (in addition to the security parameter) a bound on machine size, input length, running time and output length. Only computations that respect these bounds can be encoded using this setup. The setup outputs a *long* CRS (the length is polynomial in the aforementioned bounds) and a *short* public encoding key (which depends only on the security parameter). The public encoding key is used by the encoding algorithm, which produces encodings that are *compact* as before. The CRS is used by the evaluation algorithm.

Definition 18 (Randomized Encoding Schemes in the CRS Model). *A Randomized Encoding scheme* RE *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *in the CRS model consists of the following algorithms:*

- $(\mathsf{crs}, pk) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^m, 1^n, 1^T, 1^l)$*:* Setup *gets as input (in unary) the security parameter λ, a machine size bound m, input length bound n, time bound T and output length bound l.*
- $\hat{\Pi}_x \xleftarrow{\$} \mathsf{Enc}(pk, \Pi, x)$*:* Enc *is probabilistic and gets as input a public key pk generated by* Setup*, Turing machine $\Pi \in \mathcal{M}_\lambda$ and input x. It outputs an encoding* $\hat{\Pi}_x$[8]*.*

[8] Encoding $\hat{\Pi}_x$ can be viewed as the combination of the program encoding $\hat{\Pi}$ and the input encoding $\hat{x}$ of Definition 13

– $y \leftarrow \mathsf{Eval}(\hat{\Pi}_x, \mathsf{crs})$*: On input* $\hat{\Pi}_x$ *produced by* Enc *and* crs *produced by* Setup*,* Eval *outputs* y*.*

Correctness: *For every security parameters* $\lambda \in \mathbb{N}$*,* $m, n, T, l \in \mathbb{N}$*, Turing machine* $\Pi \in \mathcal{M}_\lambda$ *and input* x*, such that,* $|\Pi| \le m$*,* $|x| \le n$*, and* $|\Pi^T(x)| \le l$*, we have that*

$$\Pr\left[\begin{array}{c}(\mathsf{crs}, pk) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^m, 1^n, 1^T, 1^l) \\ \hat{\Pi}_x \xleftarrow{\$} \mathsf{Enc}(pk, \Pi, x)\end{array} : \mathsf{Eval}(\hat{\Pi}_x, \mathsf{crs}) = \Pi^T(x)\right] = 1$$

In the CRS model, it is possible to have a compact RE for all Turing machines with simulation security.

Definition 19. *A randomized encoding scheme* RE *for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *in the CRS model satisfies* $(\lambda_0, S(\cdot))$**-simulation security***, if there exists a PPT algorithm* Sim *and a constant* c*, such that, for every ensemble* $\{\Pi_\lambda, x_\lambda, m_\lambda, n_\lambda, l_\lambda, T_\lambda\}$ *where* $\Pi_\lambda \in \mathcal{M}_\lambda$ *and* $|\Pi_\lambda|, |x_\lambda|, m_\lambda, n_\lambda, l_\lambda, T_\lambda \le B(\lambda)$ *for some polynomial* B*, the following ensembles are* $(\lambda_0, S'(\lambda))$ *indistinguishable, with* $S'(\lambda) = S(\lambda) - B(\lambda)^c$ *for all* $\lambda \in N$*.*

$$\left\{(\mathsf{crs}, \mathsf{pk}) \xleftarrow{\$} \mathsf{Setup}(1^\lambda, 1^m, 1^n, 1^T, 1^l),\ \hat{\Pi}_x \xleftarrow{\$} \mathsf{Enc}(pk, \Pi, x)\ :\ (\mathsf{crs}, pk, \hat{\Pi}_x)\right\}$$

$$\left\{(\mathsf{crs}, pk, \hat{\Pi}_x) \xleftarrow{\$} \mathsf{Sim}(1^\lambda, \Pi^T(x), 1^{|\Pi|}, 1^{|x|}, 1^m, 1^n, 1^T, 1^l)\ :\ (\mathsf{crs}, pk, \hat{\Pi}_x)\right\}$$

where subscripts of security parameter are suppressed.

Definition 20 (Compactness and Sublinear Compactness in the CRS model). *A randomized encoding scheme* RE = (Setup, Enc, Eval) *for Turing machines in the CRS model is* compact *(or sublinear compact) if* Setup *is PPT, and* Enc *and* Eval *have the same efficiency as their counterparts in a compact (or sublinear compact) randomized encoding scheme for Turing machines in the plain model.*

Remark 2 *We note that a distributional-indistinguishability notion of security (analogous to Definition 14) can be defined for randomized encoding schemes in the CRS model. In the full version of this paper [LPST15], we provide this definition and show* (λ_0, S)*-simulation security implies* (λ_0, S)*-indistinguishability security both in the plain model and the CRS model.*

5.2 Succinctness and Weak-Compactness

We also consider a different weakening of compactness, called succinctness [BGL+15], where encoding time can depend linearly on the length of the output (but only polylogarithmically on the time bound T).

Definition 21 (Succinct Randomized Encoding for Turing machines [BGL+15]). *A* succinct randomized encoding *scheme for Turing machines in the CRS model is succinct if it has the following efficiency:*

- *For every security parameters $\lambda \in \mathbb{N}$, $m, n, T, l \in \mathbb{N}$, Turing machine $\Pi \in \mathcal{M}_\lambda$ and input x, such that, $|\Pi| \leq m$, $|x| \leq n$, and $|\Pi^T(x)| \leq l$, every public key $pk \leftarrow \mathsf{Setup}(1^\lambda, 1^m, 1^n, 1^T, 1^l)$ and every encoding $\hat{\Pi}_x \leftarrow \mathsf{Enc}(1^\lambda, \Pi, x, T)$, it holds*

$$\mathsf{Time}_{\mathsf{Setup}}(1^\lambda, 1^m, 1^n, 1^T, 1^l) = \mathrm{poly}(\lambda, m, n, T, l)$$
$$\mathsf{Time}_{\mathsf{Enc}}(pk, \Pi, x) = \ell \cdot \mathrm{poly}(\lambda, |\Pi|, |x|, \log T)$$
$$\mathsf{Time}_{\mathsf{Eval}}(\hat{\Pi}, \hat{x}) = \mathrm{poly}(\lambda, m, n, T)$$

We finally consider a notion of RE that is weaker than sublinear-compactness, where we allow the encoding time to be polynomially dependent on the time bound T, but still require the encoding size be sub-linear in T. We call such RE schemes *weakly sublinear compact.*

Definition 22 (Weakly Sublinear Compact Randomized Encoding scheme). *We say a randomized encoding scheme* $\mathsf{RE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval})$ *in the CRS model for a class of Turing machines* $\{\mathcal{M}_\lambda\}$ *is* weakly sublinear compact *if the efficiency requirement on* Enc *in Definition 21 is changed to: For some constant* $\varepsilon \in (0, 1)$,

$$\mathsf{Time}_{\mathsf{Enc}}(pk, \Pi, x) = \mathrm{poly}(\lambda, |\Pi|, |x|, T)$$
$$\mathsf{outlen}_{\mathsf{Enc}}(pk, \Pi, x) = T^{1-\epsilon} \cdot \mathrm{poly}(\lambda, |\Pi|, |x|)$$

Next, we observe that RE schemes satisfying the notions defined above (*i.e.* succinctness and weak sublinear compactness) can be composed to get a RE scheme satisfying sub-linear compactness. In particular, by composing a succinct RE scheme with a weakly compact RE scheme, one can obtain a sub-linearly compact RE scheme. We defer the proof to the full version of the paper.

Theorem 9. *Assume the existence of pseudorandom generators. If there is a* succinct *RE scheme and a* weakly sublinear compact *RE scheme for Turing machines, then there is a* sub-linearly compact randomized encoding *scheme for Turing machines.*

5.3 Randomized encodings with CRS from Compact Functional Encryption

In this section we construct RE schemes in the CRS model from Compact Functional encryption schemes and pseudorandom generators.

Let $(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be a public key, compact functional encryption scheme for $\mathbf{P}/poly$, and let PRG be a pseudorandom generator. We define a randomized encoding scheme in the CRS model $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Eval})$ as follows.

The setup algorithm $\mathsf{Setup}(1^\lambda, 1^m, 1^n, 1^T, 1^l)$:

- Setup first generates keys for the functional encryption scheme $(mpk, msk) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and samples a uniformly random string $s \leftarrow \{0, 1\}^\lambda$.

- Next, it generates the string $c \leftarrow 0^l \oplus \mathsf{PRG}(s, l)$. That is, it encrypts 0^l using a one-time pad with the key coming from $\mathsf{PRG}(s, l)$
- Let U be the universal circuit that on input (Π, x) where $|\Pi| \leq m$ and $|x| \leq n$ runs machine Π on x for at most T steps and outputs the first l bits of the tape as output. We define a circuit $C_{U,c}$, that has the string c and circuit U hardcoded in it, as follows.
  1. $C_{U,c}$ takes as input (Π, x, s', b) where (Π, x) satisfies the size constraints as described above, $s' \in \{0,1\}^\lambda$ and $b \in \{0,1\}$.
  2. If $b = 0$ then $C_{U,c}$ outputs $U(\Pi, x)$.
  3. Otherwise $C_{U,c}$ outputs $c \oplus \mathsf{PRG}(s')$.
- Setup runs $sk_C \leftarrow \mathsf{FE.KeyGen}(msk, C_{U,c})$ and outputs sk_C as the common reference string crs and mpk as the public encoding key pk

The encoding algorithm $\mathsf{Enc}(pk, \Pi, x)$**:** Enc parses pk as the functional public key mpk and runs $ct \leftarrow \mathsf{FE.Enc}(mpk, (\Pi, x, 0^\lambda, 0))$. Enc outputs the functional ciphertext ct as the encoding $\hat{\Pi}_x$.

The evaluation algorithm $\mathsf{Eval}(\hat{\Pi}_x, \mathsf{crs})$: Eval parses $\hat{\Pi}_x$ as a functional ciphertext ct and crs as the functional secret key $sk_{C_{U,c}}$. Eval runs $y \leftarrow \mathsf{FE.Dec}(sk_{C_{U,c}}, ct)$ and outputs y.

The correctness of the above encoding scheme follows directly from that of the underlying functional encryption scheme. When a randomized encoding of (Π, x) is evaluated, it outputs the result of running the universal circuit U on (Π, x) that is $\Pi^T(x)$. Also the efficiency properties of the above scheme follow directly from the compactness properties of the functional encryption scheme. For example, if the functional encryption scheme we start from has sub-linear compactness (the ciphertext size is sub-linear in the circuit size of the function for which the functional secret keys are generated) then we get an encoding scheme with sub-linear compactness.

We have the following theorem. We refer the reader to the full version for the proof.

Theorem 10. *Let* $(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *be a public key functional encryption scheme for* $\mathbf{P}/poly$ *with* $(\lambda_0, S(\cdot))$ *selective security, and let* PRG *be a pseudorandom generator with* $(\lambda_0, S(\cdot))$ *security. The randomized encoding scheme defined above is* $(\lambda_0, \frac{S(\cdot)}{4})$*-simulation secure.*

Corollary 1. *If there exists a public key, compact (resp. succinct, weakly sublinear compact) functional encryption for* $\mathsf{P/poly}$ *scheme with selective security, and a secure PRG, then there exists a compact (resp. succinct*[9]*, weakly sublinear compact) randomized encoding scheme for Turing machines in the CRS model that is simulation secure.*

[9] We note that for succinct RE, we first apply the transformation from succinct FE to get succinct RE with 1-bit output, and to encode Turing Machines with multi-bit outputs, we generate one such RE for each output bit

The above theorem and corollary also work in the regime of sub-exponential security. That is, starting with a functional encryption scheme and pseudorandom generator that are sub-exponentially secure we obtain a RE scheme with sub-exponential security.

The following corollary is obtained by combining Corollary 1 with Theorem 6 and Theorem 7. While we use this corollary in our results, we believe it is of independent interest too. Succinct RE schemes for Turing machines were shown by [BGL$^+$15] to have a variety of applications. However the only known construction of it ([KLW14]) relies on **iO** for circuits. We observe that in the CRS model, succinct RE schemes can be based simply on LWE.

Corollary 2. *Assuming LWE (resp. with sub-exponential hardness), there exists a succinct RE scheme for Turing machines in the CRS model with (resp. sub-exponential) simulation security.*

Finally, the following corollary shows that, assuming LWE, weakly sublinear compact FE is sufficient to construct sublinearly-compact RE in the CRS model. This corollary follows by combining Corollary 1, which shows that weakly sublinear compact FE implies weakly sublinear compact RE in the CRS model, Corollary 2, which constructs succinct RE in the CRS model from LWE, and finally Theorem 9, which shows that weakly sublinear compact RE and succinct RE can be combined to produce sublinearly-compact RE in the CRS model.

Corollary 3. *Assuming LWE (resp. with sub-exponential hardness), if there exists a weakly sublinear compact FE scheme for* $\mathsf{P/poly}$ *(resp. with sub-exponential security), then there exists a sublinearly-compact RE scheme for Turing machines in the CRS model with (resp. sub-exponential) simulation security.*

5.4 IO for Circuits from RE in the CRS model

In this section we show that compact RE schemes for Turing machines in the CRS model implies **iO** for circuits; combining with the result of [KLW14] that **iO** for circuits implies **iO** for (bounded-input) Turing machines, we obtain the following theorem:

Theorem 11. *Assume the existence of sub-exponentially secure one-way functions. If there exists a sublinearly compact randomized encoding scheme in the CRS model with sub-exponential simulation security, then there exists an bounded-input indistinguishability obfuscator for Turning machines.*

We note that the theorem also holds w.r.t. sublinearly compact randomized encoding scheme in the CRS model, satisfying, weaker, distributional indistinguishability security, *with auxiliary inputs* (i.e., Definition 14 w.r.t. distributions $\{D_{b,\lambda}\}$ that additionally samples an auxiliary input z_b, and the security requirement is that if the output distributions together with the auxiliary inputs are indistinguishable, then the encodings together with the auxiliary inputs are also indistinguishable, with appropriate security loss). Since the distributional indistinguishability security is implied by simulation security, and in the CRS model, we can construct sublinearly compact RE with simulation security from sublinearly compact FE schemes, for simplicity, we directly state and prove the theorem w.r.t. simulation security.

The construction and proof is very similar to that of unbounded-input **iO** from compact RE schemes in the plain model presented in Section 4. We refer the reader to the full version [LPST15] for more details.

5.5 Summary of Results using RE in the CRS model

We observe that by combining Theorem 11 with Corollary 1, we reprove the results of [AJ15,BV15]

Theorem 12. *Assuming the existence of compact functional encryption with subexponential security, there exists a bounded-input indistinguishability obfuscator for Turing Machines.*

Further, we get the following new result, as a consequence of Corollary 3 and Theorem 11:

Theorem 13. *Assuming the existence of weakly sublinear compact functional encryption with subexponential security and LWE with subexponential security, there exists a bounded-input indistinguishability obfuscator for Turing Machines.*

6 Impossibility of Compact RE

In this section, we mention several impossibility results related to sublinear (and hence compact) RE with different security. We refer the reader to the full version [LPST15] for the proofs.

Theorem 14. *The following impossibility results hold in the* plain *model:*

1. *Sublinear randomized encoding schemes with (polynomial) simulation security do not exist, assuming one-way functions.*
2. *Sublinear randomized encoding schemes with* sub-exponential *indistinguishability security do not exist, assuming sub-exponentially secure one-way functions.*
3. *Sublinear randomized encoding schemes with (polynomial) indistinguishability security do not exist, assuming bounded-input* ***iO*** *for Turing machines and one-way functions.*

Acknowledgment: We are extremely grateful to Nir Bitansky and Omer Paneth for informing us of their impossibility result for compact RE assuming differing-input obfuscation, SNARKs and collision-resistant hash functions; this results was the inspiration behind our main impossibility result. We are also very grateful to them for many delightful and insightful discussions.

References

[ABG$^+$13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. *IACR Cryptology ePrint Archive*, 2013:689, 2013.

[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.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

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

[App11] Benny Applebaum. Randomly encoding functions: A new cryptographic paradigm - (invited talk). In *Information Theoretic Security - 5th International Conference, ICITS 2011, Amsterdam, The Netherlands, May 21-24, 2011. Proceedings*, pages 25–31, 2011.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, volume 0, pages 106–115, 2001.

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

[BCPR13] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. *IACR Cryptology ePrint Archive*, 2013:641, 2013.

[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.

[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.

[BKS15] Zvika Brakerski, Ilan Komargodski, and Gil Segev. From single-input to multi-input functional encryption in the private-key setting. *IACR Cryptology ePrint Archive*, 2015:158, 2015.

[BP15] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 401–427, 2015.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011.

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

[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.

[GGH$^+$13] 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.

[GGHW13] Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In *CRYPTO'14*, 2013.

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

[GKP$^+$13a] 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.

[GKP$^+$13b] 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.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554, 2013.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304, 2000.

[IK02a] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP*, pages 244–256, 2002.

[IK02b] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002, Malaga, Spain, July 8-13, 2002, Proceedings*, pages 244–256, 2002.

[IPS15] Yuval Ishai, Omkant Pandey, and Amit Sahai. Public-coin differing-inputs obfuscation and its applications. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 668–697, 2015.

[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.

[KMN$^+$14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. 2014.

[LPST15] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. Cryptology ePrint Archive, Report 2015/720, 2015. `http://eprint.iacr.org/`.

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

[Yao82] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164, 1982.