Indistinguishability Obfuscation from Semantically-Secure Multilinear Encodings

Rafael Pass*, Karn Seth, and Sidharth Telang

Cornell University
{rafael,karn,sidtelang}@cs.cornell.edu

Abstract. We define a notion of semantic security of multilinear (a.k.a. graded) encoding schemes, which stipulates security of a class of algebraic "decisional" assumptions: roughly speaking, we require that for every nuPPT distribution D over two *constant-length* sequences $\boldsymbol{m}_0, \boldsymbol{m}_1$ and auxiliary elements $\boldsymbol{z}$ such that all arithmetic circuits (respecting the multilinear restrictions and ending with a zero-test) are *constant* with overwhelming probability over $(\boldsymbol{m}_b, \boldsymbol{z})$, $b \in \{0,1\}$, we have that encodings of $\boldsymbol{m}_0, \boldsymbol{z}$ are computationally indistinguishable from encodings of $\boldsymbol{m}_1, \boldsymbol{z}$. Assuming the existence of semantically secure multilinear encodings and the LWE assumption, we demonstrate the existence of indistinguishability obfuscators for all polynomial-size circuits.

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. Precisely defining what it means to "scramble" a program is non-trivial: on the one hand, we want a definition that can be plausibly satisfied, on the other hand, we want a definition that is useful for applications.

Hada [Had00] and Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan, and Yang [BGI$^+$01] show that simulation-based notion such as *virtual black-box obfuscation (VBB)* [BGI$^+$01]—which, roughly speaking, require that everything that can be learn from the code of the obfuscated program can be simulated using just black-box access to the functionality—run into strong impossibility results.

We here focus on the notion of *indistinguishability obfuscation*, first defined by Barak *et al.* [BGI$^+$01] and explored by Garg, Gentry, Halevi, Raykova, Sahai, and Waters [GGH$^+$13b]. Roughly speaking, this notion requires that obfuscations $\mathcal{O}(C_1)$ and $\mathcal{O}(C_2)$ of any two *equivalent* circuits C_1 and C_2 (i.e., whose

* Work supported in part by a Microsoft 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.

outputs agree on all inputs) from some class $\mathcal{C}$ are computationally indistinguishable. In a very recent breakthrough result, Garg, Gentry, Halevi, Raykova, Sahai, and Waters [GGH$^+$13b] provided the first candidate constructions of indistinguishability obfuscators for all polynomial-size circuits, based on so-called *multilinear (a.k.a. graded) encodings* [BS03,Rot13,GGH13a]—for which candidate constructions were recently discovered in the seminal work of Garg, Gentry and Halevi [GGH13a], and more recently, alternative constructions were provided by Coron, Lepoint and Tibouchi [CLT13].

The obfuscator construction of Garg et al proceeds in two steps. They first provide a candidate construction of an indistinguishability obfuscator for NC^1 (this construction is essentially assumed to be secure); next, they demonstrate a "bootstrapping" theorem showing how to use fully homomorphic encryption (FHE) schemes [Gen09] and indistinguishability obfuscators for NC^1 to obtain indistinguishability obfuscators for all polynomial-size circuits. Further constructions of obfuscators for NC^1 were subsequently provided by Brakerski and Rothblum [BR14] and Barak, Garg, Kalai, Paneth and Sahai [BGK$^+$13]—in fact, these constructions achieve the even stronger notion of virtual-black-box obfuscation in idealized "generic" multilinear encoding models.

In parallel with the development of candidate obfuscation constructions, several surprising applications of indistinguishability have emerged (see e.g., [GGH$^+$13b,SW14,HSW14,BZ14,GGHR14,BCP14,BCPR14,GGG$^+$14,KNY14,KMN$^+$14]). Furthermore, as shown by Goldwasser and Rothblum [GR07], indistinguishability obfuscators provide a very nice "best-possible" obfuscation guarantee: if a functionality can be VBB obfuscated (even non-efficiently!), then any indistinguishability obfuscator for this functionality is VBB secure.

1.1 Towards "Provably-Secure" Obfuscation

But despite these amazing developments, the following question remains open:

> *Can the security of general-purpose indistinguishability obfuscators be reduced to some "natural" intractability assumption?*

The principal goal of the current paper is to make progress toward addressing this question.

Note that while the construction of indistinguishability obfuscation of Garg et al is based on *some* intractability assumption, the assumption is very tightly tied to their scheme—in essence, the assumption stipulates that their scheme is a secure indistinguishability obfuscator.

The VBB constructions of Brakerski and Rothblum [BR14] and Barak et al [BGK$^+$13] give us more confidence in the plausible security of their obfuscators, in that they show that at least "generic" attacks—that treat multilinear encoding as if they were "physical envelopes" on which multilinear operations can be performed—cannot be used to break security of the obfuscators. But at the same time, non-generic attacks against their scheme are known—since general-purpose VBB obfuscation is impossible. Thus, it is not clear to what extent security arguments in the generic multilinear encoding model should make us more confident that these constructions satisfy e.g., a notion of indistinguishability obfuscation.[1] In particular, the question of to what extent one can capture "real-world" security properties from security proofs in the generic model through a "meta-assumption" (regarding multilinear encoding) was raised (but not investigated) in [BGK$^+$13]; see Remark 1 there. In this work, we initiate a study of this question.

1.2 Security of Multilinear (Graded) Encodings

Towards explaining the assumptions we consider, let us start by briefly recalling multilinear (a.k.a. graded) encoding schemes [GGH13a,GGH$^+$13b]. Roughly speaking, such schemes enable anyone that has access to a *public parameter* pp and *encodings* $E_S^x = \mathsf{Enc}(x, S)$, $E_S^y = \mathsf{Enc}(y, S')$ of ring elements x, y under the sets $S, S' \subset [k]$ to *efficiently*:[2]

- compute an encoding $E_{S \cup S'}^{x \cdot y}$ of $x \cdot y$ under the set $S \cup S'$, as long as $S \cap S' = \emptyset$;
- compute an encoding E_S^{x+y} of $x + y$ under the set S as long as $S = S'$;
- compute an encoding E_S^{x-y} of $x - y$ under the set S as long as $S = S'$.

(Given just access to the public-parameter pp, generating an encoding to a particular element x may not be efficient; however, it can be efficiently done given access to the *secret parameter* sp.) Additionally, given an encoding E_S^x where the set S is the whole universe $[k]$—called the "target set"—we can efficiently check whether $x = 0$ (i.e., we can "zero-test" encodings under the target set $[k]$.) In essence, multilinear encodings enable computations of certain restricted set of arithmetic circuits (determined by the sets S under which the elements are encoded) and finally determine whether the output of the circuit is 0; we refer to these as the *legal arithmetic circuits.*

Semantical Security of Multilinear (Graded) Encodings The above description only explains the *functionality* of multlinear encodings, but does not discuss *security.* As far as we are aware, there have been two approaches to defining security of multilinear encodings. The first approach, initiated in [GGH13a], stipulates specific hardness assumptions closely related to the DDH assumption.

[1] In fact, mirroring ideas from [GGSW13], assuming the existence of indistinguishability obfuscation and one-way functions it is easy to come up with a method to sample C_1, C_2, z such that with high probability $C_1(z) \neq C_2(z)$ (and thus, given z, we can easily distinguish obfuscations of them), yet the pair of circuits (C_1, C_2) are indistinguishable from a pair of functionally equivalent circuits. Thus, there are "fake attacks" on indistinguishability obfuscation that cannot be efficiently distinguished from a real attack.

[2] Just as [BR14,BGK$^+$13], we here rely on "set-based" graded encoding; these were originally called "generalized" graded encodings in [GGH13a]. Following [GGH$^+$13b,BGK$^+$13] (and in particular the notion of a "multilinear jigsaw puzzles" in [GGH$^+$13b]), we additionally enable anyone with the secret parameter to encode *any* elements (as opposed to just *random* elements as in [GGH13a]).

The second approach instead focuses on *generic attackers* and assumes that the attacker does not get to see the actual encodings but instead can only access them through legal arithmetic circuits.

In this work, we consider the first approach, but attempt to capture a general *class* of algebraic "decisional" assumptions (such as the the graded DDH assumption of [GGH13a]) which holds against generic attackers (and as such, it can be viewed as a merge of the two approaches). In essence, our notion of (single-message) *semantical security* attempts to capture the intuition that encodings of elements m_0 and m_1 (under the set S) are indistinguishable in the presence of encodings of "auxiliary" elements $\boldsymbol{z}$ (under sets $\boldsymbol{T}$), as long as $m_0, m_1, \boldsymbol{z}$ are sampled from *any* "nice" distribution D; in the context of a graded DDH assumption, think of $\boldsymbol{z}$ as a vector of independent uniform elements, m_0 as the product of the elements in $\boldsymbol{z}$ and m_1 as an independent uniform element. We analogously consider stronger notions of *constant-message* and *multi-message* semantical security, where m_0, m_1 (and S) are replaced by either constant-length or arbitrary polynomial-length vectors $\boldsymbol{m}_0, \boldsymbol{m}_1$ of elements (and sets $\boldsymbol{S}$).

Defining what makes a distribution D "nice" turns out to be quite non-trivial: A first (and minimal) approach—similar to e.g., the uber assumption of [BBG05] in the context of bilinear maps—would be to simply require that D samples elements $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$ such that no generic attacker can distinguish $\boldsymbol{m}_0, \boldsymbol{z}$ and $\boldsymbol{m}_0, \boldsymbol{z}$. As we discuss in Section 1.3, the most natural formalization of this approach can be attacked assuming standard cryptographic hardness assumptions. The distribution D considered in the attack, however, is "unnatural" in the sense that encodings of $\boldsymbol{m}_b, \boldsymbol{z}$ actually leak information about $\boldsymbol{m}_b$ even to generic attackers (in fact, this information fully determines the bit b, it is just that it cannot be computed in polynomial time).

Our notion of a *valid* message distribution disallows such information leakage w.r.t. generic attacks. More precisely, we require that every (even unbounded-size) legal arithmetic circuit C is *constant* over $(m_b, \boldsymbol{z})$, $b \in \{0, 1\}$ with overwhelming probability; that is, there exists some bit c such that with overwhelming probability over $m_0, m_1, \boldsymbol{z} \leftarrow D$, $C(m_b, \boldsymbol{z}) = c$ for $b \in \{0, 1\}$ (recall that a legal arithmetic circuit needs to end with a zero-test and thus the output of the circuit will be either 0 or 1). We refer to any distribution D satisfying this property as being *valid*, and our formal definition of semantical security now only quantifies over such valid message distributions.

Obfuscation from Semantically-Secure Multilinear Encodings As a starting point, we observe that slight variants of the constructions of [BR14,BGK$^+$13] can be shown to satisfy indistinguishability obfuscation for NC^1 assuming *multi-message* semantically-secure multilinear encodings. In fact, any VBB secure obfuscation in the generic model where the construction only releases encodings of elements (as the constructions of [BR14,BGK$^+$13] do) satisfies indistinguishability obfuscation assuming a slight strengthening of multi-message semantical security where validity only consider *polynomial-size* (as opposed to arbitrary-size) legal arithmetic circuits:[3] let $\boldsymbol{m}_0$ denote the elements corresponding to

[3] We thank Sanjam Garg for this observation.

an obfuscation of some program $\mathit{\Pi}_0$, and $\boldsymbol{m}_1$ the elements corresponding to an obfuscation of some functionally equivalent program $\mathit{\Pi}_1$. VBB security implies that all polynomial-size legal arithmetic circuits are constant with overwhelming probability over both $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ (as any such query can be simulated given black-box access to the functionality of the program), and thus encodings of $\boldsymbol{m}_0$ and $\boldsymbol{m}_1$ (i.e., obfuscations of $\mathit{\Pi}_0$ and $\mathit{\Pi}_1$) are indistinguishable. By slightly tweaking the construction of [BGK+13] and the analysis[4], we can extend this to hold against *all* (arbitrary-size) legal arithmetic circuits, and thus indistinguishability of the encodings (which implies indistinguishability of the obfuscations) follows as a direct consequence of the multi-message security assumption.

While this observation does takes us a step closer towards basing the security of obfuscation on a simple, natural, assumption, it is unappealing in that the assumption itself directly implies the security of the scheme (without any security reduction).

Our central result shows how to construct indistinguishability obfuscators for NC^1 based on the existence of *constant-message* semantically-secure multilinear encodings; in the sequel, we simply refer to such schemes as being semantically secure (dropping "constant-message" from the notation). Note that the constant-message restriction not only simplifes (and reduces the complexity) of the assumption, it also takes us a step closer to the more standard GDDH assumption. (As far as we know, essentially all "DDH-type" assumptions in "standard"/bilinear or multilinear settings consider a constant-message setting, stipulating indistinguishability of either a *single* or a *constant* number of elements in the presence of polynomially many auxiliary elements. It is thus safe to say that such constant-message indistinguishability assumptions are significantly better understood than their multi-message counterpart.)

Theorem 1 (Informally stated) *Assume the existence of semantically secure multilinear encodings. Then there exists an indistinguishability obfuscator for* NC^1.

As far as we know, this is the first result presenting indistinguishability obfuscators for NC^1 based on any type of assumption with a "non-trivial" security reduction w.r.t. arbitrary nuPPT attackers.

The core of our result is a general technique for transforming any obfuscator for matrix branching programs that satisfies a weak notion of *neighboring-matrix* indistinguishability obfuscation—which roughly speaking only requires indistinguishability of obfuscations of branching programs that differ only in a constant number of matrices—into a "full-fledged" indistinguishability obfuscator. We next show how to adapt the construction of [BGK+13] and its analysis to satisfy neighboring-matrix indistinguishability obfuscation based on semantical secure multilinear encodings; on a high-level, the security analysis in the generic model is useful for proving that the particular message distribution we consider is "valid".[5]

[4] Briefly, we need to tweak the construction to ensure a "perfect" simulation property.

[5] As we explain in more details later, to use our transformation, we need to deal with branching programs that satisfy a slightly more liberal definition of a branching

If additionally assuming the existence of a leveled FHE [RAD78,Gen09] with decryption in NC^1—implied, for instance, by the LWE assumption [BV11,BGV12]—our construction can be bootstrapped up to obtain indistinguishability obfuscators for all polynomial-size circuits by relying on the technique from [GGH$^+$13b].

Theorem 2 (Informally stated) *Assume the existence of semantically secure multilinear encodings and a leveled FHE with decryption in NC^1. Then there exists indistinguishability obfuscators for $\mathsf{P}/poly$.*

Semantical Security w.r.t. Restricted Classes of Distributions Our most basic notion of semantical security requires indistinguishability to hold w.r.t. to *any* "valid" message distribution. This may seem like a strong assumption. Firstly, such a notion can clearly not be satisfied by a *deterministic* encoding schemes (as envisioned in the original work of [BS03])—we can never expect encodings of 0 and 1 (under a non target set, and without any auxiliary inputs) to be indistinguishable. Secondly, even if we have a randomized encoding scheme in mind (such as the candidates of [GGH13a,CLT13]), giving the attacker access to encodings of *arbitrary* elements may be dangerous: As mentioned in [GGH13a], attacks (referred to as "weak discrete logarithm attacks") on their scheme are known in settings where the attacker can get access to "non-trivial" encodings of 0 under any *non-target* set $S \subset [k]$. (We mention that, as far as we know, no such attacks are currently known on the candidate construction of [CLT13].)

For the purposes of the results in our paper, however, it suffices to consider a notion of semantical security w.r.t. *restricted classes of distributions* D. In particular, to deal with both of the above issues, we consider "high-entropy" distributions D that sample elements $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$ such that 1) each individual element has high-entropy, and 2) any element, associated with a *non-target* set $S \subset [k]$, that can be obtained by applying "legal" algebraic operations to $(\boldsymbol{m_b}, \boldsymbol{z})$ (for $b \in \{0,1\}$) has high-entropy (and thus is non-zero with overwhelming probability).[6] We refer to such message distributions as being *entropically valid.*

Basing Security on "Single-Distribution" Semantical Security The assumption that a scheme satisfies semantical security may be viewed as an (exponential-size) *class* of algebraic "decisional" assumptions (or as a "meta"-assumption, just like the "uber assumption" of [BBG05]): we have one assumption for each valid message distributions D. Indeed, to prove indistinguishability of obfuscations of two circuits C_0, C_1, we rely on an instance in this class which is a function of the circuits C_0, C_1—in the language of [GGSW13,GLW14], security is thus based on an "instance-dependent" assumption.

This view-point also clarifies that semantical security is not an *efficiently falsifiable* assumption [Nao03]: the problem is that there may not exist an efficient

program than what is used in earlier works. This is key reason why we need to modify the construction and analysis from [BGK$^+$13].

[6] Technically, by high-entropy, we here mean that the min-entropy is at least $\log |R| - O(\log\log |R|)$ where R is the ring associated with the encodings; that is, the min-entropy is "almost" optimal (i.e., $\log |R|$).

way of checking whether a distribution D is valid (as this requires checking that *all* legal arithmetic circuits are constant with overwhelming probability, which in our particular case would require checking whether C_0 and C_1 are functionally equivalent).

We finally observe that both of these issues can be overcome if we make subexponential hardness assumptions: there exists a single (uniform PPT samplable) distribution Sam over (nuPPT message distributions D) that are *provably* entropically valid such that it suffices to assume the existence of an encoding scheme that is entropic semantically secure w.r.t., this particular distribution with *subexponentially small indistinguishability gap*.[7] Note that this is a single, non-interactive and efficiently falsifiable, decisional assumption.

1.3 Alternative Notions of Semantical Security

We finally investigate various ways of defining a "super" (or uber) assumption for multilinear encodings. As mentioned above, a natural way of defining security of multilinear encodings would be to require that for specific classes of problems, generic attacks cannot be beaten (this is the approach alluded to in [BGK$^+$13]). Perhaps the most natural instantiation of this in the context of a multilinear DDH assumption would be to require that for any distribution D over $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$ (where $\boldsymbol{m}_0, \boldsymbol{m}_1$ are constant-length sequences), if encodings of $\boldsymbol{m}_0, \boldsymbol{z}$ and and $\boldsymbol{m}_0, \boldsymbol{z}$ are indistinguishable w.r.t. to generic attackers, then they are also indistinguishable w.r.t. arbitrary nuPPT attackers; in essence, "if an algerbraic decisional assumption holds w.r.t. to generic attacks, then it also holds with respect to nuPPT attackers". We refer to this notion of security as *extractable uber security*.[8]

Our second main result shows that, assuming the existence of a leveled FHE with decryption in NC^1, there do not exist extractable uber-secure multilinear encodings (even if we only require security to hold w.r.t high-entropy distributions D).

Theorem 3 (Informally stated) *Assume the existence of a leveled FHE with decryption in NC^1. Then no multilinear encodings can satisfy extractable (entropic) uber security.*

The high-level idea behind this result is to rely on the "conflict" between the feasibility of VBB obfuscation in the generic model of [BGK$^+$13] and the impossibility of VBB obfuscation in the "standard" model [BGI$^+$01]: we let $\boldsymbol{m}_b, \boldsymbol{z}$ contain a generically-secure VBB obfuscation of a program Π_b that hides b given just black-box access to Π_b, yet b can be recovered given the code of Π_b. By

[7] These results were added to our e-Print report April 25, 2014, motivated in part by [GLW14] (which bases witness encryption on an instant-independent assumption) and a conversation with Amit Sahai.

[8] We use the adjective "extractable" as this security notion implies that if an nuPPT attacker can distinguish encodings, then the arithmetic circuits needed to distinguish the elements can be efficiently extracted out.

generic security of the obfuscation, it follows that *efficient* generic attackers cannot distinguish $\boldsymbol{m}_0, \boldsymbol{z}$ and $\boldsymbol{m}_1, \boldsymbol{z}$ yet, "non-generic" (i.e., standard PPT) attackers can. In our formal treatment, to rule out *constant-message* (as opposed to multi-message) security, we rely on a variant of the obfuscator presented in this paper, enhanced using techniques from [BGK$^+$13].

We emphasize that in the above attack it is cruicial that we restrict to efficient (nuPPT) generic attacks. In the full version of the paper we consider several plausible ways of defining uber security for multilinear encodings, which circumvent the above impossibility results by requiring indistinguishability of encodings only if the encodings are *statistically* close w.r.t. *unbounded* generic attackers (that are restricted to making polynomially many zero-test queries). We highlight that none of these assumptions are needed for our construction of an indistinguishability obfuscation and are stronger than semantical security, but they may find other applications.

2 Definition of Semantically Secure Graded Encodings

2.1 Graded Encoding Schemes

Graded (multilinear) encoding schemes were originally introduced in the work of Garg, Gentry and Halevi [GGH13a]. Just as [BR14,BGK$^+$13], we here rely on "set-based" (or "asymmetric") graded encoding; these were originally called "generalized" graded encodings in [GGH13a]. Following [GGH$^+$13b,BGK$^+$13] and the notion of "multilinear jigsaw puzzles" from [GGH$^+$13b], we additionally enable anyone with the secret parameter to encode *any* elements (as opposed to just *random* elements as in [GGH13a]).

Definition 1 ((k, R)-Graded Encoding Scheme) *A (k, R)-graded encoding scheme for $k \in \mathbb{N}$ and ring R is a collection of sets $\{E_S^\alpha : \alpha \in R, S \subseteq [k]\}$ with the following properties*

- *For every $S \subseteq [k]$ the sets $\{E_S^\alpha : a \in R\}$ are disjoint.*
- *There are associative binary operations $\oplus$ and $\ominus$ such that for every $\alpha_1, \alpha_2 \in R$, $S \subseteq [k]$, $u_1 \in E_S^{\alpha_1}$ and $u_2 \in E_S^{\alpha_2}$ it holds that $u_1 \oplus u_2 \in E_S^{\alpha_1+\alpha_2}$ and $u_1 \ominus u_2 \in E_S^{\alpha_1-\alpha_2}$ where '$+$' and '$-$' are the addition and subtraction operations in R.*
- *There is an associative binary operation $\otimes$ such that for every $\alpha_1, \alpha_2 \in R$, $S_1, S_2 \subseteq [k]$ such that $S_1 \cap S_2 = \emptyset$, $u_1 \in E_{S_1}^{\alpha_1}$ and $u_2 \in E_{S_2}^{\alpha_2}$ it holds that $u_1 \otimes u_2 \in E_{S_1 \cup S_2}^{\alpha_1 \cdot \alpha_2}$ where '$\cdot$' is multiplication in R.*

Definition 2 (Graded Encoded Scheme) *A graded encoding scheme $\mathcal{E}$ is associated with a tuple of PPT algorithms,* $(\mathsf{InstGen}_\mathcal{E}, \mathsf{Enc}_\mathcal{E}, \mathsf{Add}_\mathcal{E}, \mathsf{Sub}_\mathcal{E}, \mathsf{Mult}_\mathcal{E}, \mathsf{isZero}_\mathcal{E})$ *which behave as follows:*

- *Instance Generation:* $\mathsf{InstGen}_\mathcal{E}$ *takes as input the security parameter 1^n and multilinearity parameter 1^k, and outputs secret parameters* sp *and public parameters* pp *which describe a (k, R)-graded encoding scheme $\{E_S^\alpha : \alpha \in$*

$R, S \subseteq [k]\}$. We refer to E^{α}_{S} as the set of encodings of the pair (α, S). We restrict to graded encoding schemes where R is $\mathbb{Z}_p$ and p is a prime exponential in n and k.

- *Encoding:* $\mathsf{Enc}_{\mathcal{E}}$ *takes as input the secret parameters* sp*, an element $\alpha \in R$ and set $S \subseteq [k]$, and outputs a random encoding of the pair (α, S).*
- *Addition:* $\mathsf{Add}_{\mathcal{E}}$ *takes as input the public parameters* pp *and encodings $u_1 \in E^{\alpha_1}_{S_1}, u_2 \in E^{\alpha_2}_{S_2}$, and outputs an encoding of the pair $(\alpha_1 + \alpha_2, S)$ if $S_1 = S_2 = S$ and outputs $\perp$ otherwise.*
- *Negation:* $\mathsf{Sub}_{\mathcal{E}}$ *takes as input the public parameters* pp *and encodings $u_1 \in E^{\alpha_1}_{S_1}, u_2 \in E^{\alpha_2}_{S_2}$, and outputs an encoding of the pair $(\alpha_1 - \alpha_2, S)$ if $S_1 = S_2 = S$ and outputs $\perp$ otherwise.*
- *Multiplication:* $\mathsf{Mult}_{\mathcal{E}}$ *takes as input the the public parameters* pp *and encodings $u_1 \in E^{\alpha_1}_{S_1}, u_2 \in E^{\alpha_2}_{S_2}$, and outputs an encoding of the pair $(\alpha_1 \cdot \alpha_2, S_1 \cup S_2)$ if $S_1 \cap S_2 = \emptyset$ and outputs $\perp$ otherwise.*
- *Zero testing:* $\mathsf{isZero}_{\mathcal{E}}$ *takes as input the public parameters* pp *and an encoding $u \in E_S(\alpha)$, and outputs 1 if and only if $\alpha = 0$ and S is the universe set $[k]$.*[9]

Whenever it is clear from the context, to simplify notation we drop the subscript $\mathcal{E}$ when we refer to the above procedures (and simply call them $\mathsf{InstGen}, \mathsf{Enc}, \ldots$*).*

In known candidate constructions [GGH13a,CLT13], encodings are "noisy" and the noise level increases with each operation; the parameters, however, are set so that any $\mathrm{poly}(n, k)$ operations can be performed without running into trouble. For convenience of notation (and just like all other works in the area), we ignore this noise issue.[10]

Note that the above procedures allow algebraic operations on the encodings in a restricted way. Given the public parameters and encodings made under the sets $\boldsymbol{S}$, one can only perform algebraic operations that are allowed by the structure of the sets in $\boldsymbol{S}$. We call such operations $\boldsymbol{S}$-respecting and formalize this notion as follows:

Definition 3 (Set Respecting Arithmetic Circuits) *For any sequence $\boldsymbol{S}$ of subsets of $[k]$, we say that an arithmetic circuit C (i.e. gates perform only ring operations $\{+, -, \cdot\}$) is $\boldsymbol{S}$-*respecting *if it holds that*

[9] In the candidate scheme given by [GGH13a], isZero may not have perfect correctness: the generated instances $(\mathsf{pp}, \mathsf{sp})$ can be "bad" with some negligible probability, so that there could exist an encoding u of a nonzero element where $\mathsf{isZero}(\mathsf{pp}, u) = 1$. However, these "bad" parameters can be efficiently detected during the execution of $\mathsf{InstGen}$. We can thus modify the encoding scheme to simply set $\mathsf{Enc}(\mathsf{pp}, e) = e$ whenever the parameters are "bad" (and appropriately modify $\mathsf{Add}, \mathsf{Sub}, \mathsf{Mult}$ and isZero so that the operate on "unencoded" elements. This change ensures that, for every pp, including "bad" ones, the zero test procedure isZero works with perfect correctness. We note that since bad parameters occur only with negligible probability, this change does not affect the security of the encodings.

[10] The above definition can be easily generalized to deal with the candidates by only requiring that the above conditions hold when u_1, u_2 have been obtained by $\mathrm{poly}(n, k)$ operations.

- *Eevery input wire of C is tagged with some set in $\boldsymbol{S}$.*
- *For every $+$ and $-$ gate in C, if the tags of the two input wires are the same set S then the output wire of the gate is tagged with S. Otherwise the output wire is tagged with $\perp$.*
- *For every $\cdot$ gate in C, if the tags of the two input wires are sets S_1 and S_2 and $S_1 \cap S_2 = \emptyset$ then the output wire of the gate is tagged with $S_1 \cup S_2$. Otherwise the output wire is tagged with $\perp$.*
- *It holds that the output wire is tagged with the universe set $[k]$.*[11]

2.2 Semantical Security

We now turn to defining semantical security of graded encoding schemes. As outlined in the introduction, we start by defining the notion of a respecting (or valid) message sampler w.r.t. to sets $\boldsymbol{S}, \boldsymbol{T}$. Such a message sampler samples elements $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$ such that for every $(\boldsymbol{S}, \boldsymbol{T})$-respecting circuit C, $\mathsf{isZero}(C(\cdot))$ is *constant* over $(m_b, \boldsymbol{z})$, $b \in \{0,1\}$ with overwhelming probability.

Definition 4 (Respecting Message Sampler) *Let $\mathcal{E}$ be a graded encoding scheme, and $\{(\boldsymbol{S}_n, \boldsymbol{T_n})\}_{n\in\mathbb{N}}$ be an ensemble of pairs of sequences of sets over $[k_n]$. We say that a nuPPT M is a $\{(\boldsymbol{S}_n, \boldsymbol{T}_n)\}_{n\in\mathbb{N}}$-*respecting message sampler *(or* valid w.r.t. $\{(\boldsymbol{S}_n, \boldsymbol{T}_n)\}_{n\in\mathbb{N}}$*) if*

- *M on input 1^n and a public parameter pp computes the ring R associated with pp and next based on only 1^n and R generates and outputs a pair $(\boldsymbol{m_0}, \boldsymbol{m_1})$ of sequences of $|S_n|$ ring elements and a sequence $\boldsymbol{z}$ of $|T_n|$ ring elements;*
- *There exists a polynomial $Q(\cdot,\cdot)$ such that for every $n \in \mathbb{N}$, every $(\mathsf{sp}, \mathsf{pp})$ in the support of $\mathsf{InstGen}(1^n, 1^{k_n})$, every $(\boldsymbol{S}, \boldsymbol{T})$-respecting arithmetic circuit C, there exists a constant $c \in \{0,1\}$ such that for any $b \in \{0,1\}$,*

$$Pr[(\boldsymbol{m_0}, \boldsymbol{m_1}, \boldsymbol{z}) \leftarrow M(1^n, \mathsf{pp}) : \mathsf{isZero}(C(\boldsymbol{m_b}, \boldsymbol{z})) = c] \geq 1 - Q(n, k_n)/|R|.$$

Let us comment that Definition 4 allows the message sampler M to select $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$ based on the ring $R = \mathbb{Z}_p$ (or else we could not pick a uniform element in the ring). On the other hand, to make the notion of valid message samplers as restrictive as possible, we prevent the message selection from depending on pp in any other way.

We can now define what it means for a graded encoding scheme to be semantically secure. Roughly speaking, we require that encodings of $(\boldsymbol{m}_0, \boldsymbol{z})$ and $(\boldsymbol{m}_1, \boldsymbol{z})$ under the sets $(\boldsymbol{S}, \boldsymbol{T})$ are indistinguishable as long as $(\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z})$ is sampled by a message sampler that is valid w.r.t. $(\boldsymbol{S}, \boldsymbol{T})$.

Definition 5 (Semantic Security) *Let $\mathcal{E}$ be a graded encoding scheme and $q(\cdot)$ and $c(\cdot)$ be polynomials. We say a graded encoding scheme $\mathcal{E}$ is (c, q)-*semantically secure *if for every polynomial $k(\cdot)$, every ensemble $\{(\boldsymbol{S}_n, \boldsymbol{T_n})\}_{n\in\mathbb{N}}$*

[11] For ease of notation, we assume that the description of a set S also contains a description of the universe set $[k]$.

where $\boldsymbol{S}_n$ and $\boldsymbol{T}_n$ are sequences of subsets of $[k(n)]$ of length $c(k(n)))$ and $q(k(n))$ respectively, for every $\{(\boldsymbol{S}_n, \boldsymbol{T}_n)\}_{n\in\mathbb{N}}$-respecting message sampler M and every nuPPT adversary A, there exists a negligible function ϵ such that for every security parameter $n \in \mathbb{N}$,

$$|Pr[\boldsymbol{Output}_0(1^n) = 1] - Pr[\boldsymbol{Output}_1(1^n) = 1]| \leq \epsilon(n)$$

where $\boldsymbol{Output}_b(1^n)$ is A's output in the following game:

- *Let $(\mathsf{sp}, \mathsf{pp}) \leftarrow \mathsf{InstGen}(1^n, 1^{k(n)})$.*
- *Let $\boldsymbol{m_0}, \boldsymbol{m_1}, \boldsymbol{z} \leftarrow M(1^n, \mathsf{pp})$.*
- *Let $\boldsymbol{u_b} \leftarrow \{\mathsf{Enc}(\mathsf{sp}, \boldsymbol{m_0}[i], \boldsymbol{S}_n[i])\}_{i=1}^{c(k_n)}, \{\mathsf{Enc}(\mathsf{sp}, \boldsymbol{z}[i], \boldsymbol{T}_n[i])\}_{i=1}^{q(k(n))}$.*
- *Finally, run $A(1^n, \mathsf{pp}, \boldsymbol{u_b})$.*

We say that $\mathcal{E}$ is (constant-message) semantically secure *if it is $(O(1), O(k))$-semantically secure; we say that $\mathcal{E}$* multi-message semantically secure *if it is $(O(k), O(k))$-semantically secure. We additionally say that $\mathcal{E}$ is* subexponentially-hard semantically secure *if there exists some exists some constant $\alpha > 0$ such that for every nuPPT A the above indistinguishability gap is bounded by $\varepsilon(n) = 2^{-O(n^\alpha)}$.*

In analogy with the GDDH assumption [GGH13a], our notion of semantical security restricts to the case when the number of elements encoded is $O(k)$.[12] Let us end this section by remarking that (sub-exponentially hard) semantical security trivially holds against polynomial-time "generic" attackers that are restricted to "legally" operating on the encodings—in fact, it holds even against *unbounded* generic attackers that are restricted to only making polynomially (or even subexponentially) many zero-test queries: recall that each legal zero-test query is constant with overwhelming probability (whether we operate on $\boldsymbol{m}_0, \boldsymbol{z}$ or $\boldsymbol{m}_1, \boldsymbol{z}$) and thus by a Union Bound, the output of any generic attacker restricted to polynomially many zero-test queries is also constant with overwhelming probability.

Semantical Security w.r.t. Restricted Classes of Message Samplers For our specific construction of indistinguishability obfuscators it suffices to assume the existence of *semantically secure encodings w.r.t. restricted classes of message samplers M*, where the $\{(\boldsymbol{S}_n, \boldsymbol{T}_n)\}_{n\in\mathbb{N}}$-respecting condition on M is replaced by some stronger restriction on M. It particular, it suffices to restrict to message samplers M that induce a *high-entropy*[13] distribution over $\boldsymbol{m}_0, \boldsymbol{m}_1, \boldsymbol{z}$—not only

[12] This restriction was suggested in [BCKP14] and independently by Hoeteck Wee; our original formulation of semantical security considered an unbounded polynomial number of elements in $\boldsymbol{z}$ (but our proof of security only relied on security for $O(k)$ elements). We now refer to the unbounded notion as *unbounded semantical security*, but it will not be needed for any of our results.

[13] Technically, by high-entropy, we here mean that the min-entropy is at least $\log |R| - O(\log\log |R|)$ where R is the ring associated with the encodings; that is, the min-entropy is "almost" optimal (i.e., $\log |R|$).

the individual elements have high min-entropy but also any element computed by applying a "non-terminal" sequence of legal arithmetic operations to $\boldsymbol{m_b}, \boldsymbol{z}$ (for $b \in \{0,1\}$); we refer to schemes satisfying this weaker notion of semantical security *entropic semantically secure* (and refer the reader to the full version for a formal definition).

3 Proof Overview

We here provide an overview of our obfuscator and its proof of security, and refer the reader to the full version [PST13] for further details.

The Basic Obfuscator We start by providing a construction of a "basic" obfuscator; our final construction will then rely on the basic obfuscator as a black-box. The construction of this obfuscator closely follows the design principles laid out in the original work by Garg et al [GGH+13b] and follow-up constructions [BR14,BGK+13] (in fact, the basic obfuscator may be viewed as a simplified version of the obfuscator from [BGK+13]). As these works, we proceeds in three steps:

- We view the NC^1 circuit to be obfuscated as a *branching program* BP (using Barrington's Theorem [Bar86])—that is, the program is described by m pairs of matrices $(B_{i,0}, B_{i,1})$, each one labelled with an input bit $\mathsf{inp}(i)$. The program is evaluated as follows: for each $i \in [m]$, we choose one of the two matrices $(B_{i,0}, B_{i,1})$, based on the input. Next, we compute the product of the chosen matrices, and based on the product determine the output—there is a unique "accept" (i.e., output 1) matrix, and a unique "reject" (i.e., output 0) matrix.
- The branching program BP is *randomized* using Kilian's technique [Kil88] (roughly, each pair of matrices is appropriately multiplied with the same random matrix R while ensuring that the output is the same), and then "randomized" some more—each individual matrix is multiplied by a random *scalar* α. Let us refer to this step as Rand.
- Finally the randomized matrices are encoded using multilinear encodings with the sets selected appropriately. We here rely on a (simple version) of the *straddling set* idea of [BGK+13] to determine the sets. We refer to this step as Encode.

(The original construction as well as the subsequent works also consisted of several other steps, but for our purposes these will not be needed.) The obfuscated program is now evaluated by using the multilinear operations to evaluate the branching program and finally appropriately use the zero-test to determine the output of the program.

Roughly speaking, the idea behind the basic obfuscator is that the multilinear encodings *intuitively* ensure that any attacker getting the encoding needs to multiply matrices along paths that corresponds to some input to the branching program (the straddling sets are used to ensure that the input is used consistently in the evaluation)[14]; the scalars α, roughly speaking, ensure that a potential attacker without loss of generality can use a *single* "multiplication-path" and still succeed with roughly the same probability, and finally, Kilian's randomization steps ensures that if an attacker *only* operates on matrices along a single path that corresponds to some input x (in a consistent way), then its output can be perfectly simulated given just the output of the circuit on input x. (The final step relies on the fact that the output of the circuit uniquely determines product of the branching program along the path, and Kilian's randomization then ensures that the matrices along the path are random conditioned on the product being this unique value.) Thus, if an attacker can tell apart obfuscations of two programs BP_0, BP_1, there must exist some input on which they produce different outputs. The above intuitions can indeed be formalized w.r.t. *generic attackers* (that only operate on the encodings in a legal way, respecting the set restrictions), relying on arguments from [BR14,BGK$^+$13]. This already suffices to prove that the basic obfuscator is an indistinguishability obfuscator assuming the encodings are *multi-message* semantically secure.[15]

The Merge Procedure To base security on the weaker assumption of (constant-message) semantical security, we will add an additional program transformation steps before the Rand and Encode steps. Roughly speaking, we would like to have a method $\mathsf{Merge}(BP_0, BP_1, b)$ that "merges" BP_0 and BP_1 into a single branching program that evaluates BP_b; additionally, we require that $\mathsf{Merge}(BP_0, BP_1, 0)$ and $\mathsf{Merge}(BP_0, BP_1, 1)$ only differ in a constant number of matrices. We achieve this merge procedure by connecting together BP_0, BP_1 into a branching program of double width and adding two "switch" matrices in the beginning and the end, determining if we should go "up" or "down". Thus, to switch between $\mathsf{Merge}(BP_0, BP_1, 0)$ (which is functionally equivalent to BP_0) and $\mathsf{Merge}(BP_0, BP_1, 1)$ (which is functionally equivalent to BP_1) we just need to switch the "switch matrices". More precisely, given branching programs BP_0 and BP_1 described respectively by pairs of matrices $\{(B^0_{i,0}, B^0_{i,1}),\ (B^1_{i,0}, B^1_{i,1})\}_{i\in[m]}$, we construct a merged program $\mathsf{Merge}(BP_0, BP_1, b)$ described by $\{(\hat{B}^0_{i,0}, \hat{B}^0_{i,1})\}_{i\in[m+2]}$ such that

$$\hat{B}^0_{i,b} = \hat{B}^1_{i,b} = \begin{pmatrix} B^0_{(i-1),b} & 0 \\ 0 & B^1_{(i-1),b} \end{pmatrix} \quad \text{for all } 2 \le i \le m+1 \text{ and } b \in \{0,1\}$$

[14] The encodings, however, still permit an attacker to add elements within matrices.

[15] As mentioned above, there are still some minor subtleties involved in doing this: the analyses of [BR14,BGK$^+$13] implicitly show that all *polynomial-size* legal arithmetic circuits are constant with overwhelming probability, but by slightly tweaking the constructions and the analyses to ensure a "perfect" simulation property, we can extend these arguments to hold against *all* (arbitrary-size) legal arithmetic circuits and thus base security on multi-message semantical security.

and the first and last matrices are given by:

$$\hat{B}^0_{1,b} = \hat{B}^0_{m+2,b} = I_{2w\times 2w} \qquad \text{for } b \in \{0,1\}$$
$$\hat{B}^1_{1,b} = \hat{B}^1_{m+2,b} = \begin{pmatrix} 0 & I_{w\times w} \\ I_{w\times w} & 0 \end{pmatrix} \qquad \text{for } b \in \{0,1\}$$

It directly follows from the construction that $\mathsf{Merge}(BP_0, BP_1, 0)$ and $\mathsf{Merge}(BP_0, BP_1, 1)$ differ only in the first and the last matrices (i.e., the "switch" matrices). Furthermore, it is not hard to see that $\mathsf{Merge}(BP_0, BP_1, b)$ is functionally equivalent to BP_b.

Our candidate obfuscator is now defined as $i\mathcal{O}(B) = \mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP, I, 0)))$, where I is simply a "dummy" program of the same size as BP.[16]

The idea behind the merge procedure is that to prove that obfuscations of two programs BP_0, BP_1 are indistinguishable, we can come up with a sequence of hybrid experiments that start with $i\mathcal{O}(BP_0)$ and end with $i\mathcal{O}(BP_1)$, but between any two hybrids only changes a constant number of encodings, and thus we may rely on semantic security of multilinear encodings to formalize the above intuitions. At a high level, our strategy will be to matrix-by-matrix, replace the dummy branching program in the obfuscation of BP_0 with the branching program for BP_1. Once the entire dummy branching program has been replaced by BP_1, we flip the "switch" so that the composite branching program now computes the branching program for BP_1. We then replace the branching program for BP_0 with BP_1, matrix by matrix, so that we have two copies of the branching program for BP_1. We now flip the "switch" again, and finally restore the dummy branching program, so that we end up with one copy of BP_1 and one copy of the dummy, which is now a valid obfuscation of BP_1. In this way, we transition from an obfuscation of BP_0 to an obfuscation of BP_1, while only changing a small piece of the obfuscation in each step. (On a very high-level, this approach is somewhat reminiscient of the Naor-Yung "two-key" approach in the context of CCA security [NY90] and the "two-key" bootstrapping result for indistinguishability obfuscation due to Garg et al [GGH$^+$13b]—in all these approaches the length of the scheme is artificially doubled to facilitate a hybrid argument. It is perhaps even more reminiscient of the Feige-Shamir "trapdoor witness" approach for constructing zero-knowledge arguments [FS90], whereby an additional "dummy" trapdoor witness is introduced in the construction to enable the security proof.)

More precisely, consider the following sequence of hybrids.

- We start off with $i\mathcal{O}(BP_0) = \mathsf{Enc}(\mathsf{Rand}(\mathsf{Merge}(BP_0, I, 0)))$
- We consider a sequence of hybrids where we gradually change the dummy program I to become BP_1; that is, we consider $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_0, BP', 0)))$, where BP' is "step-wise" being populated with elements from BP_1.
- We reach $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_0, BP_1, 0)))$.

[16] This description oversimplifies a bit. Formally, the Rand step needs to depends on the field size used in the Encode steps, and thus in our formal treatment we combine these two steps together.

- We turn the “switch” : $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_0, BP_1, 1)))$.
- We consider a sequence of hybrids where we gradually change the BP_0 to become BP_1; that is, we consider $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP', BP_1, 1)))$, where BP' is “step-wise” being populated with elements from BP_1.
- We reach $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_1, BP_1, 1)))$.
- We turn the “switch” back: $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_1, BP_1, 0)))$.
- We consider a sequence of hybrids where we gradually change the second BP_1 to become I; that is, we consider $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_1, BP', 0)))$, where BP' is “step-wise” being populated with elements from I.
- We reach $\mathsf{Encode}(\mathsf{Rand}(\mathsf{Merge}(BP_1, I, 0))) = i\mathcal{O}(BP_1)$.

By construction we have that if BP_0 and BP_1 are functionally equivalent, then so will all the hybrid programs–the key point is that we only “morph” between two branching programs on the “inactive” part of the merged branching program. Furthermore, by construction, between any two hybrids we only change a constant number of elements. Thus, if some distinguisher can tell apart $i\mathcal{O}(BP_0)$ and $i\mathcal{O}(BP_1)$, it must be able to tell apart two consecutive hybrids. But, by semantic security it then follows that some “legal” arithmetic circuit can tell apart the encodings in the two hybrids. Roughly speaking, we can now rely on simulation security of the basic obfuscator w.r.t. to just *legal* arithmetic circuits to complete the argument. A bit more precisely, based on BP_0, BP_1 and the hybrid index i, we can define a message sampler M_{i,BP_0,BP_1} that is valid (by the simulation arguments in [BGK+13]) as long as BP_0 is functionally equivalent to BP_1, yet our distinguisher manages to distinguish messages sampled from M_{i,BP_0,BP_1}, contradicting semantical security.

Dealing with branching programs with non-unique outputs There is a catch with the final step though. Recall that to rely on Kilian’s simulation argument it was crucial that there are *unique* accept and reject matrices. For our “merged” programs, this is no longer the case (the output matrix is also a function of the second “dummy” program), and thus it is no longer clear how to prove that the message distribution above is valid. We overcome this issue by noting that the *first column* of the output matrix actually is unique, and this is all we need to determine the output of the branching program; we refer to such branching programs as *fixed output-column branching programs.* Consequently it suffices to release encodings of the *just* first column (as opposed to the whole matrices) of the last matrix pair in the branching program, and we can still determine the output of the branching program. As we show, for such a modified scheme, we can also simulate the (randomized) matrices along an “input-path” given just the first column of the output matrix.

A Modular Analysis: Neighboring-Matrix $i\mathcal{O}$ In the actual proof, we provide a modular analysis of the above two steps (that may be interesting in its own right).

- We define a notion of *neighboring-matrix indistinguishability obfuscation*, which relaxes indistinguishability obfuscation by only requiring security to hold w.r.t. any two functionally equivalent branching programs that differ in at most a constant number of matrices.

- We then use the above merge procedure (and the above hybrid argument) to show that the existence of a neighboring-matrix $i\mathcal{O}$ for all "fixed output column" branching programs implies the existence of a "full-fledged" $i\mathcal{O}$.
- We finally use the "basic obfuscator" construction to show how to construct a neighboring-matrix $i\mathcal{O}$ for all fixed output column branching programs based on (constant-message) semantical security.

Basing Security on a (Single) Falsifiable Assumption To base security on a falsifiable assumption, we rely on a different merge procedure from the work of Boyle, Chung and Pass [BCP14]: Given two NC^1 circuits C_0, C_1 taking (at most) n-bit inputs, and a string z, let $\widehat{\mathsf{Merge}}(C_0, C_1, z)$ be a circuit that on input x runs $C_0(x)$ if $x \geq z$ and $C_1(x)$ otherwise; in essence, this procedure lets us "traverse" between C_0 and C_1 while provably only changing the functionality on at most one input. ([BCP14] use this type of merged circuits to perform a binary search and prove that indistinguishability obfuscation implies differing-input obfuscation for circuits that differ in only polynomially many inputs.) We now define a notion of *neighboring-input* $i\mathcal{O}$, which relaxes $i\mathcal{O}$ by only requiring that security holds with respect to "neigboring-input" programs $\widehat{\mathsf{Merge}}(C_0, C_1, z)$, $\widehat{\mathsf{Merge}}(C_0, C_1, z+1)$ that are functionally equivalent. Note that checking whether $\widehat{\mathsf{Merge}}(C_0, C_1, z)$, $\widehat{\mathsf{Merge}}(C_0, C_1, z+1)$ are functionally equivalent is easy: they are equivalent iff $C_0(z) = C_1(z)$. (As such, the assumption that a scheme satisfies neighboring-input $i\mathcal{O}$ is already an efficiently falsfiable assumption.) Furthermore, by a simple hybrid argument over $z \in \{0,1\}^n$, *exponentially-secure* neighboring-input $i\mathcal{O}$ implies "full" $i\mathcal{O}$—exponential security is needed since we have 2^n hybrids. (We mention a very recent work by Gentry, Lewko and Waters [GLW14] in the context of *witness encryption* [GGSW13] that similarly defines a falsifiable primitive "positional witness encryption" that implies the full-fledged notion with an exponential security loss.)

Additionally, note that to show that our construction satisfies exponentially-secure neighboring-input $i\mathcal{O}$, we only need to rely on exponentially-secure semantical security w.r.t. classes of message distributions corresponding to programs of the form $\widehat{\mathsf{Merge}}(C_0, C_1, z)$, $\widehat{\mathsf{Merge}}(C_0, C_1, z+1)$. Equivalently, it suffices to rely on exponentially-secure semantical security w.r.t. a *single* distribution over sets and message samplers corresponding to uniformly selected z and programs C_0, C_1 (again, this only results in an exponential security loss). Finally, by padding the security parameter of the multilinear encodings in the construction, it actually suffices to rely on subexponential security.

4 Acknowledgments

We are very grateful to Benny Applebaum, Omer Paneth, Ran Canetti, Kai-Min Chung, Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, abhi shelat, Hoeteck Wee and Daniel Wichs for many helpful comments. We are especially gratefeul to Shai for pointing out the connection between semantical security for multilinear encodings and the "uber" assumption for bilinear maps of [BBG05], and for several very useful conversations about multilinear encodings and the security of the [GGH13a] constructions, to Amit for several helpful conversations about the presentation of our results, and Benny for suggesting we make our proof more modular (which lead to the notion of neigboring-matrix branching programs). Finally thanks to the anonymous Crypto reviewers for their useful comments. Thanks so very much!

References

[Bar86] David A. Mix Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. In *STOC*, pages 1–5, 1986.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology EUROCRYPT 2005*, pages 440–456. 2005.

[BCKP14] Nir Bitansky, Ran Canetti, Yael Kalai, and Omer Paneth. Virtual-grey-box obfuscation from general circuits. In *Advances in Cryptology CRYPTO 2014*, 2014.

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

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In *STOC 2014*, 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.

[BGK$^+$13] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. Cryptology ePrint Archive, Report 2013/631, 2013.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *TCC*, pages 1–25, 2014.

[BS03] Dan Boneh and Alice Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324(1):71–90, 2003.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.

[BZ14] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in Cryptology CRYPTO 2014*, 2014.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology, CRYPTO 2013*, pages 476–493, 2013.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.

[Gen09] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[GGG^{+}14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *EUROCRYPT*, pages 578–602, 2014.

[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.

[GGHR14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. In *TCC*, pages 74–94, 2014.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, STOC '13, pages 467–476, 2013.

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

[GR07] Shafi Goldwasser and Guy Rothblum. On best-possible obfuscation. In *Theory of Cryptography*, volume 4392, pages 194–213. 2007.

[Had00] Satoshi Hada. Zero-knowledge and code obfuscation. In *Advances in Cryptology–ASIACRYPT 2000*, pages 443–457. Springer, 2000.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. In *EUROCRYPT*, pages 201–220, 2014.

[Kil88] Joe Kilian. Founding crytpography on oblivious transfer. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 20–31. ACM, 1988.

[KMN^{+}14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. Cryptology ePrint Archive, Report 2014/347, 2014. `http://eprint.iacr.org/`.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for np from indistinguishability obfuscation. *CoRR*, abs/1403.5698, 2014.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 427–437, 1990.

[PST13] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. Cryptology ePrint Archive, Report 2013/781, 2013.

[RAD78] R L Rivest, L Adleman, and M L Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179, 1978.

[Rot13] Ron D Rothblum. On the circular security of bit-encryption. In *Theory of Cryptography*, pages 579–598. Springer, 2013.

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