On Virtual Grey Box Obfuscation for General Circuits

Nir Bitansky* Ran Canetti† Yael Tauman Kalai‡ Omer Paneth§

July 15, 2014

Abstract

An obfuscator $\mathcal{O}$ is Virtual Grey Box (VGB) for a class $\mathcal{C}$ of circuits if, for any $C \in \mathcal{C}$ and any predicate π, deducing $\pi(C)$ given $\mathcal{O}(C)$ is tantamount to deducing $\pi(C)$ given unbounded computational resources and polynomially many oracle queries to C. VGB obfuscation is often significantly more meaningful than indistinguishability obfuscation (IO). In fact, for some circuit families of interest VGB is equivalent to full-fledged Virtual Black Box obfuscation.

We investigate the feasibility of obtaining VGB obfuscation for general circuits. We first formulate a natural strengthening of IO, called *strong IO* (SIO). Essentially, $\mathcal{O}$ is SIO for class $\mathcal{C}$ if $\mathcal{O}(C) \approx \mathcal{O}(C')$ whenever the pair (C, C') is taken from a distribution over $\mathcal{C}$ where, for all x, $C(x) \neq C'(x)$ only with negligible probability.

We then show that an obfuscator is VGB for a class $\mathcal{C}$ if and only if it is SIO for $\mathcal{C}$. This result is unconditional and holds for any $\mathcal{C}$. We also show that, for some circuit collections, SIO implies virtual black-box obfuscation.

Finally, we formulate a slightly stronger variant of the semantic security property of graded encoding schemes [Pass-Seth-Telang Crypto 14], and show that existing obfuscators, such as the obfuscator of Barak et al. [Eurocrypt 14], are SIO for all circuits in NC^1, assuming that the underlying graded encoding scheme satisfies our variant of semantic security.

Put together, we obtain VGB obfuscation for all NC^1 circuits under assumptions that are almost the same as those used by Pass et al. to obtain IO for NC^1 circuits. We also show that semantic security is in essence *necessary* for showing VGB obfuscation.

*Tel Aviv University. Email: nirbitan@tau.ac.il. Supported by an IBM Ph.D. Fellowship, the Check Point Institute for Information Security, and The Israeli Ministry of Science and Technology.

†Boston University and Tel Aviv University. Email: canetti@bu.edu. Supported by the Check Point Institute for Information Security, an ISF grant 20006317, an NSF EAGER grant, and an NSF Algorithmic foundations grant 1218461.

‡Microsoft Research

§Boston University. Email: omer@bu.edu. Supported by the Simons award for graduate students in theoretical computer science and an NSF Algorithmic foundations grant 1218461.

Contents

1 Introduction

Program obfuscation, namely the ability to efficiently compile a given program into a functionally equivalent program that is "unintelligible", is an intriguing concept. Indeed, much effort has been devoted to understanding this concept from the definitional aspect, the algorithmic aspect, and the applications aspect. Here let us concentrate on the first two aspects.

Starting with the works of Hada [Had00] and Barak et al. [BGI+01], a number of measures of security for program obfuscation have been proposed. Let us briefly review three notions of interest. The first, *virtual black box (VBB)* obfuscation [BGI+01], requires that having access to the obfuscated program is essentially the same as having access to the program only as black box. Concretely, focusing on programs represented as circuits, an obfuscator $\mathcal{O}$ for a family of circuits is worst-case VBB if for any poly-time adversary $\mathcal{A}$, there exists a poly-time simulator S, such that for any circuit C from the family, and any predicate $\pi(\cdot)$, $\mathcal{A}$ cannot learn $\pi(C)$ from $\mathcal{O}(C)$ with noticeably higher probability than $\mathcal{S}$ can, given only oracle access to C. The obfuscator $\mathcal{O}$ is average-case VBB if the above is only required to hold for circuits C that are sampled at random from the family.

While this VBB obfuscation is natural and strong, Barak et al. [BGI+01] showed that this definition, and variants thereof, are unobtainable in general by demonstrating a family of *unobfuscatable functions:* these are functions f where any circuit computing the function inherently leaks secrets that are infeasible to compute given only black box access to f. Moreover it turns out that, under cryptographic assumptions, if the simulator $\mathcal{S}$ is universal (or equivalently, works for any adversarial auxiliary input) then VBB obfuscation is unobtainable for *any* circuit family whose functionality has super-polynomial "pseudo entropy" [GK05, BCC+14].

A weaker variant of VBB, called *virtual grey-box (VGB)* [BC10], allows the simulator to be *semi-bounded,* namely it can be computationally unbounded, while still making only a polynomial number of queries to the circuit C. While significantly weaker than VBB in general, VGB is still meaningful for circuits that are unlearnable even by semi-bounded learners. Furthermore, VGB obfuscators for circuits escape the general impossibility results that apply to VBB obfuscators.

A weaker notion yet, called indistinguishability obfuscation (IO) [BGI+01], allows the (now computationally unbounded) simulator to also make an unbounded number of queries to C. Equivalently, $\mathcal{O}$ is an IO for a circuit collection if for any two circuits C_0 and C_1 in the collection, having the same size and functionality, $\mathcal{O}(C_0)$ and $\mathcal{O}(C_1)$ are indistinguishable.

While IO has some attractive properties, and some important cryptographic applications [GR07, SW13, GGH+13b], the security guarantees provided by IO are significantly weaker than those provided by either VBB or VGB obfuscation.

On the algorithmic level, for many years we had candidate obfuscators only for very simple functions such as point functions and variants. The landscape has changed completely with the recent breakthrough work of [GGH+13b], which proposed a candidate general-purpose obfuscation algorithm for all circuits. [GGH+13b] show that their scheme resists some simple attacks; but beyond that, they do not provide any analytic evidence for security.

Considerable efforts have been made to analyze the security of the [GGH+13b] obfuscator and variants. The difficulty appears to be in capturing the security properties required from the *graded encodings schemes* [GGH13a, CLT13], which is a central component in the construction. As a first step towards understanding the security of the [GGH+13b] obfuscator, [BR13, BGK+13] consider an ideal algebraic model, where the adversary is given "generic graded encodings" that can only be manipulated via admissible algebraic operations. They show that, in this model, variants of the [GGH+13b] scheme are VBB obfuscators for all poly-size circuits. (We remark that [CV13] construct a VBB general obfuscator with similar properties; however their abstract model is different and does not seem to correspond to any existing cryptographic primitive.)

Still, neither of these idealized constructions or their analyses have, in of themselves, any bearing on

the security of obfuscation algorithms in the plain model.

Pass et al. [PTS13] make the first step towards proving the security of a general obfuscation scheme based on some natural hardness assumption in the plain model. Specifically, they define a *semantic security* property for graded encoding schemes, which is aimed at capturing what it means for a graded encoding scheme to "behave essentially as an ideal multi-linear graded encoding oracle". They then show that a specially-crafted variant of the [BGK$^+$13] obfuscator, with the ideal graded encoding scheme replaced by a semantically-secure graded encoding scheme, is IO for all circuits.

But what about stronger security notions? In particular:

> *What are the strongest security guarantees for general obfuscation that can be based on natural cryptographic assumptions such as semantically-secure graded encoding?*

Our contributions. We obtain worst-case VGB obfuscation for NC^1, based on almost the same assumptions as those used in [PTS13] to show IO for NC^1. As an intermediate step towards this goal, we put forth a somewhat stronger variant of indistinguishability obfuscation, called *strong IO* (SIO). Informally, an obfuscator $\mathcal{O}$ is SIO for a class of circuits $\mathcal{C}$ if $\mathcal{O}(C) \approx \mathcal{O}(C')$ not only when $C, C' \in \mathcal{C}$ have the same functionality, but also when C and C' come from distributions over circuits in $\mathcal{C}$ that are "close together", in the sense that for any given input x, the probability that $C(x) \neq C'(x)$ is negligible. An alternative view of the definition (which turns out to be equivalent) is that if no semi-bounded adversary can distinguish oracle access to C from access to C'. We then show that:

1. SIO is in fact *equivalent* to worst-case VGB obfuscation. Furthermore, for certain classes of functions, such as point functions, hyperplanes, or fuzzy point functions, SIO is equivalent to full-fledged worst-case VBB obfuscation. These equivalences hold unconditionally.

2. Assuming existence of graded encoding schemes that satisfy a somewhat stronger variant of the semantic security notion of Pass et al. [PTS13], we show that known obfuscation schemes are SIO for all circuits in NC^1. More generally, we show that *any* obfuscator for a class of circuits $\mathcal{C}$ that is VBB in the ideal graded encoding model, is SIO in the plain model, when the ideal graded encoding oracle is replaced by a graded encoding scheme that satisfies a variant of the [PTS13] assumption.

We also give evidence for the *necessity* of semantically-secure graded encoding for obtaining VGB. Specifically we show that, assuming existence of VGB obfuscators for all circuits, there exist *mutlilinear jigsaw puzzles* satisfying a form of semantic security. Multilinear jisaw puzzles, defined in [GGH$^+$13b], are a limited-functionality variant of multilinear maps. They suffice for obtaining the positive result described in Item 2 above.

Finally, we investigate the plausibility of the semantic security assumption on graded encoding schemes, propose some relaxed variants, and show that our main results can be obtained under all these relaxations. Namely, we first give new evidence for the relative strength of the semantically-secure graded encodings assumption. Specifically, we show that semantically-secure graded encodings are subject to the following limitations:

1. *SAT lower bounds.* We show that semantically-secure graded encodings imply exponential circuit lower bounds for SAT. Such lower bounds are currently not known to follow from IO (even assuming P $\neq$ NP).

2. *A generic attack.* We present an attack showing that any graded encoding scheme with certain efficiency properties cannot satisfy semantic security. While the attack does not apply to currently known candidate graded encodings [GGH13a, CLT13], it does point out potential limitations of this notion. We complement this observation by suggesting a natural relaxation of semantic security called *bounded semantic-security* that bypasses this attack. Our main results can be obtained also under this relaxed assumption.

In addition to the above relaxation, we consider several other relaxations of semantic-security, and investigate their relations. We show that our main results can be obtained under all these relaxations.

The rest of the introduction provides a more detailed overview of our results. Section 1.1 presents the implication from SIO to VGB and VBB obfuscation. Section 1.2 provides background on graded encoding schemes and the semantic security assumption. Sections 1.3 presents the construction of SIO from semantically-secure graded encoding schemes, and Section 1.4 describes additional results on the viability of the semantic security assumption of graded encoding schemes, and relations among various variants.

1.1 From SIO to VGB and VBB Obfuscation

We first define SIO a bit more precisely. A distribution $\tilde{C}$ over circuits is said to be *ν-concentrated* around a boolean function f if for any value x in the domain of f we have that $\Pr[\tilde{C}(x) \neq f(x)] \leq \nu$. We say that $\tilde{C}$ is simply *concentrated* if it is ν-concentrated for some negligible function ν. An obfuscator $\mathcal{O}$ is SIO for a class $\mathcal{C}$ of circuits if for any two (not necessarily efficiently samplable) distributions $\tilde{C}, \tilde{C}'$ over circuits in $\mathcal{C}$ that are concentrated around the same function, it holds that $\mathcal{O}(\tilde{C})$ and $\mathcal{O}(\tilde{C}')$ are computationally indistinguishable. We show the following.

Theorem 1.1. *[informal] An obfuscator is SIO for a class $\mathcal{C}$ of circuits if and only if it is worst-case VGB obfuscator for $\mathcal{C}$.*

Theorem 1.1 motivates the study of SIO as an independent notion, beyond its role in the construction of worst-case VGB obfuscation for NC^1 from semantically-secure graded encodings. Future results obtaining SIO for more functions, or based on different assumptions, will directly apply for VGB obfuscation as well. We note that existing candidate indistinguishability obfuscators *for all circuits* [GGH$^+$13b, BR13, BGK$^+$13], may also be considered as candidates for SIO, and thus also for VGB obfuscation, for all circuits.

Ideas behind the proof of Theorem 1.1. Showing that VGB implies SIO is straightforward. In the other direction we construct an (inefficient) simulator $\mathcal{S}$ for any adversary $\mathcal{A}$. Recall that, for *any* circuit $C \in \mathcal{C}$ in the given collection $\mathcal{C}$, the simulator $\mathcal{S}$ should simulate what $\mathcal{A}$ learns from an obfuscation $\mathcal{O}(C)$, given only oracle access to C. The high level idea is as follows: $\mathcal{S}$ will use its oracle to C to gradually reduce the set $\mathcal{K}$ of candidates for the circuit C, starting from $\mathcal{K}_0 = \mathcal{C}$, and continuing with progressively smaller sets of candidates:

$$\mathcal{K}_i \subsetneq \mathcal{K}_{i-1} \subsetneq \cdots \subsetneq \mathcal{K}_0 = \mathcal{C} \ .$$

$\mathcal{S}$ will continue this process until it obtains a set $\mathcal{K}^*$ where $\mathcal{A}$ cannot distinguish an obfuscation $\mathcal{O}(C)$ of C from an obfuscation $\mathcal{O}(C')$ of a random circuit C' in $\mathcal{K}^*$.

To carry out this plan, $\mathcal{S}^C$ iteratively performs two main steps: *concentration*, and *majority separation*. In the concentration step $\mathcal{S}$ tries to learn C in a straightforward way: it queries C on a point x_i that splits the current set of candidate circuits $\mathcal{K}_i$ as evenly as possible. Based on the value of $C(x_i)$, $\mathcal{S}$ rules out some of the candidates. This process is repeated until there is no query that necessarily shrinks the set of candidates by a factor of at least $1 - \varepsilon$, where ε is a parameter of the simulation. ε is chosen such that $1/\varepsilon$ is a polynomial, depending only on $\mathcal{A}$ and on the required simulation accuracy. Note that at the end of the concentration step, $\mathcal{S}$ must reach a set of candidates $\mathcal{K}_j$ that is ε-concentrated. This occurs after at most $\log |\mathcal{C}|/\varepsilon$ queries. The concentration step alone essentially suffices to ensure *average-case* VGB simulation; indeed, it follows from SIO security that if a circuit C is chosen at random from a concentrated set $\mathcal{K}_j$, $\mathcal{A}$ cannot compute any predicate $\pi(C)$, given $\mathcal{O}(C)$, better than it can given an obfuscation $\mathcal{O}(C')$ of an independent random $C' \leftarrow \mathcal{K}_j$.

However, the concentration step alone does not guarantee *worst-case* simulation. In particular, $\mathcal{A}$ may have some hardwired information that allows it to distinguish C from a random circuit in $\mathcal{K}_j$. In

this case, however, $\mathcal{S}$ can further reduce the set of candidates $\mathcal{K}_j$ by separating any such distinguishable circuit C from the majority of functions in $\mathcal{K}_j$. Concretely, we define the set $\mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$ of *distinguishable circuits* in $\mathcal{K}_j$, as those circuits C in $\mathcal{K}_j$ such that $\mathcal{A}$ can ε-distinguish between $\mathcal{O}(C)$ and $\mathcal{O}(C')$ for a random $C' \leftarrow \mathcal{K}_j$. We say that a point x *separates* a circuit C from the majority if $C(x) \neq \mathrm{maj}_{\mathcal{K}_j}(x)$ where $\mathrm{maj}_{\mathcal{K}_j}$ is the majority of all circuits in $\mathcal{K}_j$.

In the majority-separation step, the simulator will query its oracle C on a *small* set of roughly $\log|\mathcal{C}|/\varepsilon$ points $L_{\mathcal{K}_j}$ that separates all the distinguishable circuits in $\mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$ from the majority. This means that, if the oracle C agrees with $\mathrm{maj}_{\mathcal{K}_j}$ on all points $x \in \mathcal{L}_{\mathcal{K}_j}$, then $\mathcal{A}$ cannot tell apart $\mathcal{O}(C)$ form $\mathcal{O}(C')$ for a random $C' \leftarrow \mathcal{K}_j$, in which case, the simulation can be completed. Otherwise, if C disagrees with $\mathrm{maj}_{\mathcal{K}_j}$ on some point $x \in \mathcal{L}_{\mathcal{K}_j}$, $\mathcal{S}$ obtains a new set of candidates $\mathcal{K}_{j+1} \subsetneq \mathcal{K}_j$ which is necessarily smaller by a $1-\varepsilon$ factor, since $\mathcal{K}_j$ is ε-concentrated.

By iteratively applying the two steps we either reach some $\mathcal{K}^*$ for which $\mathcal{A}$ cannot distinguish $\mathcal{O}(C)$ from $\mathcal{O}(C')$ for a random $C' \leftarrow \mathcal{K}^*$, or we have completely exhausted the collection $\mathcal{C}$ and found exactly the circuit C. Since we reduce $\mathcal{K}_j$ at each step by a $1-\varepsilon$ factor, the process must end after at most $\log|\mathcal{C}|/\varepsilon$ steps, and at most $\mathrm{poly}(\log|\mathcal{C}|/\varepsilon)$ queries.

But how do we establish the existence of a small set $L_{\mathcal{K}_j}$ that separates $\mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$ from the majority in $\mathcal{K}_j$? Here we rely on the SIO security of $\mathcal{O}$. Specifically, SIO implies that any subset S of the distinguishable circuits $\mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$, cannot be ε-concentrated around $\mathrm{maj}_{\mathcal{K}_j}$, because $\mathcal{A}$ distinguishes $\mathcal{O}(C)$, for $C \leftarrow \mathcal{K}_j$ from $\mathcal{O}(C')$ for $C' \leftarrow S \subseteq \mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$.[1] Since no S as above is ε-concentrated around $\mathrm{maj}_{\mathcal{K}_j}$, we can separate all of the circuits in $\mathbb{D}_{\mathcal{A}}(\mathcal{K}_j)$ from $\mathrm{maj}_{\mathcal{K}_j}$ with at most $\mathrm{poly}(\log|\mathcal{C}|/\varepsilon)$ points, as required.

On the possibility of VBB obfuscation. The simulation strategy described above requires only a polynomial number of queries, however, the overall running time of the simulator may not be bounded in general. Indeed, in the concentration step, finding a point x_j that significantly splits $\mathcal{K}_j$ may require super-polynomial time. Also, in the majority-separation step, while the set $L_{\mathcal{K}_j}$ is small, computing it from $\mathcal{K}_j$ may also require super-polynomial time.

Nevertheless, we show that for certain classes of circuits, simulation can be done more efficiently, or even in polynomial time. Specifically, abstracting away from the above simulation process, we consider the notion of *learning via a majority-separation oracle*, where a given circuit C (or more generally a function) in a prescribed family is learned via oracle access to C and oracle access to the majority separation oracle $\mathbb{S}$, which takes as input (the description of a) a concentrated sub-family $\mathcal{K}$ that includes C and outputs a point x that separates C from the majority in $\mathcal{K}$.

While the strategy described above shows that any class of circuits can be learned with polynomially many queries to C and $\mathbb{S}$, the learner itself may be inefficient, which results in inefficient simulation. We show that a more efficient learning procedure can sometimes be translated into a more efficient simulation strategy, depending on pattern of queries made by the learning algorithm. We identify several function classes, for which such efficient learning is possible, yielding new feasibility results for worst-case VBB obfuscation. Examples include fuzzy point functions, conjunctions, and constant-dimension linear subspaces. We also get a unified proof for all existing worst-case VBB obfuscation results (such as point functions, constant-size set functions, and constant-dimension hyper-planes).

Connection to previous worst-case VBB/VGB obfuscators. The *majority separation technique* is rooted in the *sack of distinguishable points technique* of Canetti [Can97]. There, and in [Wee05], it was used to get worst-case (simulation-based) VBB obfuscation from indistinguishability-based obfuscation, for the simple case of point functions. The technique was then extended to VBB obfuscation of constant-dimension hyperplanes [CRV10] and VGB obfuscation of set functions [BC10]. The majority separation technique generalizes the above for arbitrary functions. Indeed, in the above works, the indistinguishability guarantee considered is equivalent to SIO (for the classes in question).

[1] We assume here (for simplicity and without loss of generality), that the distinguishing gap is always of the same sign.

Connection to VGB obfuscation of evasive functions. *Evasive* collections are function collections concentrated around the all-zero function. Barak et al. [BBC+14] show that average-case VGB obfuscation for *all* evasive collections implies *weak average-case* VGB for all collections.[2] Here "weak" means that the simulator is allowed to make a slightly super-polynomial number of queries.

We show that an obfuscator is SIO for a collection $\mathcal{C}$ of circuits if and only if it is IO for $\mathcal{C}$ and in addition, it is average-case VGB for any evasive sub-collection of $\mathcal{C}$. In particular, it follows that if an obfuscator is IO and average-case VGB for all evasive collections in $\mathcal{C}$, then it is a *worst-case* VGB obfuscator for $\mathcal{C}$. This is incomparable to the Barak et al. result: on the one hand, they do not need to assume that the obfuscator is IO; on the other hand they only show that it is average-case *weak* VGB, rather than worst-case standard VGB.

To better compare the two techniques, let us state the result we would get using our techniques, assuming only average-case VGB for all evasive collections, and without assuming IO (in particular, without assuming SIO). Roughly, we would get a weak kind of obfuscation where any adversary $\mathcal{A}$ has an $\mathcal{A}$*-designated* obfuscator $\mathcal{O}_{\mathcal{A}}$, which may be inefficient. The security guarantee is that $\mathcal{A}$ has a *worst-case* VGB simulator $\mathcal{S}$, so that for any circuit $C \in \mathcal{C}$, it holds that $\mathcal{A}(\mathcal{O}_{\mathcal{A}}(C)) \approx_{\varepsilon} \mathcal{A}(\mathcal{S}^{C})$; namely, $\mathcal{A}$ cannot tell an $\mathcal{A}$-designated obfuscation of C from a circuit sampled by the semi-bounded simulator, using only black-box access to C. The size of circuits output by $\mathcal{O}_{\mathcal{A}}(C)$ is a polynomial $p(|C|)$ that depends only on the class $\mathcal{C}$, but not on $\mathcal{A}$.

Assuming also IO allows us to "switch quantifiers", and show that there is *a single* efficient obfuscator $\mathcal{O}$ that works for *all* adversaries. This obfuscator would simply output an IO obfuscation of C (padded up to size $p(|C|)$). Security against all adversaries would then follow from the fact that IO is the "best-possible" obfuscator [BGI+01, GR07], and thus would achieve the same security as any adversary-designated obfuscator. [3]

1.2 Semantically-Secure Graded Encoding Schemes: Background

Before describing how we get SIO from semantically secure graded encoding schemes, we provide some background on the latter. A graded encoding scheme [GGH13a] consists of the following algorithms: InstGen that given a universe set $[k]$, outputs public parameters pp and secret parameters sp, where pp contains a description of a ring R; Encode that given sp, a set $S \subseteq [k]$ and $\alpha \in R$, generates an encoding $[\alpha]_S$; Add and Sub that, given encodings $[\alpha_1]_S$ and $[\alpha_2]_S$, generate encodings $[\alpha_1 + \alpha_2]_S$ and $[\alpha_1 - \alpha_2]_S$ respectively; Mult that, given encodings $[\alpha_1]_{S_1}$ and $[\alpha_2]_{S_2}$ such that $S_1 \cap S_2 = \emptyset$, generates an encoding $[\alpha_1 \cdot \alpha_2]_{S_1 \cup S_2}$; and isZero that given an encoding $[\alpha]_{[k]}$ outputs 1 if and only if $\alpha = 0$ (all the algorithms above also take as input pp).

[GGH13a, CLT13] consider standard versions of DDH-type security that can be conjectured to hold for their graded encoding schemes. Basing the security of obfuscation mechanisms on these assumptions seems at this point far out of reach, even if one considers only IO security. So which security properties of encoding schemes would suffice for this purpose? The high-level approach of Pass et al. [PTS13] is to devise a property that, not only hides "DDH-type relations" between encodings, but also any other relation that cannot be revealed using the admissible algebraic operations provided by the graded encoding interface. In other words, the encoding scheme should amount to an "ideal encoding scheme", where encodings are truly accessed only through admissible algebraic operations. This may, in particular, allow leveraging the existing proofs of VBB security in the ideal graded encoding model [BR13, BGK+13].

More specifically, Pass et al. take the following approach (described first in an oversimplified manner). Consider a *message sampler* $\mathbb{M}([k], R)$ that samples a tuple $(S_1, m_1), \ldots, (S_\ell, m_\ell)$ from one of two distributions $\mathcal{D}_0$ or $\mathcal{D}_1$, where each $S_i \subseteq [k]$, each $m_i \in R$, and ℓ is polynomial in the security

[2] In fact, for concentrated, and in particular evasive, collections, average-case VGB and average-case VBB are equivalent.

[3] We note that, in the body, our actual proof relies directly on SIO, which we show to follow from average-case VGB for evasive collections and standard IO.

parameter. We say that the sampler is *admissible* if no polynomially-bounded "algebraic adversary" that is given $\vec{S} = (S_1, \ldots, S_\ell)$, and can access the ring elements $\vec{m} = (m_1, \ldots, m_\ell)$ only via *an ideal encoding oracle*, is able to tell whether $(\vec{S}, \vec{m})$ were taken from $\mathcal{D}_0$ or $\mathcal{D}_1$. The ideal encoding oracle only allows the same algebraic manipulations allowed by the graded encoding interface, or put abstractly, it allows the adversary to choose any arithmetic circuit C that respects the set structure given by $\vec{S}$, and test whether $C(\vec{m}) = 0$. The requirement is that, for such an admissible sampler, an efficient adversary that obtains actual encodings $\left\{[m_i]_{S_i} : i \in [\ell]\right\}$, along with the corresponding public parameters pp, also cannot tell whether $(\vec{S}, \vec{m})$ was sampled from $\mathcal{D}_0$ or $\mathcal{D}_1$.

As noticed by Pass et al., the assumption formulated above is actually false—it is susceptible to a diagonalization attack in the spirit of the [BGI$^+$01] impossibility result for general VBB obfuscation. To get around this caveat, Pass et al. strengthen the admissibility requirement to require that $\mathcal{D}_0, \mathcal{D}_1$ are indistinguishable even to a *semi-bounded* algebraic adversary, namely an algebraic adversary that is computationally unbounded, but makes only a polynomial number of queries to the ideal graded encoding oracle. Furthermore, even this relaxed assumption suffices for obtaining IO in the plain model.

To get IO in the plain model, the idea is to rely on a construction of VBB obfuscation for NC^1 in the ideal graded encoding model [BR13, BGK$^+$13] and replace the ideal graded encoding with a concrete graded encoding scheme satisfying semantic security as stated above. To show that obfuscations of two equivalent circuits C_0, C_1 are indeed indistinguishable, consider a message sampler $\mathbb{M}$ that samples a pair of distributions $\mathcal{D}_0, \mathcal{D}_1$ such that the distribution $\mathcal{D}_i$ is an obfuscation of the circuit C_i in the ideal graded encoding model. The admissibility of this sampler follows from the fact that the [BGK$^+$13] obfuscator is secure even against *semi-bounded* adversaries. Specifically, an algebraic adversary accessing $\mathcal{D}_i$ via an ideal encoding oracle essentially has black-box access to the circuit C_i. Since C_0, C_1 cannot be distinguished given only black-box access, their obfuscations are indistinguishable as well. The eventual Pass et al. assumption is further relaxed in several ways, while still yielding their main application to IO.

1.3 SIO from Semantically-Secure Graded Encoding, and Back Again

We sketch our variant of the semantic security assumption, and explain how we obtain SIO for NC^1 circuits from this variant. We also give evidence for the *necessity* of semantic security for obtaining SIO.

Essentially, the reason that semantic security of graded encoding schemes implies SIO is that semantic security considers *any* admissible distributions over encodings, not only ones that come out of obfuscating a given program. In particular, distributions that consist of ideal-graded-encoding-VBB obfuscations of circuits that are concentrated around the same function are admissible, thus their instantiations via a semantically secure graded encoding scheme are guaranteed to be indistinguishable.

However, some care has to be taken here: note that SIO considers even distributions that are not necessarily efficiently samplable. (Indeed, this property is crucial in the proof that SIO implies worst-case VGB.) This means that we will need to somewhat modify the formulation of the semantic security assumption.

A naive attempt to formalize this variant of semantic security may simply allow the sampler to be computationally unbounded. However, recall that the message sampler is given the description of a ring R. (This is required in order to sample obfuscations in the ideal graded encoding model that consist, for example, of random elements in R.) A computationally unbounded sampler that sees R may be able to recover information that compromises the security of the encodings (for example, the secret parameters). The sampler can produce encodings that reveal this secret information. Note that such a sampler may still be admissible since learning the secret parameters gives no advantage to an algebraic adversary.

Instead we sample messages in two stages: first, an unbounded sampler S generates a poly-size

auxiliary input string s; second, an *efficient* encoder $\mathbb{M}$ gets the ring R and the auxiliary input string s, and generates the final samples. We call this variant *strong-sampler semantic security.*

Theorem 1.2 (informal)**.** *Let $\mathcal{O}$ be any obfuscator for a class $\mathcal{C}$ of circuits, that is VBB against semi-bounded adversaries in the ideal graded encoding model. Then instantiating the graded encoding oracle with a strong-sampler semantically-secure graded encoding scheme results in an obfuscator $\mathcal{O}'$ that is SIO for $\mathcal{C}$ in the plain model.*

Then, relying on the Barak et al. obfuscation for NC^1 in the ideal graded encoding model [BGK+13] (which is indeed VBB against semi-bounded adversaries), we obtain the following corollary.

Corollary 1.1 (informal)**.** *Assume there exists a strong-sampler semantically-secure encoding scheme. Then there exists SIO for NC^1.*

We also give evidence for the *necessity* of semantically-secure graded encoding schemes for obtaining VGB. To this end, we focus on a version of graded encoding with restricted functionality called *mutlilinear jigsaw puzzles* [GGH+13b]. Unlike graded encodings, in mutlilinear jigsaw puzzles, encodings can only be generated together with the system parameters. We refer to the public parameters, together with the set of initialized encodings, as a puzzle. Instead of performing individual permitted operations over the encodings, all the jigsaw puzzle user can do is to specify an arithmetic circuit C that respects the set structure of the initialized encodings, and test whether C evaluates to 0 on these encodings or not. Semantic security of mutlilinear jigsaw puzzles is defined similarly to the graded encoding case. Despite their restricted functionality, semantically-secure mutlilinear jigsaw puzzles can replace graded encodings in our construction of SIO for NC^1.

We observe that the existence of semantically-secure jigsaw puzzles is implied by VGB obfuscation for all circuits. To see why this is the case, consider the circuit P that has a set of ring elements $\vec{m} = (m_1, \ldots, m_\ell)$ hardwired into it, together with the corresponding sets $\vec{S} = (S_1, \ldots, S_\ell)$. The circuit P takes as input an arithmetic circuit C that respects the set structure given by $\vec{S}$, and tests whether $C(\vec{m}) = 0$. To initialize a puzzle from a set of encodings $(\vec{S}, \vec{m})$ we simply VGB obfuscate the circuit P.

1.4 More on Semantic Security

Next we discuss our results pertaining to the study of semantic security of graded encoding schemes. The negative results discussed below hold even for the basic notion of semantic security, where the message sampler is of polysize.

SAT lower bounds. As additional evidence to the power of semantically-secure graded encodings, we observe that they imply an exponential lower-bound for Circuit-SAT. Specifically, there do no exist SAT solvers that run in time $2^{o(n)} \cdot \mathrm{poly}(|C|)$, for a boolean circuit C with n input variables. To show this lower bound, we rely on a result by Wee [Wee05] that shows a similar lower from any point function obfuscation.

Efficiency limitations via a generic attack. We present an attack against any graded encoding scheme satisfying certain efficiency properties. Before specifying these efficiency properties, let us first describe the high-level idea behind the attack, from which they emerge.

Let $C_{b,r}$ be a circuit associated with a bit b and ring element $r \in R$. The circuit $C_{b,r}$ reveals the bit b only when given as input some public parameters pp and a proper encoding of r with respect to pp. The corresponding admissible sampler $\mathbb{M}$ would then sample from one of two distributions $\mathcal{D}_0, \mathcal{D}_1$, where sampling from $\mathcal{D}_b$ is done by sampling a random r and outputting a vector of ring elements $\vec{m}$, that represents an ideal obfuscation of $C_{b,r}$, and the additional element r. Intuitively, a semi-bounded, algebraic adversary with ideal access to $(\vec{m}, r)$ gains no more than oracle-access to $C_{b,r}$ and since it

cannot recover r, it cannot learn the bit b.[4] In contrast, the real world distinguisher, which is given the public parameters pp and an actual encoding of r, can simply run the obfuscation on pp and the encoding of r, and learn b.

The above attack is applicable for graded encoding schemes that possess certain efficiency properties. First, since we only have ideal obfuscation against semi-bounded adversaries for NC^1, the circuit $C_{b,r}$ should be implementable in NC^1. Second, it is required that the size of the public parameters pp and the size of an encoding do not grow with the size of the universe set k. Indeed, in order to obfuscate $C_{b,r}$ in the ideal graded encoding model, it is required that k is appropriately large (in particular, larger than the circuit's input). Thus, the size of the public parameters and encodings received by $C_{b,r}$ must be independent of k. [5]

Both of the above efficiency requirements are not satisfied by the candidate constructions of [GGH13a, CLT13] in their current forms. Indeed, for these schemes it is not known how to implement $C_{b,r}$ in NC^1. Also, in these schemes the size of the public parameters and encodings does grow with the size of the universe set k. Still, this attack motivates us to consider more restricted notions of semantically-secure graded encoding.

Bounded semantic security. To get around diagonalization attacks such as the one described above, we suggest a relaxation of semantic security where there is a fixed polynomial bound on the length of the elements vector produced by the admissible sampler. This bound is fixed a a-priori before the encoding scheme's parameters, such as the size of the universe set k, are chosen. This relaxed definition is sufficient for all of our applications. It also appears to be resilient to diagonalization attacks.

Other relaxations of semantic security. We also discuss several other relaxations of the semantic-security definition that were considered by Pass et al. [PTS13] (and are already embedded into their main definition of semantic security). We show that under any combination of these relaxations, the positive implications to obfuscation still hold. For most combinations we do this generically given any VBB obfuscation construction in the ideal graded encoding model.

SIO from new assumptions? Pass et al. also consider an alternative version of semantic security, where instead of requiring indistinguishability with respect to *any* admissible sampler, it is only required for a *single* specific sampler (that depends on the obfuscation construction). Gentry et al. [GLSW14] recently constructed IO based on a new, construction-independent assumption of a different flavor. Getting IO from these assumptions involves an exponential loss in security, and they are therefore incomparable to the assumption discussed here. Whether these assumptions suffice for constructing SIO is left for future research to ascertain.

Organization. Section 2 reviews the definitions of VBB, VGB and IO. Section 3 defines SIO and shows its equivalence to VBB for concentrated circuit distributions. Section 4 constructs worst case VGB and VBB obfsucators from SIO. Section 5 shows that SIO is equivalent to IO and average-case VGB for evasive sub-collections. Section 6 constructs SIO from semantically-secure graded encoding schemes. Finally, Section 7 studies the semantic security assumption.

2 Obfuscation: VBB, VGB, Indistinguishability

We review three basic definitions of obfuscation that are used throughout the paper. We start by defining the functionality requirement, which all the notions share, and then define different security notions.

Definition 2.1 (Functionality). *A PPT algorithm $\mathcal{O}$ is an obfuscator for a collection of circuits $\mathcal{C} =$*

[4] This argument is a bit oversimplified; indeed, to argue that one cannot learn b given oracle access to r, we should also deal with the case that the adversary queries with improper public parameters and encodings.

[5] We thank Rafael Pass for pointing this out.

$\bigcup_{n\in\mathbb{N}} \mathcal{C}_n$, *if for any* $C \in \mathcal{C}$,

$$\Pr_{\mathcal{O}}[\forall x : \mathcal{O}(C)(x) = C(x)] = 1 \ .$$

VBB and VGB Obfuscation. Virtual Black Box (VBB) obfuscation [BGI+01] guarantees that an obfuscated circuit $\mathcal{O}(C)$ does not reveal any predicate $\pi(C)$ that cannot be learned by an efficient simulator that is given only black-box access to C. The basic definition is *worst-case* in the sense that the simulator needs to be successful for any circuit in a given circuit collection. We later also address an *average-case* notion. In the definition below we use a slightly weaker definition than the standard one, and allow the simulator to depend on the distinguishing probability p.

Definition 2.2 (Worst-case VBB Obfuscation)**.** *An obfuscator $\mathcal{O}$ for a collection of circuits $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ is worst-case VBB if for every poly-size adversary $\mathcal{A}$, and polynomial p, there exists a poly-size simulator $\mathcal{S}$, such that for every $n \in \mathbb{N}$, every predicate $\pi : \mathcal{C}_n \to \{0,1\}$, and every $C \in \mathcal{C}_n$:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C)) = \pi(C)] - \Pr_{\mathcal{S}}[\mathcal{S}^C(1^n) = \pi(C)] \right| \leq 1/p(n) \ .$$

Virtual Grey Box (VGB) obfuscation [BC10] relaxes VBB by allowing the simulator to have unbounded computational power, but still only a bounded number of oracle queries to C.

Definition 2.3 (Worst-case VGB Obfuscation)**.** *An obfuscator $\mathcal{O}$ for a collection of circuits $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ is worst-case VGB if for every poly-size adversary $\mathcal{A}$, and polynomial p, there exists an unbounded simulator $\mathcal{S}$, and a polynomial q, such that for every $n \in \mathbb{N}$, every predicate $\pi : \mathcal{C}_n \to \{0,1\}$, and $C \in \mathcal{C}_n$:*

$$\left| \Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C)) = \pi(C)] - \Pr_{\mathcal{S}}[\mathcal{S}^{C[q(n)]}(1^n) = \pi(C)] \right| \leq 1/p(n) \ ,$$

where $C[q(n)]$ is an oracle that allows at most $q(n)$ queries.

We also consider relaxed versions of VBB and VGB, where the corresponding guarantee only holds for a random circuit sampled from a distribution, rather than for any circuit.

Definition 2.4 (Average-case obfuscation)**.** *Each of Definitions 2.2,2.3 is said to hold in the average case, for a distribution ensemble $\tilde{\mathcal{C}} = \bigcup_{n\in\mathbb{N}} \tilde{\mathcal{C}}_n$ on the collection $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$, if each of the corresponding probability statements is over a random $C \leftarrow \tilde{\mathcal{C}}_n$, rather than required for every $C \in \mathcal{C}$.*

Remark 2.1 (Simulation accuracy)**.** *In the above definitions (and throughout the paper), the simulator $\mathcal{S}$, and in VGB, also its number of queries q, are allowed to depend on the required simulation accuracy $1/p(n)$. This definition is stronger than the VBB definition in common also to all previous works that have established worst-case VBB or VGB (for specific classes), but is weaker then the original definition of [BGI+01] where the same simulator $\mathcal{S}$ should be $1/p(n)$-accurate for all polynomials p (for large enough security parameter n).*

Indistinguishability Obfuscation. We next define the notion of indistinguishability obfuscation, introduced in [BGI+01].

Definition 2.5 (Indistinguishability obfuscation [BGI+01])**.** *An obfuscator for $\mathcal{C}$ is said to be an* indistinguishability obfuscator *for $\mathcal{C}$, denoted by $i\mathcal{O}$, if for any poly-size distinguisher $\mathcal{D}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$, and any two circuits $C_0, C_1 \in \mathcal{C}_n$ of the same size and functionality,*

$$\Pr[b \leftarrow \{0,1\}; \mathcal{D}(C_0, C_1, i\mathcal{O}(C_b)) = b] \leq \frac{1}{2} + \mu(n) \ .$$

It can be readily seen that if an obfuscator $\mathcal{O}$ is VBB for a function collection $\mathcal{C}$ then it is also VGB for $\mathcal{C}$. Furthermore, if $\mathcal{O}$ is VGB for $\mathcal{C}$ then it is also an indistinguishability obfuscator for $\mathcal{C}$.

3 Strong Indistinguishability Obfuscation

In this section we define the notion of strong indistinguishability obfuscation (SIO). We start by defining the notion of concentrated distributions over circuits.

Concentrated Circuit Distributions. At a high-level, a distribution ensemble $\tilde{\mathcal{C}}$, over a circuit collection $\mathcal{C}$, is *concentrated*, if given polynomially many oracle queries to a random circuit C from the distribution, it is information theoretically hard to find an input x such that C does not agree with $\mathrm{maj}_{\tilde{\mathcal{C}}}$ on the point x, where $\mathrm{maj}_{\tilde{\mathcal{C}}}$ is the common output of circuits distributed according to $\tilde{\mathcal{C}}$. If $\tilde{\mathcal{C}}$ corresponds to the uniform distribution on some collection $\mathcal{C}$, $\mathrm{maj}_{\tilde{\mathcal{C}}}$ is simply the majority vote. Concentrated distributions naturally generalize the concept of *evasive distributions* studied in [BBC+14], in which the majority is always the all-zero function, i.e. $\mathrm{maj}_{\tilde{\mathcal{C}}} \equiv 0$.

Definition 3.1 (Concentrated circuit distributions)**.** *Let $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ be a circuit collection, where $\mathcal{C}_n$ consists of circuits $C : \{0,1\}^n \to \{0,1\}$ of size $\mathrm{poly}(n)$, and let $\tilde{\mathcal{C}}_n$ be a distribution on $\mathcal{C}_n$. Let $\mathrm{maj}_{\tilde{\mathcal{C}}_n}(x) := \lfloor \mathbb{E}_{C\leftarrow\tilde{\mathcal{C}}_n} C(x) \rceil$ be the common output at point x of circuits drawn from $\tilde{\mathcal{C}}_n$.*

1. *For any $\varepsilon \in [0,1]$, $\tilde{\mathcal{C}}_n$ is said to be ε-concentrated if*

$$\max_{x\in\{0,1\}^n} \Pr_{C\leftarrow\tilde{\mathcal{C}}_n} \left[C(x) \neq \mathrm{maj}_{\mathcal{C}_n}(x)\right] \leq \varepsilon \ .$$

2. *$\tilde{\mathcal{C}}$ is said to be concentrated if for some negligible $\mu(\cdot)$, and any $n \in \mathbb{N}$, $\tilde{\mathcal{C}}_n$ is $\mu(n)$-concentrated.*
3. *$\tilde{\mathcal{C}}$ is said to be evasive if it is concentrated, and for any $n \in \mathbb{N}$ and any $x \in \{0,1\}^n$, $\mathrm{maj}_{\tilde{\mathcal{C}}_n}(x) = 0$.*
4. *We say that the collection $\mathcal{C}$ itself is concentrated (evasive) if the uniform distribution ensemble on circuits in $\mathcal{C}$ is concentrated (evasive).*

Strong Indistinguishability Obfuscation. Strong Indistinguishability Obfuscation requires that indistinguishability holds, even when C_0 and C_1 do not necessarily compute the exact same function, but are taken from two distributions $\tilde{\mathcal{C}}^0_n$ and $\tilde{\mathcal{C}}^1_n$ that are concentrated around the same function; namely, $\mathrm{maj}_{\tilde{\mathcal{C}}^0_n} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}^1_n}$:

Definition 3.2 (Strong indistinguishability obfuscation)**.** *An obfuscator for $\mathcal{C}$ is said to be a* **strong** indistinguishability obfuscator *for $\mathcal{C}$, denoted by $i\mathcal{O}^*$, if for any two concentrated distribution ensembles $\tilde{\mathcal{C}}^0, \tilde{\mathcal{C}}^1$ on $\mathcal{C}$, such that $\forall n \in \mathbb{N} : \mathrm{maj}_{\tilde{\mathcal{C}}^0_n} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}^1_n}$, and any poly-size distinguisher $\mathcal{D}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$,*

$$\Pr[b \leftarrow \{0,1\}; (C_0, C_1) \leftarrow (\tilde{\mathcal{C}}^0_n, \tilde{\mathcal{C}}^1_n); \mathcal{D}(i\mathcal{O}^*(C_b)) = b] \leq \frac{1}{2} + \mu(n) \ .$$

Remark 3.1. *Above, we do not require that the distributions $\tilde{\mathcal{C}}^0, \tilde{\mathcal{C}}^1$ are efficiently samplable. We can also consider a weaker definition where this restriction is added. Later we show that this weaker version can be obtained from a weaker notion of semantic security (see Remark 6.4).*

We observe that any SIO obfuscator for $\mathcal{C}$ is also an IO obfuscator for $\mathcal{C}$. Indeed, for any two circuits C_0, C_1 of equivalent functionality, each of these circuits on its own is trivially concentrated around their common functionality.

4 SIO is Equivalent to Worst-Case VGB

In this section, we prove that the notion of strong indistinguishability obfuscation (SIO) is equivalent to VGB. Clearly, any VGB obfuscator for a class $\mathcal{C}$ is also a SIO for $\mathcal{C}$. We show that the converse is true as well. Namely, we show that any strong indistinguishability obfuscator $\mathcal{O}$ for a class $\mathcal{C}$ of circuits is a worst-case VGB obfuscator for $\mathcal{C}$. In addition, we show that for classes $\mathcal{C}$ with some additional properties, $\mathcal{O}$ is in fact worst-case VBB. We refer the reader to Section 1.1 for an overview.

4.1 Definitions and Statement of Main Theorem

Notation and terminology. For a function $f : \{0,1\}^n \to \{0,1\}$, we say that a point $x \in \{0,1\}^n$ *separates* a circuit C from f if $C(x) \neq f(x)$. We say that a set $L \subseteq \{0,1\}^n$ separates C from f, if some $x \in L$ separates C from f. Given a circuit collection $\mathcal{K}$, we say that L separates $\mathcal{K}$ from f, if L separates any $C \in \mathcal{K}$ from f. Recall, that we say that a collection $\mathcal{K}$ is concentrated if the uniform distribution on $\mathcal{K}$ is concentrated around its majority function $\mathrm{maj}_{\mathcal{K}}$.

Definition 4.1 (Majority-separating oracle)**.** *Let $\mathcal{C}$ be a collection of boolean circuits defined over $\{0,1\}^n$, let $C \in \mathcal{C}$, and let $\varepsilon > 0$. An oracle $\mathbb{S}$ is said to be $(\mathcal{C}, C, \varepsilon)$-separating if given any ε-concentrated sub-collection $\mathcal{K} \subseteq \mathcal{C}$, represented by a circuit that samples uniform elements in $\mathcal{K}$, $\mathbb{S}(\mathcal{K})$ outputs a point $x \in \{0,1\}^n$ that separates C from $\mathrm{maj}_{\mathcal{K}}$, or $\perp$ if no such point exists.*

Remark 4.1. *In the above definition, and throughout this section, we often abuse notation and denote by $\mathcal{K}$ both the sub-collection and the circuit that samples uniform elements from the sub-collection.*

Definition 4.2 (Learnability by majority-separating oracles)**.** *A collection $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ of boolean circuits is said to be $(t, \vec{c}, s, \varepsilon)$-learnable by a majority-separation oracle if there exists a deterministic oracle-aided machine $\mathcal{L}$ such that, given oracle access to $C \in \mathcal{C}_n$ and a $(\mathcal{C}_n, C, \varepsilon(n))$-separating oracle $\mathbb{S}$, $\mathcal{L}^{C,\mathbb{S}}(1^n)$ outputs $\hat{C} \in \mathcal{C}_n$ of equivalent functionality to C, in time $t(n)$, using at most $s(n)$ queries to $\mathbb{S}$, and at most $\vec{c}_i(n)$ queries to C between the $i-1$-st and the i-th calls to $\mathbb{S}$.*

Our main technical theorem shows that any strong indistinguishability obfuscator for a circuit collection $\mathcal{C}$ that is learnable via a majority separation oracle is also a worst-case simulation-based obfuscator. The size and query complexity of the worst-case simulator, in particular whether it is a VBB or VGB simulator, is determined by the learnability parameters $(t, \vec{c}, s, \varepsilon)$.

Theorem 4.1. *Let $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ be a circuit collection that is $(t, \vec{c}, s, \frac{1}{q})$-learnable by a majority-separating oracle, for some polynomial q. Let $\mathcal{O}$ be a strong indistinguishability obfuscator for $\mathcal{C}$, let $\mathcal{A}$ be a boolean poly-size adversary, and let p be a polynomial. Then $(\mathcal{A}, p)$ has a simulator $\mathcal{S}$ of size $O(|\mathcal{A}| + t \cdot s \cdot q^s \cdot \prod_{i=1}^{s} 2^{\vec{c}_i})$ with $O(\|\vec{c}\|_1 + q \cdot s)$ oracle queries. The simulator works in the worst-case for any $C \in \mathcal{C}$.*

In Section 4.3 we show that any circuit collection $\mathcal{C}$ is indeed $(t, \vec{c}, s, \frac{1}{q})$-learnable, for some setting of parameters (where $\|\vec{c}\|_1, q, s$ are polynomially bounded).

4.2 Proof of Theorem 4.1

Fix $\mathcal{C}, \mathcal{O}, \mathcal{A}, p$ satisfying the conditions of the theorem. The proof of the theorem will rely on the following key lemma that essentially shows that the set of circuits, whose obfuscation $\mathcal{A}$ can $1/p$-distinguish, can always be separated from the majority of circuits by a small separating set.

Lemma 4.1. *There exists a polynomial q, such that for any $n \in \mathbb{N}$, and any $\frac{1}{q(n)}$-concentrated sub-collection $\mathcal{K} \subseteq \mathcal{C}_n$, there exists a set $L_{\mathcal{K}} \subseteq \{0,1\}^n$ of size at most $q(n)$, such that for any $C \in \mathcal{K}$ that is not separated from $\mathrm{maj}_{\mathcal{K}}$ by $L_{\mathcal{K}}$:*

$$\left|\Pr[\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{C' \leftarrow \mathcal{K}}[\mathcal{A}(\mathcal{O}(C')) = 1]\right| \leq 1/p(n) \ ,$$

where the probability is also over the coins of $\mathcal{A}$ and $\mathcal{O}$.

Proof. For any $n \in \mathbb{N}$, and sub-collection $\mathcal{K} \subseteq \mathcal{C}_n$, let us denote by $\mathbb{D}^b(\mathcal{K})$ the collection of all circuits $C \in \mathcal{K}$ such that

$$(-1)^b \left(\Pr[\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{C' \leftarrow \mathcal{K}}[\mathcal{A}(\mathcal{O}(C')) = 1]\right) \geq 1/p(n) \ ;$$

namely, $\mathbb{D}(\mathcal{K}) := \mathbb{D}^0(\mathcal{K}) \cup \mathbb{D}^1(\mathcal{K})$ consist of all the "distinguishable circuits" in $\mathcal{K}$.

Assume towards contradiction that the lemma does not hold with respect to $\mathcal{A}$ and p. Then there exists a super-polynomial function $T(n) = n^{\omega(1)}$, such that for an infinite sequence $\mathbb{N}^* \subseteq \mathbb{N}$, and any $n \in \mathbb{N}^*$, there exists a $\frac{1}{T(n)}$-concentrated sub-collection $\mathcal{K}_n \subseteq \mathcal{C}_n$ such that any set $L_{\mathcal{K}_n}$ separating the distinguishable circuits $\mathbb{D}^0(\mathcal{K}_n) \cup \mathbb{D}^1(\mathcal{K}_n)$ from $\mathrm{maj}_{\mathcal{K}_n}$ is of size greater than $T(n)$. In particular, for some $b_n \in \{0,1\}$, any set separating $\mathbb{D}^{b_n}(\mathcal{K}_n)$ from $\mathrm{maj}_{\mathcal{K}_n}$ is of size greater than $T(n)/2$. For ease of notation, let us assume throughout that $b_n = 0$, and simply denote $\mathbb{D}(\mathcal{K}_n) = \mathbb{D}^0(\mathcal{K}_n)$. (This is indeed WLOG, by flipping the output of $\mathcal{A}$ if needed.)

Claim 4.1. *For any $n \in \mathbb{N}^*$, there exists a non-empty concentrated sub-collection $\mathbb{D}^*(\mathcal{K}_n) \subseteq \mathbb{D}(\mathcal{K}_n)$. Specifically,*

$$\max_{x \in \{0,1\}^n} \left\{ \Pr_{C \leftarrow \mathbb{D}^*(\mathcal{K}_n)} \left[C(x) \neq \mathrm{maj}_{\mathcal{K}_n}(x)\right] \right\} \leq \alpha(n) := \frac{2 \log |\mathcal{C}_n|}{T(n)} \ .$$

Proof. We describe an iterative process that results in the required $\mathbb{D}^*(\mathcal{K}_n)$. Let $\mathbb{D}_0 = \mathbb{D}(\mathcal{K}_n)$, and let $L_0 = \emptyset$. Given $\mathbb{D}_i$, we define $\mathbb{D}_{i+1}$ as follows. If $\mathbb{D}_i$ satisfies the property given by the claim, output $\mathbb{D}^*(\mathcal{K}_n) = \mathbb{D}_i$. Otherwise, there exists a point $x_i \in \{0,1\}^n$ that separates an $\alpha(n)$-fraction of the circuits in $\mathbb{D}_i$ from $\mathrm{maj}_{\mathcal{K}_n}$. Then, add x_i to current partial separating set $L_{i+1} = L_i \cup \{x_i\}$, and let $\mathbb{D}_{i+1} \subseteq \mathbb{D}_i$ be the sub-collection that is not separated from $\mathrm{maj}_{\mathcal{K}_n}$ by x_i.

Note that this process ends after at most $T(n)/2$ steps (we do not require that it is efficient). Indeed, it holds that for $i \leq T(n)/2$ and for $\alpha = \frac{2 \log |\mathcal{C}_n|}{T(n)}$,

$$|\mathbb{D}_i| \leq (1 - \alpha(n))^i \, |\mathbb{D}| \leq \left(1 - \frac{2 \log |\mathcal{C}_n|}{T}\right)^{T/2} |\mathcal{C}_n| < 2^{-\log |\mathcal{C}_n|} \cdot |\mathcal{C}_n| \leq 1 \ .$$

Moreover, this process must end with a non-empty set. This is the case since otherwise after $T(n)/2$ steps we separated all of the original $\mathbb{D}(\mathcal{K}_n)$ from $\mathrm{maj}_{\mathcal{K}_n}$ with a set $L_{T(n)/2} \subseteq \{0,1\}^n$ of size less than $T(n)/2$. This contradicts the fact that $\mathbb{D}(\mathcal{K}_n)$ cannot be separated from $\mathrm{maj}_{\mathcal{K}_n}$ by $T(n)/2$ elements or less. □

We now show how to violate the fact that $\mathcal{O}$ is a SIO obfuscator for $\mathcal{C}$. Consider the concentrated sub-collections $\mathcal{D}^* = \bigcup_{n \in \mathbb{N}^*} \mathbb{D}^*(\mathcal{K}_n)$, and $\mathcal{K} = \bigcup_{n \in \mathbb{N}} \mathcal{K}_n$. (Formally, we need to also define these for $n \in \mathbb{N} \setminus \mathbb{N}^*$. We can do so in an arbitrary way that will keep them concentrated.) Since both $\mathbb{D}^*(\mathcal{K}_n)$ and $\mathcal{K}_n$ are concentrated around $\mathrm{maj}_{\mathcal{K}_n}$, it suffices to show that $\mathcal{A}$ distinguishes

$$\{\mathcal{O}(C) : C \leftarrow \mathcal{K}_n\}_{n \in \mathbb{N}} \text{ from } \{\mathcal{O}(C) : C \leftarrow \mathbb{D}^*(\mathcal{K}_n)\}_{n \in \mathbb{N}} \ .$$

Indeed, for any $n \in \mathbb{N}^*$

$$\begin{aligned} \Pr_{C \leftarrow \mathbb{D}^*(\mathcal{K}_n)}[\mathcal{A}(\mathcal{O}(C)) = 1] \quad - \quad \Pr_{C \leftarrow \mathcal{K}_n}[\mathcal{A}(\mathcal{O}(C)) = 1] \geq \\ \min_{C \in \mathbb{D}^*(\mathcal{K}_n)} \Pr[\mathcal{A}(\mathcal{O}(C)) = 1] \quad - \quad \Pr_{C \leftarrow \mathcal{K}_n}[\mathcal{A}(\mathcal{O}(C)) = 1] \geq \frac{1}{p(n)} \ . \end{aligned}$$

□

To complete the proof of Theorem 4.1, let us fix q to be the polynomial given by Lemma 4.1 corresponding to $(\mathcal{A}, p)$, and assume that $\mathcal{C}$ is $(t, \vec{c}, s, \frac{1}{q})$-learnable by a majority-separating oracle. We next describe the simulator $\mathcal{S}$ for $(\mathcal{A}, p)$, argue its validity, and analyze its complexity.

Description of $\mathcal{S}$. Given oracle access to $C \in \mathcal{C}_n$, $\mathcal{S}$ runs the learner $\mathcal{L}^{C,\mathbb{S}}(1^n)$ given by Definition 4.2, and emulates for $\mathcal{L}$ the oracle C and the majority-separating oracle $\mathbb{S}$. Any call to C is answered by $\mathcal{S}$ using its own oracle to C. Oracle calls to $\mathbb{S}$ are handled as follows. Given a sub-collection $\mathcal{K} \subseteq \mathcal{C}_n$ that contains C (represented by a circuit that samples uniform elements in $\mathcal{K}$), $\mathcal{S}$ first retrieves the set $L_{\mathcal{K}}$ separating the distinguishable circuits $\mathbb{D}(\mathcal{K}) \subseteq \mathcal{K}$ from $\mathrm{maj}_{\mathcal{K}}$. Then, $\mathcal{S}$ queries its oracle C on all the points $x \in L_{\mathcal{K}}$, and tests whether $C(x) = \mathrm{maj}_{\mathcal{K}}(x)$, namely whether x separates C from $\mathrm{maj}_{\mathcal{K}}$.

If $\mathcal{S}$ found a separating point x, then it uses it to answer $\mathcal{L}$'s query, and continues its emulation. Otherwise, if no point in $L_{\mathcal{K}}$ separates C from $\mathrm{maj}_{\mathcal{K}}$, then $\mathcal{S}$ stops the emulation of $\mathcal{L}$, samples a random $C' \leftarrow \mathcal{K}$, and outputs the result of running $\mathcal{A}(\mathcal{O}(C'))$. In any case, after running $\mathcal{L}$ for at most $t(n)$ steps, $\mathcal{L}$ would output $\hat{C} \in \mathcal{C}_n$ of equivalent functionality to C, and $\mathcal{S}$ outputs the result of running $\mathcal{A}(\mathcal{O}(\hat{C}))$.

Validity. The validity of $\mathcal{S}$ follows from Lemma 4.1 and the guarantee on $\mathcal{L}$. Indeed, by Lemma 4.1, if at any point C agrees with $\mathrm{maj}_{\mathcal{K}}$ on all of $\mathcal{L}_{\mathcal{K}}$, then $\mathcal{A}$ distinguishes an obfuscation $\mathcal{O}(C)$ from an obfuscation $\mathcal{O}(C')$ for a random $C' \leftarrow \mathcal{K}$ with probability at most $\frac{1}{p(n)}$. Otherwise, we successfully implement a $(\mathcal{C}_n, C, \frac{1}{q(n)})$-majority-separating oracle $\mathbb{S}$, and learn $\hat{C} \in \mathcal{C}_n$ of equivalent functionality to C. Since $\mathcal{O}$ is a strong indistinguishability obfuscator, and in particular an indistinguishability obfuscator, $\mathcal{A}$'s advantage is also bounded by $1/p(n)$ in this case.

Complexity of $\mathcal{S}$. The queries of $\mathcal{S}$ to C include the $c(n) = \sum_1^{s(n)} \vec{c}_i(n)$ queries that $\mathcal{L}$ makes to C, and $q(n)$ queries for each of the $s(n)$ queries made by $\mathcal{L}$ to $\mathbb{S}$; indeed, remember that the size of each $\mathcal{L}_{\mathcal{K}}$ is bounded by $q(n)$. Thus the overall query complexity of $\mathcal{S}$ is $c(n) + s(n) \cdot q(n)$.

The total size of $\mathcal{S}$ can be bounded by the sum of

1. the size of the adversary $|\mathcal{A}|$,
2. the total size $t(n)$ of $\mathcal{L}$,
3. the number of sets $L_{\mathcal{K}}$ that the simulator may have to use throughout the simulation, times the size $q(n)$ of each $L_{\mathcal{K}}$,
4. the time it takes to compute the values $\mathrm{maj}_{\mathcal{K}}(x)$ throughout.

We now count the number of sets $L_{\mathcal{K}}$ necessary for $\mathcal{S}$; for ease of notation, we suppress the security parameter n. Consider the (deterministic) learner $\mathcal{L}$, we view its tree of possible executions. Let us view each node at level $0 \leq i \leq s$, as corresponding to the state of $\mathcal{L}$ before making $\vec{c}_{i+1}$ queries to C and the $i+1$-st query to $\mathbb{S}$. Here a node at the i-th level has $2^{\vec{c}_{i+1}} \cdot q$ sons. Indeed, there are at most $2^{\vec{c}_{i+1}}$ possible sequences of queries and answers made by $\mathcal{L}$ to the oracle C, before the i-th query to $\mathbb{S}$; then, each such possible sequence determines a set $L_{\mathcal{K}}$ of q values that $\mathcal{L}$ will query $\mathbb{S}$ on. The overall number

of sets $L_{\mathcal{K}}$ is thus

$$\sum_{i=0}^{s-1} q^i \cdot \prod_{j=1}^{i+1} 2^{\vec{c}_j} \leq s \cdot q^{s-1} \prod_{i=1}^{s} 2^{\vec{c}_i} \ .$$

Throughout the simulation, $\mathrm{maj}_{\mathcal{K}}(x)$ is computed at most $s(n) \cdot q(n)$ times. The computation itself can be done by computing $C_1(x), \ldots, C_n(x)$, where each $C_i \leftarrow \mathcal{K}$ is a random circuit from $\mathcal{K}$ and taking their majority. This computation errs with negligible probability $2^{-\Omega(n)}$ (see Claim 5.1).[6] Sampling $C_i \leftarrow \mathcal{K}$ and computing $C_i(x)$, can be done in time at most $t(n)$ (i.e., the size of $\mathcal{L}$) since we assume that the learner $\mathcal{L}$ represents each of its queries $\mathcal{K}$ to its majority-separating oracle $\mathbb{S}$ by a circuit that samples uniform elements in $\mathcal{K}$.

This completes the proof of Theorem 4.1. □

4.3 VGB and VBB by Majority-Separation Learning

In this section, we show that any class of circuits is learnable by a majority-separating oracle, with parameters that yield VGB simulation. We then discuss additional classes that can be learned with better parameters, yielding VBB simulation. This includes previously obfuscated classes as well as new ones.

4.3.1 VGB Obfuscation for All Circuits.

We show

Theorem 4.2. *Let $\mathcal{C}$ be any circuit collection and let $\mathcal{O}$ be a strong indistinguishability obfuscator for $\mathcal{C}$. Then $\mathcal{O}$ is also a worst-case VGB obfuscator for $\mathcal{C}$.*

To prove Theorem 4.2, we show that any circuit collection is learnable by a majority-separating oracle, where the learner is of unbounded size, but only performs a polynomial number of queries to its oracles. Theorem 4.2 then follows from Theorem 4.1.

Lemma 4.2. *For any $q > 2$, any circuit collection $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ is $(t, \vec{c}, s, \frac{1}{q})$-learnable by a majority-separating oracle for $t(n) = \infty$, $s(n) \leq \|\vec{c}(n)\|_1 \leq q(n) \cdot \log|\mathcal{C}_n|$.*

Proof. We describe the required learner $\mathcal{L}$. $\mathcal{L}^{C,\mathbb{S}}$ works iteratively; starting from the entire collection $\mathcal{K}_0 = \mathcal{C}_n$, it each time reduces the current set of candidates $\mathcal{K}_i$ to a strict sub-collection $\mathcal{K}_{i+1} \subsetneq \mathcal{K}_i$, until it finds a circuit $\hat{C}$ computing the same function as C, or C itself. Specifically, as long as $\mathcal{K}_i$ contains some x that separates at least a $\frac{1}{q}$-fraction of the circuits in $\mathcal{K}_i$ from $\mathrm{maj}_{\mathcal{K}_i}$, $\mathcal{L}$ queries C on x and defines $\mathcal{K}_{i+1}$ to be the subset of all circuits in $\mathcal{K}_i$ that agree with C on x. If no such x exists then $\mathcal{K}_i$ is $\frac{1}{q}$-concentrated, in which case $\mathcal{L}$ asks the majority-separating oracle $\mathbb{S}$ for a point x that separates C from $\mathrm{maj}_{\mathcal{K}_i}$. If $\mathbb{S}$ returns $\perp$, then $C \equiv \mathrm{maj}_{\mathcal{K}_i}$ is returned. Otherwise, $\mathcal{L}$ queries C on x and continues as before.

It is left to note that, with each query to C of the first type, $\mathcal{K}_i$ is reduced by a factor of $(1 - 1/q(n))$, and with each query of the second type it is reduced by a factor of $1/q(n) < 1 - 1/q(n)$, and thus:

$$|\mathcal{K}_i| \leq (1 - 1/q(n))^i |\mathcal{K}_0| \leq 2^{-i/q(n)} |\mathcal{C}_n| \leq 2^{\log|\mathcal{C}_n| \cdot (1 - \frac{i}{q(n)\cdot\log|\mathcal{C}_n|})} \ .$$

This implies that $\mathcal{L}$ learns some $\hat{C} \in \mathcal{C}_n$ of equivalent functionality to C after at most $q(n) \cdot \log|\mathcal{C}_n|$ iterations, and thus $s(n) \leq \|\vec{c}(n)\|_1 \leq q(n) \cdot \log|\mathcal{C}_n|$. □

[6] Formally, we should account for this error in the simulation accuracy. This can be done by fixing q that corresponds to $(\mathcal{A}, p/2)$ rather than to $(\mathcal{A}, p)$.

4.3.2 VBB Obfuscation for Sets of Constant Size.

A k-set circuit C_S is associated with a set S of k points, it accepts all points in S, and rejects all other points. A special well-studied case of set circuits is that of point circuits, where $k = 1$.

Definition 4.3 (Set circuits)**.** *For a set $S \subseteq \{0,1\}^n$ of size k, the set circuit C_S returns 1 for any $x \in S$, and 0 for all $x \notin S$. $\mathcal{S}^k = \bigcup_{n\in\mathbb{N}} \mathcal{S}^k_n$, where $\mathcal{S}^k_n = \{C_S : S \subseteq \{0,1\}^n, |S| = k(n)\}$, is the collection of k-set circuits.*

Theorem 4.3. *Let $\mathcal{O}$ be a strong indistinguishability obfuscator for $\mathcal{S}^k$. Then $\mathcal{O}$ is also a worst-case VGB obfuscator for $\mathcal{S}^k$, with a simulator of size $n^{O(k)}$, and polynomially many queries. In particular, for $k = O(1)$ it is also a VBB obfuscator.*

As for Theorem 4.2, Theorem 4.3 is proven by showing how to learn set circuits via a majority-separating oracle with certain efficiency parameters, and plugging it in Theorem 4.1.

Lemma 4.3. *$\mathcal{S}^k$ is $(t, \vec{c}, s, \varepsilon)$-learnable by a majority-separating oracle for $t(n) = \mathrm{poly}(n)$, $\|\vec{c}\|_1 = 0$, $s(n) = k(n)$, and $\varepsilon(n) = 2^{-\Omega(n)}$.*

Proof. We describe the required learner $\mathcal{L}$. $\mathcal{L}^{C_S,\mathbb{S}}$ works iteratively, revealing the points in the set S one by one, as follows. Having already revealed a subset $T \subsetneq S$, $\mathcal{L}$ queries $\mathbb{S}$ on the sub-collection $\mathcal{K}_T$ corresponding to all the set circuits $C_{S'}$ such that $T \subseteq S'$ and $|S'| = k$, where $\mathcal{K}_T$ is represented via a $\mathrm{poly}(n)$-size circuit that samples a random element in $\mathcal{K}_T$.

Note that each such sub-collection $\mathcal{K}_T$ is $2^{-\Omega(n)}$-concentrated, since for a point $x \in T$ all circuits in $\mathcal{K}_T$ return 1, whereas for $x \notin T$, all but $2^{-\Omega(n)}$-fraction of the circuits in $\mathcal{K}_T$ return 0. Eventually, after at most $k(n)$ queries to $\mathbb{S}$, the entire set S is revealed. □

4.3.3 VBB Obfuscation for Linear Subspaces over Finite Fields.

A subspace circuit tests whether a given $x \in \mathbb{F}^d$ is a member of some linear subspace $L \subseteq \mathbb{F}^d$, or equivalently whether x belongs to the kernel of some given matrix $A \in \mathbb{F}^{d\times d}$.

Definition 4.4 (Subspace circuits)**.** *Let $\mathbb{F} = \{\mathbb{F}_n\}_{n\in\mathbb{N}}$ be a sequence of fields, each of size $2^{\Theta(n)}$, with efficient operations, let $d(n)$ be a polynomially bounded function, and let $A \in \mathbb{F}^{d(n)\times d(n)}$ be a matrix. The circuit C_A returns 1 if and only if $x \in \ker(A)$. Let $\mathcal{L}^{d,\mathbb{F}} = \bigcup_{n\in\mathbb{N}} \mathcal{L}^{d,\mathbb{F}}_n$, where $\mathcal{L}^{d,\mathbb{F}}_n = \left\{C_A : A \in \mathbb{F}^{d(n)\times d(n)}\right\}$ is the collection of subspace circuits.*

A special case of subspace circuit obfuscation, studied by [CRV10], is that of hyper-plane circuits where $\mathrm{rank}(A) = 1$. They show how to VBB obfuscate hyperplanes for dimension $d = O(1)$, under a strong variant of Decision Diffie Hellman.

We prove the following theorem.

Theorem 4.4. *Let $\mathcal{O}$ be a strong indistinguishability obfuscator for $\mathcal{L}^{d,\mathbb{F}}$. Then $\mathcal{O}$ is also a worst-case VGB obfuscator for $\mathcal{L}^{d,\mathbb{F}}$, with a simulator of size $n^{O(d)}$, and polynomially many queries. In particular, for $d = O(1)$ it is also a VBB obfuscator.*

As before, the theorem is proven by showing how to learn subspace circuits using a majority-separating oracle with ceratin efficiency features, and plugging it in Theorem 4.1.

Lemma 4.4. *$\mathcal{L}^{d,\mathbb{F}}$ is $(t, \vec{c}, s, \varepsilon)$-learnable by a majority-separating oracle for $t(n) = \mathrm{poly}(n)$, $\|\vec{c}(n)\|_1 = 0$, $s(n) \leq d(n)$, and $\varepsilon(n) = 2^{-\Omega(n)}$.*

Proof. We describe the required learner $\mathcal{L}$. Let $d = d(n)$ and let $A \in \mathbb{F}^{d \times d}$. $\mathcal{L}^{C_A, \mathbb{S}}$ gradually constructs a basis for $\ker(A)$. Having found a matrix $B \in \mathbb{F}^{d \times i}$ of $\mathrm{rank}(B) = i$, for $i < d$, and such that $AB = 0^i$, $\mathcal{L}$ queries the majority separating oracle $\mathbb{S}$ on the sub-collection $\mathcal{K}_B = \left\{C_{A'} : A'B = 0^i\right\}$, where $\mathcal{K}_B$ is represented by a $\mathrm{poly}(n)$-size circuit that samples uniform elements from $\mathcal{K}_B$. The collection $\mathcal{K}_B$ is $2^{-\Omega(n)}$-concentrated; indeed, for any $x \in \mathrm{span}(B)$, $x \in \ker(A')$ and $C_{A'}(x) = 1$ for all $C_{A'} \in \mathcal{K}_B$. For $x \notin B$, as long as $i < d$, $x \in \ker(A')$ with probability at most $|\mathbb{F}|^{-1} = 2^{-\Omega(n)}$. Thus, $\mathcal{L}$ obtains a separating vector $b_{i+1} \in \ker(A) \setminus \mathrm{span}(B)$, and can extend B to $B_{i+1} = (B_i | b_{i+1})$. If for some $i < d$, at the i-th step $\mathbb{S}$ returns $\perp$, it holds that $\ker(A) = \mathrm{span}(B_i)$, and $\mathcal{L}$ outputs some A' such that $\ker(A') = \mathrm{span}(B_i)$. Otherwise $A = 0^{d \times d}$. □

4.3.4 VBB Obfuscation for Connected Circuits.

Connected circuits consist of circuits C that can be fully learned efficiently given any point x such that $C(x) = 1$ and given oracle access to C. We further focus on collections of connected circuits that are also evasive, so that the pre-image $C^{-1}(1)$ of a typical circuit is sparse, and C is still hard to learn. We thus think of $C^{-1}(1)$ as a "connected" set, where given any point in the set, we can reach the others, but without such a point it is typically hard to hit the set.

Known examples of connected collections include point circuits (discussed above), conjunction circuits $C_{T,F}(x) = \bigwedge_{i \in T} x_i \bigwedge_{i \in F} \neg x_i$, for disjoint $T, F \subset [n]$, or Hamming ball circuits $C_{y,d}(x) = 1$ iff $|y - x| \leq d$, for $y \in \{0,1\}^n$ and $d \in [n]$ (these are also known as fuzzy point circuits).[7]

Definition 4.5 (Connected circuits)**.** *An evasive circuit collection $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$ is said to be connected, if there exists a poly-time learner $\mathcal{R}$ such that for every $C \in \mathcal{C}_n$ and $x \in C^{-1}(1)$, $\mathcal{R}^C(x) = \hat{C}$, for some $\hat{C} \in \mathcal{C}_n$ that is functionally equivalent to C.*

We prove the following theorem.

Theorem 4.5. *Let $\mathcal{O}$ be a strong indistinguishability obfuscator for any collection $\mathcal{C}$ of connected circuits. Then $\mathcal{O}$ is also a worst-case VBB obfuscator for $\mathcal{C}$.*

Again, the theorem is proved by showing how to learn the set connected circuits using a majority-separating oracle with ceratin efficiency features, and plugging it in Theorem 4.1.

Lemma 4.5. *Any collection $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$ of connected circuits is $(t, \vec{c}, s, \varepsilon)$-learnable by a majority-separating oracle for $t(n) = \mathrm{poly}(n)$, $s(n) = 1$, $\vec{c}_1(n) = 0$, $\vec{c}_2(n) = \mathrm{poly}(n)$, and $\varepsilon(n) = n^{-\omega(1)}$.*

Proof. We describe the required learner $\mathcal{L}$. $\mathcal{L}^{C_S, \mathbb{S}}$ queries the majority-separating $\mathbb{S}$ only once, on the entire collection $\mathcal{C}_n$ (which we assume to have an efficient circuit that samples uniform elements in $\mathcal{C}_n$), obtains a point x such that $C(x) = 1$, and uses the learner $\mathcal{R}$, given by Definition 4.5, to learn $\hat{C} \in \mathcal{C}_n$ with equivalent functionality to C. Recall that indeed, since $\mathcal{C}$ is evasive, $\mathrm{maj}_{\mathcal{C}_n} \equiv 0$ and $\mathcal{C}_n$ is $\mu(n)$-concentrated for some $\mu(n) = n^{-\omega(1)}$. □

4.3.5 Remarks.

Having presented the above results, a few technical remarks are in place.

Remark 4.2. *Barak et al. [BBC+14] show that if there exists an average-case VBB obfuscator for every evasive function collection, then for every collection of poly-size circuits, there exists a* weak *average-case VGB obfuscation, where the simulator is allowed some super-polynomial number of oracle queries.*

[7] Indeed, the first two examples are also evasive collections. The Hamming ball collection, for a given d, is evasive up to a certain threshold $d^* \in [n]$, and beyond that threshold, every function in the collection is exactly learnable.

Their result can also be scaled down to speak of all collections in NC^1. The VGB obfuscators constructed here are stronger in two aspects: first they are worst-case, rather than average-case, and second, the obfuscation simulator is only allowed a polynomial number of queries.

Remark 4.3 (Non-boolean functions)**.** *Our results are stated for boolean functions. For our result on VGB obfuscation for all circuits, this is without loss of generality, since for non-boolean circuit $C(x)$, we can obfuscate the boolean circuit $C'(x,i) = C_i(x)$ that returns the i-th output bit, given additional input i. As for our VBB results on restricted classes, such as set circuits, subspace circuits, or connected circuits, our results can be rather directly generalized to also allow a given multi-bit output. Namely, the image of any circuit is still boolean, but rather than $\{0,1\}$ it consists of $\{0,s\}$, for any given string s. These type of multi-bit output circuits were previously studied in [CD08, CRV10, BC10], and proven useful for strong forms of encryption.*

Remark 4.4 (Auxiliary input)**.** *The worst-case VBB and VGB definitions considered here allow a* non-universal simulator *[BCC$^+$14]. In particular, the simulator is allowed to have non-uniform advice that arbitrarily (and inefficiently) depends on the adversary's non-uniform advice. As noted in [BC10], in the case of VGB this is without loss of generality. However, for VBB, universal simulation does not follow from non-universal simulation, and we do not know how to extend our results to this setting.*

5 SIO is Equivalent to VBB Obfuscation for Concentrated Distributions

In this section we show that SIO for a given collection $\mathcal{C}$ is not only equivalent to VGB for $\mathcal{C}$, but is also equivalent to requiring average-case VBB for any concentrated distribution on $\mathcal{C}$.

Theorem 5.1. *Let $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$ be a circuit collection, and let $\mathcal{O}$ be an obfuscation algorithm for $\mathcal{C}$. Then the following two conditions are equivalent:*

1. *$\mathcal{O}$ is a strong indistinguishability obfuscator for $\mathcal{C}$.*
2. *For any concentrated sub-collection $\mathcal{B} = \bigcup_{n\in\mathbb{N}} \mathcal{B}_n \subseteq \mathcal{C}$, $\mathcal{O}$ is average-case VBB for $\mathcal{B}$.*

Before proving the theorem, we first prove the following useful lemma regarding an alternative definition for average-case obfuscation for concentrated distributions. The lemma, implicitly proven in [BBC$^+$14] for the special case of evasive distributions, shows that, for concentrated distributions, (average-case) VBB obfuscation admits a universal simulator that essentially runs the adversary on an obfuscation of a random circuit.

Lemma 5.1. *Let $\tilde{\mathcal{C}} = \bigcup_{n\in\mathbb{N}} \tilde{\mathcal{C}}_n$ be a concentrated distribution ensemble on a circuit collection $\mathcal{C} = \bigcup_{n\in\mathbb{N}} \mathcal{C}_n$. Then $\mathcal{O}$ is an average-case VBB obfuscator for $\tilde{\mathcal{C}}$ if and only if for any poly-size $\mathcal{A}$ there exists a negligible $\mu(\cdot)$, such that for any $n \in \mathbb{N}$, and any predicate $\pi : \mathcal{C}_n \to \{0,1\}$,*

$$\left| \Pr_{C\leftarrow\tilde{\mathcal{C}}_n} [\mathcal{A}(\mathcal{O}(C)) = \pi(C)] - \Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n} \left[\mathcal{A}(\mathcal{O}(C)) = \pi(C')\right] \right| \leq \mu(n) \ . \tag{1}$$

Proof. For the first direction, assume that $\mathcal{O}$ is an average-case VBB obfuscator for $\tilde{\mathcal{C}}$. Fix any poly-size $\mathcal{A}$. Fix any polynomial $p(\cdot)$, and let $\mathcal{S}$ be an average-case VBB simulator for $(\mathcal{A}, p)$ according to Definitions 2.2,2.4. Then, for any $n \in \mathbb{N}$ and predicate $\pi : \mathcal{C}_n \to \{0,1\}$,

$$\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi(C)\right]-\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C'))=\pi(C)\right]\right|\leq$$
$$\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi(C)\right]-\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[S^C(1^n)=\pi(C)\right]\right|+$$
$$\left|\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi(C')\right]-\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[S^C(1^n)=\pi(C')\right]\right|+$$
$$\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[S^C(1^n)=\pi(C)\right]-\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[S^C(1^n)=\pi(C')\right]\right|\leq$$
$$\frac{1}{p(n)}+\frac{1}{p(n)}+2q(n)\max_{x\in\{0,1\}^n}\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[C(x)\neq\mathrm{maj}_{\tilde{\mathcal{C}}_n}(x)\right]\leq$$
$$\frac{2}{p(n)}+2q(n)\cdot\mu(n)\ ,$$

where $q(n)$ is the polynomial bounding the number of queries made by $\mathcal{S}$, and $\mu(n)$ is the negligible concentration of $\tilde{\mathcal{C}}_n$. Note that the above holds for every polynomial p, which implies that

$$\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi(C)\right]-\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C'))=\pi(C)\right]\right|=\mathrm{negl}(n),$$

as desired.

For the second direction, let $\mathcal{S}_n$ be a simulator that outputs 1 with probability

$$p_n\triangleq\Pr_{\mathcal{A},\mathcal{O},C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right],$$

where p_n is non-uniformly hardwired into $\mathcal{S}_n$. By Equation (1), $\mathcal{S}$ is a valid simulator for $\tilde{\mathcal{C}}$. □

Proof of Theorem 5.1. We first prove that (2) implies (1). Specifically, assume that (1) does not hold, we show that (2) also does not hold. If (1) does not hold then there exist two concentrated ensembles $\tilde{\mathcal{C}}^0,\tilde{\mathcal{C}}^1$ such that $\mathrm{maj}_{\tilde{\mathcal{C}}^0_n}\equiv\mathrm{maj}_{\tilde{\mathcal{C}}^1_n}$, a poly-size adversary $\mathcal{A}$, and a noticeable function δ such that, for infinitely many $n\in\mathbb{N}^*\subseteq\mathbb{N}$,

$$\Pr_{C\leftarrow\tilde{\mathcal{C}}^0_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{C}}^1_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\geq\delta(n)$$

(the absolute value is discarded WLOG, by flipping $\mathcal{A}$'s output if necessary).

For any circuit $C\in\mathcal{C}_n$, let $p(C)=\Pr_{\mathcal{A},\mathcal{O}}[\mathcal{A}(\mathcal{O}(C))=1]$, and for a distribution $\tilde{D}$ on $\mathcal{C}_n$, let $p(\tilde{D})=\mathbb{E}_{C\leftarrow\tilde{D}}[p(C)]$. Then, for any $n\in\mathbb{N}^*$,

$$p(\tilde{\mathcal{C}}^0_n)-p(\tilde{\mathcal{C}}^1_n)\geq\delta(n)\ .$$

Next, denote

$$S^0_n=\left\{C:p(C)\geq p(\tilde{\mathcal{C}}^0_n)-\delta(n)/4\right\}$$
$$S^1_n=\left\{C:p(C)\leq p(\tilde{\mathcal{C}}^1_n)+\delta(n)/4\right\}\ .$$

Note that

$$
\begin{aligned}
p(\tilde{\mathcal{C}}_n^0) &\leq \Pr_{C \leftarrow \tilde{\mathcal{C}}_n^0}[C \in S_n^0] + p(\tilde{\mathcal{C}}_n^0) - \delta(n)/4 \\
&\Rightarrow \boxed{\Pr_{C \leftarrow \tilde{\mathcal{C}}_n^0}[C \in S_n^0] \geq \delta(n)/4}, \\
p(\tilde{\mathcal{C}}_n^1) &\geq \Pr_{C \leftarrow \tilde{\mathcal{C}}_n^1}[C \notin S_n^1] \cdot \Big(p(\tilde{\mathcal{C}}_n^1) + \delta(n)/4\Big) \\
&\Rightarrow \boxed{\Pr_{C \leftarrow \tilde{\mathcal{C}}_n^1}[C \in S_n^1] \geq 1 - \frac{p(\tilde{\mathcal{C}}_n^1)}{p(\tilde{\mathcal{C}}_n^1) + \delta(n)/4} \geq \frac{\delta(n)/4}{p(\tilde{\mathcal{C}}_n^0)} \geq \delta(n)/4} \ .
\end{aligned}
$$

We next consider the following two distributions conditioned on the above events

$$\tilde{\mathcal{D}}_n^b := \tilde{\mathcal{C}}_n^b | S_n^b \text{ , for } b \in \{0,1\} \ .$$

Then

$$
\begin{aligned}
p(\tilde{\mathcal{D}}_n^0) - p(\tilde{\mathcal{D}}_n^1) \geq \\
(p(\tilde{\mathcal{C}}_n^0) - \delta(n)/4) - (p(\tilde{\mathcal{C}}_n^1) + \delta(n)/4) \geq \\
p(\tilde{\mathcal{C}}_n^0) - p(\tilde{\mathcal{C}}_n^1) - \delta(n)/2 \geq \delta(n)/2 \ .
\end{aligned}
$$

We now consider the distribution $\tilde{\mathcal{D}}_n = \frac{\tilde{\mathcal{D}}_n^0 + \tilde{\mathcal{D}}_n^1}{2}$ that samples from $\tilde{D}_n^b$ for a uniform $b \in \{0,1\}$. We first claim that the corresponding ensemble $\tilde{\mathcal{D}} = \bigcup_{n \in \mathbb{N}^*} \tilde{\mathcal{D}}_n$ is concentrated. Indeed, since each $\tilde{\mathcal{D}}_n^b$ is distributed like $\tilde{\mathcal{C}}_n^b$, conditioned on S_n^b, and since S_n^b has noticeable density $\delta(n)/4$, it holds that $\tilde{\mathcal{D}}_n^b$ is concentrated around $\mathrm{maj}_{\tilde{\mathcal{C}}_n^b}$. Thus,

$$\mathrm{maj}_{\tilde{\mathcal{D}}_n^0} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}_n^0} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}_n^1} \equiv \mathrm{maj}_{\tilde{\mathcal{D}}_n^1} \ ,$$

and since $\mathcal{D}$ is the average of $\mathcal{D}_n^0, \mathcal{D}_n^1$, it is also concentrated and

$$\mathrm{maj}_{\tilde{\mathcal{D}}_n} \equiv \mathrm{maj}_{\tilde{\mathcal{D}}_n^0} \equiv \mathrm{maj}_{\tilde{\mathcal{D}}_n^1} \ .$$

Next, define a predicate π_n on the support of $\mathcal{D}_n$ such that $\pi_n(C) = b$ if and only if $C \in S_n^b$. Then, it

holds that

$$\begin{aligned}
&\Pr_{C,C'\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C'))=\pi_n(C)\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi_n(C)\right]=\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=0\right]\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=0\right]+\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=1\right]\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=1\right]\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=0\right]\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^0}\left[\mathcal{A}(\mathcal{O}(C))=0\right]=\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=0\right]\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^0}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)+\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\pi_n(C)=1\right]\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)=\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[C\in S_n^0\right]\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^0}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)+\\
&\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[C\in S_n^1\right]\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)=\\
&\frac{1}{2}\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^0}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)+\\
&\frac{1}{2}\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)=\\
&\frac{1}{2}\left(\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^0}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{D}}_n^1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right)=\\
&\frac{1}{2}\left(p(\tilde{\mathcal{D}}_n^0)-p(\tilde{\mathcal{D}}_n^1)\right)\geq\\
&\delta(n)/4\ .
\end{aligned}$$

By Lemma 5.1, this contradicts the fact that $\mathcal{O}$ is average-case VBB for the concentrated ensemble $\tilde{\mathcal{D}}$.

We next prove that (1) implies (2). Fix any concentrated ensemble $\tilde{\mathcal{C}}$ on the collection $\mathcal{C}$, and assume that (2) does not hold, we show that (1) also does not hold. By Lemma 5.1, if (1) does not hold, there exists a poly-size $\mathcal{A}$ and noticeable $\delta(\cdot)$ such that for infinitely many $n\in\mathbb{N}^*\subseteq\mathbb{N}$ and predicates $\pi_n:\mathcal{C}_n\to\{0,1\}$, it holds that

$$\begin{aligned}
&\delta(n)\leq\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=\pi_n(C)\right]-\Pr_{C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C'))=\pi_n(C)\right]\right|\leq\\
&\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\pi_n(C)=0\right]\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n:\pi_n(C)=0}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right|+\\
&\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\pi_n(C)=1\right]\left|\Pr_{C\leftarrow\tilde{\mathcal{C}}_n:\pi_n(C)=1}\left[\mathcal{A}(\mathcal{O}(C))=1\right]-\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(\mathcal{O}(C))=1\right]\right|\ .
\end{aligned}$$

Then, for infinitely many $n\in\mathbb{N}^*$, one of the two above summands is at least $\delta(n)/2$, let us assume WLOG that it is the first (the proof is similar in the second case). Now consider the distribution $\tilde{\mathcal{C}}_n^0=\left\{C\in\tilde{\mathcal{C}}_n:\pi_n(C)=0\right\}$. Since $\Pr_{C\leftarrow\tilde{\mathcal{C}}_n}[\pi_n(C)=0]\geq\frac{\delta(n)}{2}$, it holds that $\tilde{\mathcal{C}}_n^0$, like $\tilde{\mathcal{C}}_n$ is concentrated

around $\mathrm{maj}_{\tilde{\mathcal{C}}_n}$. Moreover,

$$\delta(n)/2 \leq \left| \Pr_{C \leftarrow \mathcal{C}_n^0} [\mathcal{A}(\mathcal{O}(C)) = 1] - \Pr_{C \leftarrow \tilde{\mathcal{C}}_n} [\mathcal{A}(\mathcal{O}(C)) = 1] \right| \ ,$$

implying that strong indistinguishability, required in (2), does not hold. □

5.1 Equivalence of VBB Obfuscation for Concentrated and Evasive Distributions

We complete this section by showing that indistinguishability obfuscation, plus (average-case) VBB obfuscation for evasive distributions implies (average-case) VBB obfuscation for *concentrated* distributions.

We start by noting the following fact.

Claim 5.1. *Let $\tilde{S}$ be a $\frac{1}{3}$-concentrated distribution over boolean circuits where each boolean circuit $C \in \mathrm{supp}(\tilde{S})$ is defined over $\{0,1\}^n$ and is of depth at most d and size at most ℓ. Then the majority function $\mathrm{maj}_{\tilde{S}}$ can be computed by a (non-uniform) circuit of size $O(n \cdot \ell)$ and depth $O(\log n + d)$. Also, if $\tilde{S}$ is samplable by a circuit of size s, such a majority circuit can be sampled, with overwhelming probability $1 - 2^{-\Omega(n)}$, by a circuit of size $O(n \cdot s)$.*

Proof. Since $\tilde{S}$ is $\frac{1}{3}$-concentrated, by a Chernoff bound and a union bound on 2^n inputs, the majority of $O(n)$ random circuits from $\tilde{S}$ computes $\mathrm{maj}_{\tilde{S}}$ with probability $1 - 2^{-\Omega(n)}$. □

We next state the equivalence lemma. For a circuit C of size at most ℓ, we denote by $[C]_\ell$ a canonically zero-padded version of C of size ℓ. For a collection $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$, and functions $\ell(\cdot), d(\cdot)$, we denote by $[\mathcal{C}]_\ell^d$ the class of circuits where each $C \in \mathcal{C}$ computes the same function as some $C \in \mathcal{C}_n$ and is of size $\ell(n)$, and depth $d(n)$. Let M_n be a poly-size circuit computing the majority $\mathrm{maj}_{\tilde{\mathcal{C}}_n}$ over $\tilde{\mathcal{C}}_n$. We denote by $\tilde{\mathcal{C}} \oplus M_n$ the distribution ensemble $\bigcup_{n \in \mathbb{N}} \left\{ C \oplus M_n : C \leftarrow \tilde{\mathcal{C}}_n \right\}$. Observe that $\tilde{\mathcal{C}} \oplus M_n$ is evasive.

Lemma 5.2. *Let $\mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$ be a circuit collection where each $C \in \mathcal{C}_n$ is of poly-size $\ell(n)$ and depth at most d, and let $\tilde{\mathcal{C}}$ be a concentrated distribution ensemble on $\mathcal{C}$. For any $n \in \mathbb{N}$ let M_n be a circuit of size $O(\ell(n) \cdot n)$ and depth $O(\log n + d(n))$ that computes $\mathrm{maj}_{\tilde{\mathcal{C}}}$. Assume that $e\mathcal{O}$ is an average-case VBB obfuscator for the evasive distribution ensemble $\tilde{\mathcal{C}} \oplus M_n$ that blows up the size of any circuit C by some polynomial $B(\cdot)$, i.e. $|e\mathcal{O}(C)| = B(|C|)$. Then there exist a polynomial $\ell'(n)$ and a function $d'(n) = O(d(n) + \log n)$, depending only on (ℓ, B, d'), such that if $i\mathcal{O}$ is an indistinguishability obfuscator for $[\mathcal{C}]_{\ell'}^{d'}$, then the obfuscator $c\mathcal{O}$, given by*

$$c\mathcal{O}(C) \leftarrow i\mathcal{O}([C]_{\ell'(n)})$$

is an average-case VBB obfuscator for $\tilde{\mathcal{C}}$.

Proof. Let M_n be the circuit of size $O(\ell(n) \cdot n)$ that computes $\mathrm{maj}_{\tilde{\mathcal{C}}_n}$. For $C \in \mathcal{C}_n$, consider the circuit

$$C_r := M_n \oplus e\mathcal{O}(C \oplus M_n; r)) \ ,$$

which computes the same function as C, but in a different way—it has hardwired an obfuscation $e\mathcal{O}(C \oplus M_n; r)$, using some randomness r, that computes its difference from the majority; it first runs this obfuscation on the given input, and then computes again the difference from the majority, resulting back in $C(x)$. We let $\ell'(n)$ be the size of C_r and $d'(n)$ be its depth. Note that these, indeed, only depend on ℓ, b, d and $d'(n) = O(\mathsf{depth}(C) + \mathsf{depth}(M_n)) = O(d(n) + \log n)$.

Next, applying first the IO guarantee and then the guarantee that $e\mathcal{O}$ is an average-case VBB obfuscator for the evasive ensemble $\bigcup_{n\in\mathbb{N}} M_n \oplus \tilde{\mathcal{C}}_n$, as given by Lemma 5.1, we have

$$\begin{aligned}
&\Pr_{\mathcal{A},c\mathcal{O},C\leftarrow\tilde{\mathcal{C}}_n}[\mathcal{A}(c\mathcal{O}(C)) = \pi(C)] = \\
&\Pr_{\mathcal{A},i\mathcal{O},C\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(i\mathcal{O}([C]_{\ell'(n)})) = \pi(C)\right] = \\
&\Pr_{\mathcal{A},i\mathcal{O},r,C\leftarrow\tilde{\mathcal{C}}_n}[\mathcal{A}(i\mathcal{O}(C_r)) = \pi(C)] \pm n^{-\omega(1)} = \\
&\Pr_{\mathcal{A},i\mathcal{O},r,C\leftarrow\tilde{\mathcal{C}}_n}[\mathcal{A}(i\mathcal{O}(M_n \oplus e\mathcal{O}(C \oplus M_n; r))) = \pi(C)] \pm n^{-\omega(1)} = \\
&\Pr_{\mathcal{A},i\mathcal{O},r,C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(i\mathcal{O}(M_n \oplus e\mathcal{O}(C \oplus M_n; r))) = \pi(C')\right] \pm n^{-\omega(1)} = \\
&\Pr_{\mathcal{A},i\mathcal{O},r,C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(i\mathcal{O}(C_r)) = \pi(C')\right] \pm n^{-\omega(1)} = \\
&\Pr_{\mathcal{A},i\mathcal{O},C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(i\mathcal{O}([C]_{\ell'(n)})) = \pi(C')\right] \pm n^{-\omega(1)} = \\
&\Pr_{\mathcal{A},c\mathcal{O},C,C'\leftarrow\tilde{\mathcal{C}}_n}\left[\mathcal{A}(c\mathcal{O}(C)) = \pi(C')\right] \pm n^{-\omega(1)} \ .
\end{aligned}$$

Thus, again by Lemma 5.1, we deduce that $c\mathcal{O}$ is an average-case VBB obfuscator for the concentrated ensemble $\tilde{\mathcal{C}}$.

□

6 From Semantically-Secure Graded Encodings to SIO for NC^1

In this section we show that a variant of the Pass et al. [PTS13] notion of semantically-secure graded encoding schemes, together with *any* ideal graded encoding obfuscation (i.e., any obfuscation that is virtual-black-box secure in the ideal encoding model) for a class $\mathcal{C}$ of circuits, implies *strong indistinguishability obfuscation* for $\mathcal{C}$. Combined with the recent ideal-graded-encoding obfuscators for NC^1 [BR13, BGK+13], we obtain SIO for NC^1.

6.1 Graded Encodings

The notion of graded (multi-linear) encoding schemes was originally introduced by Garg, Gentry and Halevi [GGH13a]. Just as in [BR13, BGK+13, PTS13], we rely on "set-based" graded encoding. Following [GGH+13b, BGK+13, PTS13], we enable anyone with the secret parameter to encode any elements (as opposed to just random elements).

Definition 6.1 (Graded Encoded Scheme). *A graded encoding scheme $\mathcal{E}$ is associated with a tuple of PPT algorithms* $(\mathsf{InstGen}_{\mathcal{E}}, \mathsf{Encode}_{\mathcal{E}}, \mathsf{Add}_{\mathcal{E}}, \mathsf{Neg}_{\mathcal{E}}, \mathsf{Mult}_{\mathcal{E}}, \mathsf{isZero}_{\mathcal{E}})$ *which behave as follows:*

- **Instance Generation:** $\mathsf{InstGen}_{\mathcal{E}}$ *takes as input the security parameter 1^n and multi-linearity parameter k, and outputs secret parameters sp and public parameters pp, which include a description of a ring R and the integer k. For simplicity, we consider graded encoding schemes where R is $\mathbb{Z}_p$ and p is a prime of size exponential in n. The public parameters pp determine a collection of sets $\{E_S^\alpha : S \subseteq [k], \alpha \in R\}$.*
- **Encoding:** $\mathsf{Encode}_{\mathcal{E}}$ *takes as input the secret parameters sp, an element $\alpha \in R$ and set $S \subseteq [k]$, and outputs a random encoding in E_S^α.*

- **Addition:** $\mathsf{Add}_{\mathcal{E}}$ *takes as input the public parameters* pp *and encodings* $u_1 \in E_{S_1}^{\alpha_1}$ *and* $u_2 \in E_{S_2}^{\alpha_2}$*, and outputs an encoding in* $E_S^{\alpha_1+\alpha_2}$ *if* $S_1 = S_2 = S$*, and outputs* $\perp$ *otherwise. We denote the output by* $u_1 \oplus u_2$*.*

- **Subtraction:** $\mathsf{Sub}_{\mathcal{E}}$ *takes as input the public parameters* pp *and encodings* $u_1 \in E_{S_1}^{\alpha_1}$ *and* $u_2 \in E_{S_2}^{\alpha_2}$*, and outputs an encoding in* $E_S^{\alpha_1-\alpha_2}$ *if* $S_1 = S_2 = S$*, and outputs* $\perp$ *otherwise. We denote the output by* $u_1 \ominus u_2$*.*

- **Multiplication:** $\mathsf{Mult}_{\mathcal{E}}$ *takes as input the public parameters* pp *and encodings* $u_1 \in E_{S_1}^{\alpha_1}$ *and* $u_2 \in E_{S_2}^{\alpha_2}$*, and outputs an encoding in* $E_{S_1 \cup S_2}^{\alpha_1 \cdot \alpha_2}$ *if* $S_1 \cap \mathcal{S}_2 = \emptyset$*; and outputs* $\perp$ *otherwise. We denote the output by* $u_1 \otimes u_2$*.*

- **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* $[k]$*.*

Next, we recall the definition of set-respecting arithmetic circuits from [PTS13]. A set-respecting arithmetic circuit represents the type of algebraic circuits that can be evaluated on a set of "generic" encodings.

Definition 6.2 (Set-respecting arithmetic circuits ([PTS13]))**.** *Given a level* $k \in \mathbb{N}$*, and a vector of sets* $\vec{S} \in (2^{[k]})^{\ell}$*, we say that an arithmetic circuit taking* ℓ *inputs is* $\vec{S}$*-respecting if there exists a function* Tag *from the wires of* C *to* $2^{[k]}$ *such that the following holds:*

- *For every* $i \in [\ell]$*, the* i*-th input wire* w_{in}^i *satisfies* $\mathsf{Tag}(w_{in}^i) = \vec{S}[i]$*.*
- *Every* $+$ *or* $-$ *gate in* C *connecting input wires* u *and* v *to an output wire* w*, satisfies* $\mathsf{Tag}(u) = \mathsf{Tag}(v) = \mathsf{Tag}(w)$*.*
- *Every* $\times$ *gate in* C *connecting input wires* u *and* v *to output wire* w*, satisfies* $\mathsf{Tag}(u) \cap \mathsf{Tag}(v) = \emptyset$ *and* $\mathsf{Tag}(u) \cup \mathsf{Tag}(v) = \mathsf{Tag}(w)$*.*
- *The output wire* w_{out} *satisfies* $\mathsf{Tag}(w_{out}) = [k]$*.*

6.2 Ideal Graded Encoding Obfuscation

Barak et al. [BGK+13] construct a virtual black-box obfuscator for NC^1 in the ideal graded encoding model. In what follows we define this ideal model and the notion of obfuscation in this model.

In the ideal graded encoding model algorithms have access to an ideal graded encoding oracle $\mathcal{M}$. We recall the definitions of this oracle as presented in [PTS13].

Definition 6.3 (Ideal graded encoding oracle $\mathcal{M}$ ([PTS13]))**.** *For a prime* p*, a level* $k \in \mathbb{N}$*, a vector of sets* $\vec{S} \in (2^{[k]})^{\ell}$ *and a vector of elements* $\vec{m} \in \mathbb{Z}_p^{\ell}$*, the oracle* $\mathcal{M}(R, k, \vec{S}, \vec{m})$ *is defined as follows: for every query* C*, if* C *is not a description of an* $\vec{S}$*-respecting arithmetic circuit,* $\mathcal{M}$ *outputs* $\perp$*. Otherwise,* $\mathcal{M}$ *evaluates* C *on* $\vec{m}$ *and outputs* 1 *if* C *evaluates to* 0*, and outputs* 0 *otherwise.*

Next, we define virtual-black-box obfuscation in the ideal graded encoding model. Here an obfuscation of a circuit consists of a vector of sets $\vec{S}$, and a vector of elements $\vec{m}$ used to initialize a graded encoding oracle $\mathcal{M}$. The honest evaluator and the attacker have oracle access to $\mathcal{M}$.

Definition 6.4 (Ideal graded encoding obfuscation)**.** *Let* $\mathcal{E}$ *be a graded encoding scheme. A PPT algorithm* $\mathcal{O}$ *is a virtual black-box (VBB) obfuscator in the ideal graded encoding model, for a family of poly-size circuits* $\mathcal{C} = \{C_n\}_{n \in \mathbb{N}}$*, if it satisfies the following requirements:*

- *Functionality: There exist polynomials $k = k(n)$ and $\ell = \ell(n)$ and there exists a poly-time oracle machine $\mathsf{Eval}_{\mathcal{O}}$ that satisfy the following: for every $n \in \mathbb{N}$, for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, and for every $C \in \mathcal{C}_n$, the obfuscator $\mathcal{O}(1^n, p, C)$ outputs a vector of sets $\vec{S} \in (2^{[k]})^{\ell}$ and a vector of elements $\vec{m} \in \mathbb{Z}_p^{\ell}$ such that for every input x to C:*

$$\mathsf{Eval}_{\mathcal{O}}^{\mathcal{M}(p,k,\vec{S},\vec{m})}(x) = C(x) \ .$$

  We require that the vector of sets $\vec{S}$ depends only on the security parameter n an not on the circuit C or the randomness of $\mathcal{O}$.

- *Virtual black-box: For every polynomial functions q, s and for every (unbounded) oracle machine $\mathcal{A}$ (called the algebraic adversary) making at most $q(n)$ queries, where every query describes an arithmetic circuit of size at most $s(n)$, there exist a PPT oracle machine $\mathcal{S}$ (called the simulator), and negligible function μ, such that for every $n \in \mathbb{N}$, every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, and for every $C \in \mathcal{C}_n$:*

$$\left| \begin{array}{l} \Pr[(\vec{S}, \vec{m}) \leftarrow \mathcal{O}(1^n, p, C); \mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m})}(n, p) = 1] - \\ \Pr[\mathcal{S}^{\mathcal{A},C}(1^n, p) = 1] \end{array} \right| \leq \mu(n) \ ,$$

  where the probabilities are over the coins of the obfuscator $\mathcal{O}$, the adversary $\mathcal{A}$ and the simulator $\mathcal{S}$. The notation $\mathcal{S}^{\mathcal{A}}$ means that $\mathcal{S}$ gets oracle access to $\mathcal{A}$, when answering $\mathcal{A}$'s oracle queries.

Remark 6.1. *In the above definition we do not allow $\mathcal{O}$ to choose the vector of sets $\vec{S}$ depending on its randomness or on the circuit C. This restriction simplifies the presentation of our results. Specifically, we rely on the fact that obfuscation of different circuits of the same size use the same vector $\vec{S}$. However, all our results hold also with respect to the more general definition of ideal graded encoding obfuscation where the choice of $\vec{S}$ is not restricted. We note that all existing constructions of ideal obfuscation [BGK+13, BR13, PTS13] meet the functionality requirement in its restricted form.*

Remark 6.2 (Ideal obfuscation in [BGK+13].)**.** *The syntax for the obfuscator defined above is different from the one in [BGK+13], and is adapted to the way that ideal graded oracles are dealt with in this work. A more essential difference is that Barak et al. state their final result for bounded algebraic adversaries, whereas we consider unbounded ones (but with a bounded number of oracle queries). Against such bounded algebraic adversaries, they achieve ideal graded encodings obfuscation for all poly-size circuits. For unbounded algebraic adversaries, as considered in this work, their result only holds for NC^1.*

Theorem 6.1 ([BGK+13])**.** *There exists an ideal graded encoding obfuscation for every function family in NC^1.*

6.3 Semantic Security

We formulate a simple and natural variant of semantic security. Although it appears to be somewhat stronger than the actual notion considered by Pass et al. [PTS13], we find it natural and appealing. Section 7 considers a number of relaxations of this basic notion and their relative security.

Central to the definition of semantic security is the notion of an *admissible message sampler* (analogous to *respecting message samplers* in [PTS13]). A message sampler is admissible if it samples two vectors of elements $\vec{m}_0, \vec{m}_1$ that cannot be distinguished by an unbounded adversary that can only access $\vec{m}_0$ or $\vec{m}_1$ by making a polynomial number of queries to an ideal graded encoding oracle $\mathcal{M}$ (Definition 6.3).

Definition 6.5 (Efficient message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme and let $k = k(n)$ and $\ell = \ell(n)$ be polynomials. A PPT algorithm $\mathbb{M}$ is an efficient (k, ℓ)-message sampler if for every $n \in \mathbb{N}$, and every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, $\mathbb{M}(1^n, p)$ a vector of sets $\vec{S} \in (2^{[k]})^{\ell}$ and two vectors of elements $\vec{m}_0, \vec{m}_1 \in \mathbb{Z}_p^{\ell}$. We require that the vector of sets $\vec{S}$ depends only on the security parameter n an not on randomness of $\mathbb{M}$.*

Remark 6.3. *In the above definition we do not allow $\mathbb{M}$ to choose the vector of sets $\vec{S}$ depending on its randomness or on the circuit C. This restriction simplifies the presentation of our results. However, all our results hold also with respect to the more general definition of message samplers where $\vec{S}$ can be sampled from some distribution.*

Definition 6.6 (Admissible message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme and let $k = k(n)$ and $\ell = \ell(n)$ be polynomials. A (k, ℓ)-message sampler $\mathbb{M}$ (Definition 6.5) is admissible if for every polynomial functions q and s, and for every (unbounded) oracle machine $\mathcal{A}$ (called the algebraic adversary) making at most $q(n)$ oracle queries, where every query describes an arithmetic circuit of size at most $s(n)$, there exists a negligible function μ such that for every $n \in \mathbb{N}$ and for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$, including a prime p,*

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, p) \end{array} ; \mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(n, p) = b\right] \leq \frac{1}{2} + \mu(n) \ ,$$

where the probability is also over the coins of $\mathcal{A}$.

Loosely speaking, a graded encoding scheme is semantically secure if for a tuple $(\vec{S}, \vec{m}_0, \vec{m}_1)$ generated by an admissible message sampler, given encodings of $\vec{m}_b$ with respect to the sets $\vec{S}$, it is hard to predict b with non-negligible advantage in polynomial time.

Remark 6.4 (Inefficient message samplers)**.** *For most of our results we need to rely on a stronger notion of semantic security that allows for computationally unbounded admissible message samplers. Since the message sampler $\mathbb{M}$ in Definition 6.5 takes as input the prime p, or more generally, the description of a ring R associated with the public parameters of the graded encoding, simply allowing $\mathbb{M}$ to be unbounded may result in an unachievable definition. Specifically, consider a graded encoding scheme where it is possible to recover the secret parameters sp of the graded encoding scheme from the description of the ring R in unbounded time.*[8] *An inefficient $\mathbb{M}$ may recover sp and sample encodings that reveal sp. $\mathbb{M}$ may still be admissible since knowing the secret parameters gives no advantage to the algebraic adversary, however given sp it is possible to distinguish encodings of the sampled elements (for any non-trivial $\mathbb{M}$).*

Instead, in Definition 6.7 we keep the sampler $\mathbb{M}$ efficient, but we give it auxiliary input that is sampled by an inefficient algorithm. Importantly, we do not give the auxiliary input the description of the ring.

Definition 6.7 (Inefficient message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme and let $k = k(n)$ and $\ell = \ell(n)$ be polynomials. An unbounded (k, ℓ)-message sampler is defined by a PPT algorithm $\mathbb{M}$ and an unbounded auxiliary input sampler $\mathbb{Z}$. We require that there exist a polynomial q such that for every $n \in \mathbb{N}$, and every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, $|\mathbb{Z}(n)| < q(n)$, and $\mathbb{M}(1^n, \mathbb{Z}(n), p)$ outputs a vector of sets $\vec{S} \in (2^{[k]})^{\ell}$ and two vectors of elements $\vec{m}_0, \vec{m}_1 \in \mathbb{Z}_p^{\ell}$. We require that the vector of sets $\vec{S}$ depends only on the security parameter n and not on randomness of $\mathbb{M}$ or $\mathbb{Z}$.*

The definition of admissability remains as in Definition 6.6. In what follows, we only consider semantic security with respect to inefficient admissible message samplers (unless we explicitly state

[8] We note that this is not the case for existing candidate construction of graded encodings [GGH13a, CLT13].

otherwise). We abuse notation, and when we refer to admissible message samplers we mean inefficient ones, as in Definition 6.7.

Definition 6.8 (Semantically-secure graded encoding)**.** *A graded encoding scheme $\mathcal{E}$ is semantically-secure if for every polynomials $k = k(n), \ell = \ell(n)$, every admissible (k, ℓ)-message sampler $(\mathbb{Z}, \mathbb{M})$, and every poly-size adversary $\mathcal{A}$, there exist a negligible function μ such that for every $n \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \end{array} ; A\left(pp, (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}_b[i], \vec{S}[i]))_{i\in[\ell(n)]}\right) = b\right] \leq \frac{1}{2} + \mu(n) \ ,$$

where p is described in the public parameters pp and the probability is also over the coins of $\mathcal{A}$.

6.4 Obfuscation from Semantically-Secure Graded Encodings

We show that semantically-secure graded encodings imply SIO, and other forms of obfuscation as a corollary of our result from Section 4.

Proposition 6.1. *Assume there exists a semantically-secure graded encoding scheme (Definition 6.8), and assume there exists an ideal graded encoding obfuscation (Definition 6.4) for a circuit class $\mathcal{C}$. Then there exists a strong indistinguishability obfuscator for the circuit class $\mathcal{C}$, in the plain model (Definition 3.2).*

As a corollary of Theorem 6.1, Proposition 6.1, and the transformation from (standard) IO for NC^1 to (standard) IO for all poly-size circuit classes [GGH+13b], we obtain the following theorem.

Theorem 6.2. *Assume there exists a semantically-secure graded encoding scheme. Then there exist:*

1. *SIO, for any circuit class in NC^1,*
2. *(standard) IO, for any poly-size circuit class, assuming also fully-homomorphic encryption (with decryption in NC^1).*

As a corollary of the above theorem and of our results from Section 4 we obtain the following theorem.

Theorem 6.3. *Assume there exists a semantically-secure graded encoding scheme. Then there exist:*

1. *worst-case VGB for any collection in NC^1,*
2. *worst-case VGB obfuscation for the class of set circuits $\mathcal{S}^k$ for any $k = \mathrm{poly}(n)$, and VBB obfuscation for $k = O(1)$,*
3. *worst-case VGB obfuscation for the class of linear subspaces $\mathcal{L}^{d,\mathbb{F}}$ for any $d = \mathrm{poly}(n)$, and VBB obfuscation for $d = O(1)$,*
4. *worst-case VBB for any efficiently samplable collection of connected circuits in NC^1, in particular, for Hamming balls and conjunctions.*

We now turn to give a proof sketch of Proposition 6.1.

Proof of Proposition 6.1. Let $\mathcal{C}$ be a class of poly-size circuits, let $\mathcal{O}$ be an ideal graded encoding obfuscator for $\mathcal{C}$.

The obfuscator $\tilde{\mathcal{O}}$ for $\mathcal{C}$: Let $\ell = \ell(n)$ and $k = k(n)$ be the polynomials given by the ideal graded encoding obfuscator $\mathcal{O}$ (Definition 6.4). Let $\mathcal{E}$ be a semantically-secure multi-linear encoding scheme

(Definition 6.8). Given $C \in \mathcal{C}_n$, $\tilde{\mathcal{O}}$ samples parameters $(sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)})$ for the encoding scheme, including a prime p. $\tilde{\mathcal{O}}$ then runs $\mathcal{O}(1^n, p, C)$ which outputs a vector of sets $\vec{S} \in (2^{[k(n)]})^{\ell(n)}$, and a vector of elements $\vec{m} \in \mathbb{Z}_p^{\ell(n)}$, such that for every input x, $\mathsf{Eval}_{\mathcal{O}}^{\mathcal{M}(p,k(n),\vec{S},\vec{m})}(x) = C(x)$. Finally, $\tilde{\mathcal{O}}$ outputs an obfuscated circuit that has the public parameters pp, the sets $\vec{S}$, and the encodings

$$\left(\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}[i], \vec{S}[i])\right)_{i \in [\ell(n)]} \ ,$$

hardcoded into it. The obfuscated circuit emulates the evaluation procedure of the ideal graded encoding obfuscation $\mathsf{Eval}_{\mathcal{O}}$, where the queries to the ideal oracle $\mathcal{M}$ are answered by performing algebraic operations directly on the set of encodings.

Functionality and indistinguishability: The functionality of $\tilde{\mathcal{O}}$ follows readily from that of $\mathcal{O}$ and $\mathcal{E}$. We now argue strong indistinguishability based on the semantic security of the encoding scheme. Let $\tilde{\mathcal{C}}^0, \tilde{\mathcal{C}}^1$ be two concentrated distribution ensembles on $\mathcal{C}$ such that $\mathrm{maj}_{\tilde{\mathcal{C}}_n^0} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}_n^1}$, and let $\mathcal{D}$ be any poly-size distinguisher. We show that $\mathcal{D}$ cannot distinguish whether it is given an obfuscation $\tilde{\mathcal{O}}(C_0)$ or $\tilde{\mathcal{O}}(C_1)$, for $(C_0, C_1) \leftarrow (\tilde{\mathcal{C}}_n^0, \tilde{\mathcal{C}}_n^1)$, with non-negligible advantage.

Assume towards contradiction that $\mathcal{D}$ can predict whether it is given $\tilde{\mathcal{O}}(C_0)$ or $\tilde{\mathcal{O}}(C_1)$ with a noticeable advantage over $\frac{1}{2}$. We construct an admissible (k, ℓ)-message sampler $(\mathbb{Z}, \mathbb{M})$, and show that together with $\mathcal{D}$ they violate the semantic security of $\mathcal{E}$. The auxiliary input sampler $\mathbb{Z}$ samples $(C_0, C_1) \leftarrow (\tilde{\mathcal{C}}_n^0, \tilde{\mathcal{C}}_n^1)$ (note that if $\tilde{\mathcal{C}}_n^0, \tilde{\mathcal{C}}_n^1$ are not efficiently samplable, $\mathbb{Z}$ is inefficient). The message sampler $\mathbb{M}$, given a prime p, and the auxiliary input (C_0, C_1) executes the ideal graded encoding obfuscator $\mathcal{O}(1^n, p, \cdot)$ on each of the circuits, and obtains a vector of sets $\vec{S}$ and two vectors of messages $\vec{m}_0, \vec{m}_1$, which it then outputs. (Here we use the fact that $\mathcal{O}$ outputs the same the vector of sets $\vec{S}$ in both executions. See Remark 6.1.)

To argue that the sampler $(\mathbb{Z}, \mathbb{M})$ is admissible, we need to show that the following holds for every (unbounded) oracle machine $\mathcal{A}$ (the algebraic adversary) making only polynomially many oracle queries, where every query describes an arithmetic circuit of polynomial size, for every $n \in \mathbb{N}$, and for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p:

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \end{array} ; \mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(n, p) = b\right] \leq \frac{1}{2} + \mathrm{negl}(n) \ ,$$

where the probability is also over the coins of $\mathcal{A}$.

This follows from the ideal virtual-black-box security of $\mathcal{O}$. Indeed, for any (algebraic) $\mathcal{A}$ as above, let $q(n), s(n)$ be the polynomial bounds on its number of queries and their size, and let $\mathcal{S}$ be its virtual-black-box simulator (according to Definition 6.4). Therefore,

$$\begin{aligned} &\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \end{array} ; \mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(n, p) = b\right] \leq \\ &\Pr\left[b \leftarrow \{0,1\}; (C_0, C_1) \leftarrow (\tilde{\mathcal{C}}_n^0, \tilde{\mathcal{C}}_n^1); \mathcal{S}^{\mathcal{A}, C_b}(1^n) = b\right] + \mathrm{negl}(n) \leq \\ &\frac{1}{2} + q(n) \cdot \nu(n) + \mathrm{negl}(n) = \frac{1}{2} + \mathrm{negl}(n) \ , \end{aligned}$$

where $\nu(n) = \max(\nu_0(n), \nu_1(n))$, and $\nu_b(n) = \mathrm{negl}(n)$ is the negligible concentration measure of $\mathcal{C}_n^b$ around $\mathrm{maj}_{\tilde{\mathcal{C}}_n^0} \equiv \mathrm{maj}_{\tilde{\mathcal{C}}_n^1}$.

It is left to note that, by the construction of $(\mathbb{Z}, \mathbb{M})$ and the assumption that $\mathcal{D}$ predicts b with noticeable advantage (given $\tilde{\mathcal{O}}(C_b)$ for a random b), $\mathcal{D}$ breaks the semantic security of the encoding scheme (Definition 6.8). $\square$

7 More on Semantic Security

We discuss several possible relaxations of the semantic-security definition presented in Section 6.3 (the relaxations are defined in Section 7.1). In particular, we consider relaxations to *bounded*, *entropic*, *concentrated*, *single-message* and *large-queries* security. All these relaxations, except for bounded security, were also considered by Pass et al. [PTS13]. The relaxation to bounded and entropic semantic security are particularly significant, since there are several attacks on the unbounded or non-entropic versions of semantic security. We elaborate on these attacks in Sections 7.3 and 7.1.2.

In Section 7.2 we show that under any combination of these relaxations, the positive implications to obfuscation still hold. In certain combinations, this is done by showing an equivalence between the relaxed definition and the non-relaxed definitions whereas in others, the results are obtained by relying on properties of specific obfuscation constructions which are based on the constructions in [GGH$^+$13b, BR13, BGK$^+$13, PTS13].

In Section 7.3 we motivate bounded semantic security by describing a generic attack on the (unbounded) security of any graded encoding scheme with certain efficiency properties.

All results in this section hold for both efficient and inefficient message samplers (see Remark 6.4). For generality, we use the inefficient samplers notation of Definition 6.7.

7.1 Relaxations of Semantic-Security

7.1.1 Bounded Semantic-Security.

In the definition of semantic security we consider vectors of messages $\vec{m}_0, \vec{m}_1$ of arbitrary polynomial dimension. In *bounded* semantic-security, we relax the definition by placing a fixed polynomial bound $\ell(n)$ on the dimension of $\vec{m}_0, \vec{m}_1$. The polynomial ℓ is fixed before the parameters of the encoding scheme are chosen.

This relaxation is significant since, as we show in Section 7.3, under certain conditions, allowing the dimension to be unbounded gives rise to certain diagonalization attacks. In bounded semantic-security, such attacks are mitigated, e.g. by assuring that the description size of encoded elements (determined by the public parameters) is larger than the bound ℓ.

Definition 7.1 (Bounded semantic-security)**.** *Let $\ell = \ell(n)$ be a polynomial. A graded encoding scheme $\mathcal{E}$ is bounded semantically-secure, if it is semantically-secure as in Definition 6.8, but only for (k, ℓ')-message samplers $(\mathbb{Z}, \mathbb{M})$ as in Definition 6.7, for $\ell' \leq \ell$ and for any polynomial k.*

7.1.2 Entropic Semantic-Security

In Definition 6.8, we do not put any special restrictions on the min-entropy of the message vectors $\vec{m}_0, \vec{m}_1$. Following [PTS13], in *entropic semantic-security*, we restrict attention to message vectors $\vec{m}_0, \vec{m}_1$ where every single coordinate has high min-entropy. Furthermore, we require that for any arithmetic circuit C that respects the set vector $\vec{S}$, it holds that on input distributions $\vec{m}_0$ and $\vec{m}_1$ the value of each intermediate wire in the computation $C(\vec{m}_b)$ has high min-entropy.

This relaxation is significant, since as noted in [GGH13a], their graded encoding scheme becomes vulnerable to *weak discrete log attacks*, when encodings of constants are released in any intermediate level. In such cases, the non-entropic Definition 6.8 fails to hold. (Similar attacks are not known against the scheme of [CLT13].) Entropic semantic-security circumvents such attacks.

Definition 7.2 (Entropic message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme, let $k = k(n)$ and $\ell = \ell(n)$ be polynomials and let $(\mathbb{Z}, \mathbb{M})$ be a (k, ℓ)-message sampler as in Definition 6.7. We say that $(\mathbb{Z}, \mathbb{M})$ is entropic, if the vector of sets $\vec{S} \in (2^{[k]})^\ell$ and the distributions of $\vec{m}_0, \vec{m}_1 \in \mathbb{Z}_p^\ell$ output by $\mathbb{M}(1^n, \mathbb{Z}(n), p)$ (where the prime p is included in public parameters pp that are in the support of*

$\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$*) satisfy the following. For any* $b \in \{0,1\}$*, and any* $\vec{S}$*-respecting arithmetic circuit* C*, let* w *be the random variable denoting the value of any intermediate wire in the computation* $C(\vec{m}_b)$*, corresponding to a proper subset* $S_w \subsetneq [k]$*. Then* $H_\infty(w) = \Omega(n)$*.*

Definition 7.3 (Entropic semantic-security)**.** *A graded encoding scheme* $\mathcal{E}$ *satisfies entropic semantic-security, if it semantically-secure as in Definition 6.8, but only for entropic admissible message samplers, rather than all admissible message samplers.*

7.1.3 Concentrated Semantic-Security

Pass *et al.* [PTS13] consider another weakening of semantic security, which we will refer to as *concentrated* semantic security.[9] Recall that in Definition 6.8, we require that semantic security holds for all admissible distributions $\vec{m}_0, \vec{m}_1$, where admissibility is defined in terms of indistinguishability with respect to algebraic adversaries. Concentrated semantic security further restricts the class of admissible distributions. Intuitively, rather than requiring that an algebraic adversary cannot distinguish the oracle $\mathcal{M}(p, k, \vec{S}, \vec{m}_b)$ for $b = 0$ and for $b = 1$, we now require for every S-respecting arithmetic circuit C given as a query to $\mathcal{M}$, there exists bit $c \in \{0,1\}$ such that for every $b \in \{0,1\}$, with overwhelming probability over $\vec{m}_b$, the oracle $\mathcal{M}(p, k, \vec{S}, \vec{m}_b)$ answers the query C with c. Or alternatively, $C(m_b) = 0$ if and only if $c = 1$. The name "concentrated semantic security" expresses the fact that the distribution $\mathcal{M}(p, k, \vec{S}, \vec{m}_b)$ (of functions from circuits to $\{0,1\}$) is *concentrated* (in the sense of Definition 3.1) around the same function for $b = 0$ and $b = 1$.

Definition 7.4 (Concentrated admissible message sampler)**.** *Let* $\mathcal{E}$ *be a graded encoding scheme, let* $k = k(n)$ *and* $\ell = \ell(n)$ *be polynomials and let* $(\mathbb{Z}, \mathbb{M})$ *be a* (k, ℓ)*-message sampler as in Definition 6.7. We say that* $(\mathbb{Z}, \mathbb{M})$ *is concentrated if for every polynomial* s *there exists a negligible function* μ *such that for every* $n \in \mathbb{N}$ *every public parameters* pp *in the support of* $\mathsf{InstGen}_{\mathcal{E}}(1^n, 1^k)$ *including a prime* p*, and every* $\vec{S}$*-respecting arithmetic circuit* C *(*$\vec{S}$ *is the vector of sets that* $\mathbb{M}(1^n, \mathbb{Z}(n), p)$ *outputs) of size at most* $s(n)$ *there exists a bit* $c \in \{0,1\}$ *such that for every* $b \in \{0,1\}$*:*

$$\Pr\left[\begin{array}{c}(\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p); \\ \mathcal{M}(p, k, \vec{S}, \vec{m}_b)(C) \neq c\end{array}\right] \leq \mu(n) \ .$$

Remark 7.1. *Note that a concentrated* (k, ℓ)*-message sampler is in particular a* (k, ℓ)*-message sampler. Namely, any (unbounded) algebraic adversary* $\mathcal{A}$ *who is given oracle access to* $\mathcal{M}(p, k, \vec{S}, \vec{m}_b)$ *cannot distinguish between the case where* $b = 0$ *and the case where* $b = 1$ *since with overwhelming probability all the queries* $\mathcal{A}$ *makes to* $\mathcal{M}$ *have the same output in both cases.*

Definition 7.5 (Concentrated semantic-security)**.** *A graded encoding scheme* $\mathcal{E}$ *is concentrated semantically-secure, if it semantically-secure as in Definition 6.8, but only for concentrated admissible message samplers, rather than all admissible message samplers.*

7.1.4 Single-Message Semantic Security

Recall that in the definition of semantic security (Definition 6.8), we consider two arbitrary vectors of messages $\vec{m}_0, \vec{m}_1$. Following [PTS13], one can relax the definition to *single-message semantic security*, where we consider an adversary that tries to distinguish between encodings of two single messages m_0 and m_1, given an additional vector $\vec{z}$ of polynomially many auxiliary encodings. In other words, one can relax Definition 6.8 by restricting the message samplers to output $\vec{m}_0, \vec{m}_1$ that are identical in all coordinates but the last one.

[9] The definition of semantic security in [PTS13] already includes the concentrated relaxation. [PTS13] refer to the plain (non-concentrated) notion of semantic security that we consider here as *uber security*.

Definition 7.6 (Admissible single-message sampler)**.** *Let $(\mathbb{Z}, \mathbb{M})$ be an admissible (k, ℓ)-message sampler as in Definition 6.7. We say that $(\mathbb{Z}, \mathbb{M})$ is an admissible single-message sampler, if for any two vectors $\vec{m}_0, \vec{m}_1 \in \mathbb{Z}_p^\ell$ in the support of $\mathbb{M}$, $\vec{m}_0[i] = \vec{m}_1[i]$ for all $i \in [\ell - 1]$.*

Definition 7.7 (Single-message semantic-security)**.** *A graded encoding scheme $\mathcal{E}$ is single-message semantically-secure, if it is semantically-secure as in Definition 6.8, but only for admissible single-message samplers, rather than all admissible message samplers.*

Remark 7.2 (Constant-message semantic-security)**.** *A bit more generally, [PTS13] consider constant-message semantic security, requiring that it is hard to distinguish between encodings of a constant number of messages $(m_{0,1}, \ldots, m_{0,c})$ and $(m_{1,1}, \ldots, m_{1,c})$, given an additional vector of polynomially many auxiliary encodings.*

7.1.5 Large-Queries Semantic Security

In Definition 6.8, we require that semantic security holds for all admissible distributions $\vec{m}_0, \vec{m}_1$, where admissibility is defined in terms of indistinguishability with respect to an algebraic adversary. The algebraic adversary is unbounded but can only access the distribution though a polynomial number of queries to an ideal graded encoding oracle. Every query is an arithmetic circuit of some bounded polynomial size. We relax this definition to *large-queries* semantic security where there is no size restriction on the algebraic adversary's queries.[10]

Definition 7.8 (Large-queries admissible message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme, let $k = k(n)$ and $\ell = \ell(n)$ be polynomials and let $(\mathbb{Z}, \mathbb{M})$ be a (k, ℓ)-message sampler as in Definition 6.7. We say that $\mathbb{M}$ is a large-queries admissible sampler if for every polynomial function q and for every (unbounded) oracle machine $\mathcal{A}$ (called the algebraic adversary) making at most $q(n)$ oracle queries where every query describes an arithmetic circuit of arbitrary size, there exists a negligible function μ such that for every $n \in \mathbb{N}$ and every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p:*

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \end{array} ; \mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(n, p) = b\right] \leq \frac{1}{2} + \mu(n) \ ,$$

where the probability is also over the coins of $\mathcal{A}$.

Definition 7.9 (Large-queries semantic-security)**.** *A graded encoding scheme $\mathcal{E}$ satisfies large-queries semantic-security, if it is semantically-secure as in Definition 6.8, but only for large-queries admissible message samplers, rather than all admissible message samplers.*

7.2 (Re)Obtaining Obfuscation under the different Relaxations

In this section we show that Theorem 6.2 and the rest of the results in Section 6.4, hold under any combination of the relaxations of semantic security defined in Section 7.1. For certain combinations this is done by showing an equivalence between the relaxed definition and the non-relaxed one. This allows to strengthen Proposition 6.1 to hold based on the relaxed notion of semantic security. In other combinations, we cannot recover Proposition 6.1 but we can reprove Theorem 6.2 by relying on properties of specific ideal graded encoding obfuscation constructions.

[10] The definition of semantic security in [PTS13] already includes the large-queries relaxation. [PTS13] refer to the plain (poly-size queries) notion of semantic security that we consider here as *strong uber security*.

7.2.1 Bounded, Entropic and Concentrated Semantic-Security

Since any ideal graded encoding obfuscation consists of a bounded number of encodings, Proposition 6.1 holds also with respect to bounded semantic security. For entropic and concentrated security we can no longer rely in our proof on an arbitrary construction of ideal graded encoding obfuscation, however, using the known constructions [BR13, BGK$^+$13, PTS13] it is straightforward to reprove Theorem 6.2 directly.

7.2.2 Single-Message Semantic-Security

The proof of Proposition 6.1 crucially relies on many message security. Using the construction of [PTS13] one can reprove Theorem 6.2 based on constant-message semantic security (following the same proof strategy as in [PTS13]). However, we suggest a more general solution. We show that the seemingly stronger definition of semantic security is equivalent to single-message (and thus also constant-message) semantic security. Furthermore, this equivalence holds even for the entropic and bounded variants of semantic security.

Lemma 7.1.

1. *A graded encoding scheme is semantically-secure (Definition 6.8) if and only if it is single-message semantically-secure (Definition 7.7).*
2. *A graded encoding scheme is entropic and/or bounded semantically-secure if and only if it is single-message entropic and/or bounded semantically-secure.*

The proof of Lemma 7.1 is given in Section A.

7.2.3 Single-Message Concentrated Semantic-Security

We do not know if for concentrated semantic-security the single-message variant is equivalent to the many-message one. However, we show that the variants are equivalent if we restrict ourselves to admissible message samplers that are also *composable*. Then we show that by using a variant of the existing ideal graded encoding obfuscation construction we can reprove Theorem 6.2 using only semantic security for composable concentrated samplers, and therefore single-message concentrated semantic security is sufficient.

More generally we show that the conclusion of Proposition 6.1 holds with respect to single-message concentrated semantic security. Intuitively we show that semantic security restricted to composable message samplers is "as useful" as standard semantic security. Specifically, we give a transformation from any admissible message sampler $(\mathbb{Z}, \mathbb{M})$ to a composable message sampler $(\mathbb{Z}, \mathbb{M}')$ such that any information that can be learned by an algebraic adversary from the encodings sampled by $(\mathbb{Z}, \mathbb{M})$ can also be learned from the encodings sampled by $(\mathbb{Z}, \mathbb{M}')$. Now we can transform *any* ideal graded encoding obfuscation construction into a new construction such that the message sampler defined in the proof of Proposition 6.1 is composable.

We start by defining composable message samplers. Then we prove the equivalence between single-message and and many-message concentrated semantic security for composable message samplers (Lemma 7.2). Finally we describe the transformation from admissible message samplers to composable message samplers (Lemma 7.3).

Composable message samplers. Recall that a message sampler $(\mathbb{Z}, \mathbb{M})$ is admissible if no algebraic adversary can distinguish encodings of element vectors $\vec{m}_0, \vec{m}_1$ sampled by $(\mathbb{Z}, \mathbb{M})$. A message sampler $(\mathbb{Z}, \mathbb{M})$ is composable if the algebraic adversary cannot distinguish encodings of *multiple* element vectors sampled independently from $(\mathbb{Z}, \mathbb{M})$.

Definition 7.10 (Composable message sampler)**.** *Let $\mathcal{E}$ be a graded encoding scheme and let $k = k(n)$ and $\ell = \ell(n)$ be polynomials. A composable (k, ℓ)-message sampler is defined by a PPT algorithm $\mathbb{M}$ and an unbounded auxiliary input sampler $\mathbb{Z}$. We require that there exist a polynomial q such that for every $n \in \mathbb{N}$, and every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, $|\mathbb{Z}(n)| < q(n)$, and $\mathbb{M}(1^n, \mathbb{Z}(n), p)$ outputs a vector of sets $\vec{S} \in (2^{[k]})^{\ell}$ and two vectors of elements $\vec{m}_0, \vec{m}_1 \in \mathbb{Z}_p^{\ell}$.*

We say that $(\mathbb{Z}, \mathbb{M})$ is composable if for every polynomials $s = s(n)$ and $q = q(n)$ there exists a negligible function μ such that for every $n \in \mathbb{N}$, every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, 1^k)$ including a prime p, and every arithmetic circuit C of size at most s, there exists a bit $c \in \{0, 1\}$ such that for every bit vector $\vec{b} \in \{0, 1\}^q$ the following holds: Let $z \leftarrow \mathbb{Z}(n)$ and for every $i \in [q]$ let

$$(\vec{S}, \vec{m}_{i,0}, \vec{m}_{i,1}) \leftarrow \mathbb{M}(z, p) \ ,$$

be a random and independent sample from $\mathbb{M}(z, p)$, then

$$\Pr\left[\mathcal{M}(p, k, \vec{S}^q, \vec{m}(\vec{b}))(C) \notin \{c, \perp\}\right] \leq \mu(n) \ ,$$

where the probability is over the coins of $\mathbb{Z}, \mathbb{M}$ and $\mathcal{A}$, the vector $\vec{S}^q$ is the concatenation of $\vec{S}$ with itself q times, and the vector $\vec{m}(\vec{b})$ is defined by:

$$\vec{m}(\vec{b}) = (\vec{m}_{1,b_1}, \ldots, \vec{m}_{q,b_q}) \ .$$

Remark 7.3. *Note that unlike the in Definition 6.7, we do allow a composable message sampler to choose the vector set $\vec{S}$ based on its randomness and auxiliary input. Here we follow that more general definition since the composable message sampler constructed in Lemma 7.3 will not satisfy the restricted definition. See also Remark 6.3.*

If the vector $\vec{S}$ depends on the randomness of the sampler, the fact that a circuit C is $\vec{S}$-respecting also depends on the randomness of the sampler. Therefore, in the definition above we explicitly allow $\mathcal{M}$ to output $\perp$ on the circuit C instead of requiring that C is $\vec{S}$-respecting, as in Definition 7.4.

Definition 7.11 (Semantic-security for composable message samplers)**.** *A graded encoding scheme $\mathcal{E}$ is semantically-secure for composable message samplers, if it is semantically-secure as in Definition 6.8, but only for composable message samplers rather than all admissible message samplers.*

Lemma 7.2. *A graded encoding scheme is concentrated (and entropic) semantically-secure for composable message samplers if and only if it is single-message concentrated (and entropic) semantically-secure for composable message samplers.*

The proof of Lemma 7.2 is given in Section A.

Next we describe a general transformation from any admissible message sampler to a composable one.

Lemma 7.3. *Let $\mathcal{E}$ be a graded encoding scheme. There exists an efficient randomized transformation T satisfying the following properties:*

1. *For every security parameter $n \in \mathbb{N}$, and for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, the transformation T takes as input n, p, and a pair of vectors $(\vec{S}, \vec{m})$ of length ℓ such that $\vec{S}$ is a vector of sets over some universe set $[k]$, and $\vec{m}$ is a vector of elements in $\mathbb{Z}_p$. T also takes randomness r. $\mathsf{T}(1^n, p, (\vec{S}, \vec{m}); \mathsf{r})$ outputs a pair of vectors $(\vec{S}^*, \vec{m}^*)$ of length $\ell + 1$ where $\vec{S}^*$ is a vector of sets over the universe set $[k \cdot n]$. We require that the vector of sets $\vec{S}^*$ depends only on the input $\vec{S}$ and the randomness r and not on $\vec{m}$.*

2. *For every single-message concentrated admissible (k, ℓ)-message sampler $(\mathbb{Z}, \mathbb{M})$, let $\mathbb{M}'$ be the sampler that on input $(z = 1^n, \mathbb{Z}(n), p)$:*

   (a) Samples $(\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(z, p)$.

   (b) Samples randomness r for T.

   (c) Obtains:

   $$(\vec{S}^*, \vec{m}_0^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}_0); \mathsf{r}) \quad , \quad (\vec{S}^*, \vec{m}_1^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}_1); \mathsf{r}) \ .$$

   (d) Outputs $(\vec{S}^*, \vec{m}_0^*, \vec{m}_1^*)$.

   Then $(\mathbb{Z}, \mathbb{M}')$ is a single-message composable $(k \cdot n, \ell + 1)$-message sampler.

3. *There exists an oracle machine $\mathcal{S}$ such that for every oracle machine $\mathcal{A}$, every $n \in \mathbb{N}$, every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p and every pair of vectors $(\vec{S}, \vec{m})$ we have:*

   $$\mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m})}(1^n, p, \vec{S}) = \mathcal{S}^{\mathcal{M}(p,k\cdot n,\vec{S}^*,\vec{m}^*)}(1^n, \mathcal{A}, p, \vec{S}^*) \ .$$

   where $(\vec{S}^*, \vec{m}^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}))$. Additionally, the running time of $\mathcal{S}$ is polynomially related to the running time of $\mathcal{A}$. $\mathcal{S}$ and $\mathcal{A}$ make the same number of queries, and the sizes of the queries $\mathcal{S}$ makes is polynomially related to the sizes of the queries $\mathcal{A}$ makes.

Note that in Lemma 7.3 the sampler $\mathbb{M}'$ outputs a set $\vec{S}^*$ depending on its randomness. See remark Remark 7.3. The proof of Lemma 7.3 is given in Section A.

7.2.4 Bounded Constant-Message Concentrated Semantic-Security

We note that unlike the reductions given in the proof of Lemma 7.1, the reduction in the proof of Lemma 7.2 from concentrated semantic security to single-message concentrated semantic security increases the number of encodings sampled by a polynomial factor which may depend on the adversary. In particular, we do not know if the equivalence between many-message and single-message holds also for *bounded* concentrated semantic security.

However Theorem 6.2 can be reproved using the construction of [PTS13] based on bounded constant-message concentrated semantic security (following the same proof strategy as [PTS13]).

7.2.5 Large-Queries Semantic-Security

We do not know how to prove Proposition 6.1 based on the weaker notion of large-queries semantic-security. However we can show that the construction of ideal graded encoding obfuscation from [BGK+13] instantiated with a large-queries semantically-secure graded encoding is average-case VBB (Definition 2.4) for every evasive circuit collection (Definition 3.1) in NC^1. Theorem 7.1 below, together with Theorem 5.1 and Lemma 5.2, give the conclusion that Theorem 6.2 holds even assuming large-queries semantically-secure graded encoding.

Theorem 7.1. *Assume there exists a large-queries semantically-secure graded encoding scheme. Then there exists:*

1. *Average-case VBB for every evasive circuit collection in NC^1.*
2. *IO for NC^1. Assuming also fully-homomorphic encryption with decryption in NC^1, there exists IO for all poly-size circuits.*

Corollary 7.1. *Assume there exists a semantically-secure graded encoding scheme. Then there exists SIO for any circuit class in NC*^{1}.

Proof intuition for Theorem 7.1

Intuitively, the reason we can no longer prove Proposition 6.1 is that if the encodings are only semantically-secure with large queries, an adversary may obtain some global information about the obfuscated function. For example, consider the following two families of functions: one contains the constant 1 function, and the other contains all functions that output 1 on all inputs except for one. These function families are both concentrated around the constant 1 function, however, if we instantiate the construction of [BGK$^+$13] with an encoding that is only semantically-secure for large queries, we can no longer argue that the adversary cannot distinguish an obfuscation of a random function in the first family and in the second.

Recall that in the ideal graded encoding obfuscation of [BGK$^+$13] given an obfuscation of a function f the adversary can learn $f(x)$ by querying the ideal graded encoding oracle on some set-respecting circuit C_x. When $f(x) = 1$, C_x evaluates to zero, and when $f(x) = 0$, C_x evaluates to an arbitrary non-zero element. Now the algebraic adversary can query the ideal graded encoding oracle with the large query $\sum_{x\in\{0,1\}^n} C_x$. Note that this query is indeed set respecting, it evaluates to zero on the constant 1 function, and it evaluates to non-zero if f maps one input to 0. Therefore, an algebraic adversary can use this query to distinguish in the ideal world, and hence we do not get any security guarantees in the real world.

Given the above difficulties, we only show that the obfuscation is average-case VBB secure for a random function taken from an evasive family. From the proof of security in [BGK$^+$13] we get that every arithmetic circuit C queried by the algebraic adversary can be represented as the sum of (potentially exponential number of) arithmetic expressions $\{C_x\}_{x\in S}$. We also have that for every x such that $f(x) = 0$, the value of C_x is a uniformly distributed value in a large field that is independent of the value of $C_{x'}$ for every $x' \neq x$. If the function family is evasive then for every $x \in S$, $f(x) = 0$ with overwhelming probability, and in particular we have that with overwhelming probability there exists $x \in S$ such that $f(x) = 0$. (Note that f may not zero on the entire set S that might be of exponential size.) Therefore, with overwhelming probability, every query made by the algebraic adversary will have a non-zero output. This, together with large-queries semantic security allows us to simulate the view of any adversary getting an obfuscation of a random evasive function. The second part of the Theorem 7.1, follows directly from the construction and proof in [BGK$^+$13].

7.3 Attacking Semantic Security of Efficient Graded Encodings

We describe a generic attack on the semantic security of any graded encoding scheme that has the following two efficiency properties (stated informally):

1. The length of the public parameters and any encoding is independent of the maximal level parameter k.
2. Given a element $r \in \mathbb{Z}_p$ and an encoding u of an element $r' \in \mathbb{Z}_p$, it is possible to test in NC^1 whether u is an encoding of r or not.

For simplicity of exposition, we formulate the attack against the basic semantic security formulation given in Definition 6.8. We note that the attack can be generalized to all the other relaxed notions discussed in Section 7, except for the notion of *bounded* semantic security (Definition 7.1), on which no generic attacks are known. Indeed, the described attack is the main motivation for the bounded semantic security relaxation. The attack is on the weaker notion of semantic security with efficient samplers (Definition 6.5)

We start by formally describing the efficiency properties we consider. Let $\mathcal{E}$ be a graded encoding scheme. We say that the encoding length of $\mathcal{E}$ is *level-independent* if there exists a fixed polynomial $m(\cdot)$ such that, for every security parameter n, and for every level parameter k, the size of the parameters sp, pp generated by $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ is bounded by $m(n)$. Additionally, for every element $\alpha \in \mathbb{Z}_p$ (where p is described by pp), and for every set $S \subseteq [k]$, the size of any encoding generated by $\mathsf{Encode}_{\mathcal{E}}(sp, \alpha, S)$ is bounded by $m(n)$. Next, consider a function $\mathsf{Test}_{\mathcal{E}}$ that takes as input public parameters pp (including a prime p), an element $r \in \mathbb{Z}_p$ and an encoding u in the highest level $[k]$. The function tests if u is an encoding of r under pp or not.

We show that if the encoding length of $\mathcal{E}$ is *level-independent* and there *exists* an implementation of the function $\mathsf{Test}_{\mathcal{E}}$ in NC^1 then $\mathcal{E}$ cannot be semantically-secure.

Note that the known graded encoding candidates of [GGH13a, CLT13] do not satisfy the efficiency properties above. Specifically, the encoding length is level-dependent, and we do not know of an efficient implementation of $\mathsf{Test}_{\mathcal{E}}$ (let alone, in NC^1). More specifically, given an encoding of r it is possible to test if another encoding u also encodes r using a zero-test, however, in the basic schemes of [GGH13a, CLT13] there is no efficient way to encode r given only the public parameters.

We show that the attack works even if we relax the requirement on function $\mathsf{Test}_{\mathcal{E}}$ as follows. Recall that given encodings of powers of 2 in $\mathbb{Z}_p$, it is possible to encode any element in $\mathbb{Z}_p$ without knowing the secret parameters. We will therefore allow the function $\mathsf{Test}_{\mathcal{E}}$ to receive such auxiliary encodings that will help it encode r. We will generalize this and allow $\mathsf{Test}_{\mathcal{E}}$ to take as input some arbitrary auxiliary encodings. (We do not require that the encoding scheme has any type of security given these auxiliary encodings.) Now the function $\mathsf{Test}_{\mathcal{E}}$ has an efficient implementation with respect to the known candidates, however, an implementation in NC^1 is still not known.

Let $\mathbb{B}$ be an efficient algorithm that is given a prime p, samples a vector of auxiliary elements $\vec{z} \in \mathbb{Z}_p^{\mathrm{poly}(n)}$. Consider the modified function $\mathsf{Test}'_{\mathcal{E}}$ that is defined like $\mathsf{Test}_{\mathcal{E}}$, except that it also gets as input encodings of the elements of $\vec{z}$ in the highest level $[k]$. That is, $\mathsf{Test}'_{\mathcal{E}}$ takes as input public parameters pp (including a prime p), an element $r \in \mathbb{Z}_p$, an encoding u, and a vector of encodings under pp of all the elements in $\vec{z} = \mathbb{B}(R)$.

We further relax requirement on the $\mathsf{Test}'_{\mathcal{E}}$ function and allow it to err. Specifically, we only require that for a pair of random elements $r, r' \in \mathbb{Z}_p$, and for an encodings u of r, $\mathsf{Test}'_{\mathcal{E}}$ recognizes u as an encoding of r but not as an encoding of r' with probability bounded away from $\frac{1}{2}$.

Definition 7.12 (Testable graded encoding scheme)**.** *We say that a graded encoding scheme $\mathcal{E}$ with a level-independent encoding length is testable, if there exists a PPT algorithm $\mathbb{B}$ and a family of circuits $\mathsf{Test}_{\mathcal{E}} = \{\mathsf{Test}_{\mathcal{E},n}\}_{n\in\mathbb{N}}$ in NC^1 such that for every polynomial $k = k(n)$ and for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)})$, including a prime p, the following holds:*

$$\Pr\left[\begin{array}{l} r, r' \leftarrow \mathbb{Z}_p \\ u = \mathsf{Encode}_{\mathcal{E}}(sp, r, [k(n)]) \\ \vec{z} \leftarrow \mathbb{B}(1^n, p) \\ \vec{w} = (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{z}_i, [k(n)]) : i \leq |\vec{z}|); \\ (\mathsf{Test}_{\mathcal{E},n}(pp, r, u, \vec{w}) = 1) \wedge (\mathsf{Test}_{\mathcal{E},n}(pp, r', u, \vec{w}) = 0) \end{array}\right] \geq \frac{9}{10}$$

where the probability is also over the coins of $\mathsf{Encode}_{\mathcal{E}}$.

Note that in the above definition we require that the encoding length of $\mathcal{E}$ is level-independent. Otherwise, we must allow the circuit family $\mathsf{Test}_{\mathcal{E}}$ to depend also on the polynomial k.

Theorem 7.2. *There is no testable graded encoding scheme with a level-independent encoding length that satisfies semantic-security.*

In the proof we will also use the ideal graded encoding obfuscation of [BGK+13]. We rely on an additional property of this construction, namely that it is secure even in the presence certain auxiliary

input. Specifically, we consider auxiliary input that consists of additional field elements that may depend on the obfuscated circuit, and are encoded with respect to the entire set $[k]$. The adversary can access these field elements via the ideal graded encoding oracle $\mathcal{M}$. In the following definition, the auxiliary input is generated by a sampler $\mathbb{B}$ that is given as input 1^n, a prime p and outputs a vector of elements $\vec{z} \in \mathbb{Z}_p^{\mathrm{poly}(n)}$. We stress that $\mathbb{B}$ may depend on the obfuscated circuit.

Definition 7.13 (Ideal graded encoding obfuscation with auxiliary input)**.** *A virtual black-box obfuscator $\mathcal{O}$ for a family of poly-size circuits $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ is virtual black-box with auxiliary input if for every polynomial functions q, s and for every (unbounded) oracle machine $\mathcal{A}$ (called the algebraic adversary) making at most $q(n)$ queries where every query describes an arithmetic circuit of size at most $s(n)$, there exist a PPT oracle machine $\mathcal{S}$ (called the simulator) and a negligible function μ, such that for every PPT sampler $\mathbb{B}$, every every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, and for every $C \in \mathcal{C}_n$:*

$$\left| \Pr\left[\begin{array}{l} (\vec{S}, \vec{m}) \leftarrow \mathcal{O}(1^n, p, C) \\ \vec{z} \leftarrow \mathbb{B}(1^n, p) \end{array} \quad ; \quad \mathcal{A}^{\mathcal{M}(p,k,\vec{S}|[k]^{|\vec{z}|},\vec{m}|\vec{z})}(n,p) = 1 \right] - \Pr[\vec{z} \leftarrow \mathbb{B}(1^n,p) \quad ; \quad \mathcal{S}^{\mathcal{A},C,\mathcal{M}(p,k,[k]^{|\vec{z}|},\vec{z})}(1^n,p) = 1] \right| \leq \mu(n) \ ,$$

where the probabilities are over the coins of the obfuscator $\mathcal{O}$, the adversary $\mathcal{A}$ and the simulator $\mathcal{S}$. The polynomial $k = k(n)$ is defined in the functionality requirement of $\mathcal{O}$.

Claim 7.1 (follows from [BGK+13])**.** *There exists an ideal graded encoding obfuscation with auxiliary input for NC^1.*

Proof of Theorem 7.2. Let $\mathcal{E}$ be a testable graded encoding scheme with level-independent encoding length. For a prime p, consider the family of functions $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ where $\mathcal{F}_n = \{f_{\vec{r},\vec{r}',b}\}$ and where $\vec{r}, \vec{r}' \in \mathbb{Z}_p^n$ and $b \in \{0,1\}$. The function takes as input public parameters pp (that are supposed to be in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)})$ for some polynomial k) including the prime p, a vector of n encodings $\vec{u}$ and a vector of auxiliary encodings $\vec{w}$. The function $f_{\vec{r},\vec{r}',b}(pp, \vec{u}, \vec{w})$ outputs b only if for at least $\frac{3}{4}$ of the indices $i \in [n]$, it holds that:

$$(\mathsf{Test}_{\mathcal{E},n}(pp, \vec{r}[i], \vec{u}, \vec{w}) = 1) \wedge (\mathsf{Test}_{\mathcal{E},n}(pp, \vec{r}'[i], \vec{u}, \vec{w}) = 0) \ .$$

Otherwise, $f_{\vec{r},\vec{r}',b}$ outputs $\perp$. Since $\mathcal{E}$ is testable the function family $\mathcal{F}$ is in NC^1.

Let $\mathcal{O}$ be virtual black-box obfuscator $\mathcal{O}$ in the ideal graded encoding model for the family $\mathcal{F}$, and let $k = k(n), \ell = \ell(n)$ be the polynomials specified in the definition of the functionality of $\mathcal{O}$ (Definition 6.4). Let $\mathbb{B}$ be the auxiliary element sampler required for $\mathsf{Test}_{\mathcal{E},n}$, as given by Definition 7.12. Let $\ell'(n)$ be a polynomial bound on the number of elements in the output of $\mathbb{B}$.

To demonstrate that $\mathcal{E}$ is not semantically-secure, we construct an efficient admissible message sampler $\mathbb{M}$ (Definition 6.6) and a corresponding PPT distinguisher $\mathcal{A}$. We start by describing $\mathbb{M}$ which will be a $(k, \ell + \ell' + n)$-sampler. Given 1^n and a prime p, the sampler $\mathbb{M}(1^n, p)$ samples the following elements.

- $\mathbb{M}$ samples two random vectors $\vec{r}, \vec{r}' \in \mathbb{Z}_p^n$.
- For every $b \in \{0,1\}$, $\mathbb{M}$ executes the ideal graded encoding obfuscator $\mathcal{O}(1^n, p, f_{\vec{r},\vec{r}',b})$ and obtains a vector of sets $\vec{S}$ and a vector of elements $\vec{m}_b$ of length ℓ. (Note that $\mathcal{O}$ outputs the same vector of sets $\vec{S}$ in both executions. See Remark 6.1.)
- $\mathbb{M}$ executes $\mathbb{B}(1^n, p)$ and obtains a vector of elements $\vec{z} \in \mathbb{Z}_p^{\ell'(n)}$.

$\mathbb{M}$ outputs $(\vec{S}', \vec{m}'_0, \vec{m}'_1)$ where:

$$\vec{S}' = \vec{S}|[k]^{|\vec{z}|+|\vec{r}|} \quad , \quad \vec{m}'_0 = \vec{m}_0|\vec{z}|\vec{r} \quad , \quad \vec{m}'_1 = \vec{m}_1|\vec{z}|\vec{r} \ .$$

Claim 7.2. $\mathbb{M}$ *is admissible.*

Proof. Let q be a polynomial and let $\mathcal{A}$ be an unbounded algebraic adversary making at most $q(n)$ oracle queries. We need to show that there exists a negligible function μ such that for all $n \in \mathbb{N}$:

$$\Pr\left[\begin{array}{l} b \leftarrow \{0,1\} \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}', \vec{m}'_0, \vec{m}'_1) \leftarrow \mathbb{M}(1^n, p); \\ \mathcal{A}^{\mathcal{M}(p,k,\vec{S}',\vec{m}'_b)}(n,p) = b \end{array}\right] \leq \frac{1}{2} + \mu(n) \ ,$$

where p is described in the public parameters pp and the probability is also over the coins of $\mathbb{M}$ and $\mathsf{InstGen}_{\mathcal{E}}$.

Fix $b \in \{0,1\}$. From the construction of $\mathbb{M}$ we have that:

$$\Pr\left[\begin{array}{l} (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}', \vec{m}'_0, \vec{m}'_1) \leftarrow \mathbb{M}(1^n, p); \\ \mathcal{A}^{\mathcal{M}(p,k,\vec{S}',\vec{m}'_b)}(n,p) = b \end{array}\right] = \tag{2}$$
$$\Pr\left[\begin{array}{l} (\vec{r}, \vec{r}') \leftarrow \mathbb{Z}_p^n \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}, \vec{m}_b) \leftarrow \mathcal{O}(1^n, p, f_{\vec{r},\vec{r}',b}) \\ \vec{z} \leftarrow \mathbb{B}(1^n, p); \\ \mathcal{A}^{\mathcal{M}(p,k\,,\vec{S}|[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{m}_b|\vec{z}|\vec{r}')}(n,p) = b \end{array}\right] .$$

By the auxiliary input virtual black-box security of $\mathcal{O}$, there exists a PPT simulator $\mathcal{S}$ and negligible function μ_1 such that for all vectors $\vec{r}, \vec{r}' \in \mathbb{Z}_p^n$:

$$\left| \begin{array}{l} \Pr\left[\begin{array}{l} (\vec{S}, \vec{m}_b) \leftarrow \mathcal{O}(1^n, p, f_{\vec{r},\vec{r}',b}) \\ \vec{z} \leftarrow \mathbb{B}(1^n, p); \\ \mathcal{A}^{\mathcal{M}(p,k\,,\vec{S}|[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{m}_b|\vec{z}|\vec{r}')}(n,p) = 1 \end{array}\right] \\ -\Pr[S^{\mathcal{A},f_{\vec{r},\vec{r}',b},\mathcal{M}(p,k\,,[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{z}|\vec{r}')}(1^n) = 1] \end{array} \right| \leq \mu_1(n) \ ,$$

And therefore:

$$\left| \begin{array}{l} \Pr\left[\begin{array}{l} (\vec{r}, \vec{r}') \leftarrow \mathbb{Z}_p^n \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}_b, \vec{m}_b) \leftarrow \mathcal{O}(1^n, p, f_{\vec{r},\vec{r}',b}) \\ (\vec{T}, \vec{z}) \leftarrow \mathbb{B}(1^n, p, k(n)); \\ \mathcal{A}^{\mathcal{M}(p,k\,,\vec{S}|[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{m}_b|\vec{z}|\vec{r}')}(n,p) = b \end{array}\right] \\ -\Pr\left[\begin{array}{l} (\vec{r}, \vec{r}') \leftarrow \mathbb{Z}_p^n \\ \vec{z} \leftarrow \mathbb{B}(1^n, p); \\ \mathcal{S}^{(f_{\vec{r},\vec{r}',b},\mathcal{M}(p,k\,,[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{z}|\vec{r}'))}(1^n, p) = b \end{array}\right] \end{array} \right| \leq \mu_1(n) \ . \tag{3}$$

Claim 7.3. *There exists a negligible function μ_2 such that:*

$$\Pr\left[\begin{array}{l} (\vec{r}, \vec{r}') \leftarrow \mathbb{Z}_p^n \\ \vec{z} \leftarrow \mathbb{B}(1^n, p) \end{array} ; \mathcal{S}^{(f_{\vec{r},\vec{r}',b},\mathcal{M}(p,k\,,[k]^{|\vec{z}|+|\vec{r}|}\,,\vec{z}|\vec{r}'))}(1^n) = b\right] \leq \frac{1}{2} + \mu_2(n) \ .$$

Proof. We say that a query $\mathcal{S}$ makes to its oracle is useful if it is a query to $f_{\vec{r},\vec{r}',b}$ that is answered by anything other than $\perp$ or it is a query C to $\mathcal{M}(p, k, [k]^{|\vec{z}|+|\vec{r}|}, \vec{z}|\vec{r}')$ that is answered by 1 and C is an arithmetic circuit that is not identically zero in the input $\vec{r}$ when fixing its other inputs to $\vec{z}$. Let $q'(n)$

be the number of queries $\mathcal{S}$ makes. (q' is a polynomial that depends on q the bound on the number of queries the adversary $\mathcal{A}$ makes). For $i \in [q'(n)]$ let B_i be the event that the i-th query made by $\mathcal{S}$ is the first useful query. Note that conditioned on B_i, the view of $\mathcal{S}$ before the i-th query is independent of $\vec{r}, \vec{r}'$ and of b. Fix $i \in [q'(n)]$. For B_i to hold it must be that the view of $\mathcal{S}$ before the i-th query is independent of $\vec{r}, \vec{r}'$. By the definition of $f_{\vec{r},\vec{r}',b}$, and using Chernoff bound, we have that the i-th query is a useful query to $f_{\vec{r},\vec{r}',b}$ only with negligible probability.

Let C be a an arithmetic circuit that is given as a query to $\mathcal{M}$ such that C is not identically zero in the input $\vec{r}$ when fixing its other inputs to $\vec{z}$. Since C is set-respecting, and the elements of $\vec{r}$ are given in the set $[k]$. C must be of the form $C(\vec{r}, \vec{z}) = C_z(\vec{z}) + L(\vec{r})$ where C_z is a set-respecting arithmetic circuit computing only on the elements of $\vec{z}$, and L is a non-zero linear function of the elements of $\vec{r}$. For B_i to hold it must be that the view of $\mathcal{S}$ before the i-th query, and specifically the function L, is independent of $\vec{r}$. Therefore, with overwhelming probability over $\vec{r}$ we have that $L(\vec{r}) \neq -C_z(\vec{z})$ and the query C is not useful. Overall we have that the probability of B_i is negligible for any $i \in [q'(n)]$ and therefore the probability that $\mathcal{S}$ makes any useful queries is negligible. We have that with overwhelming probability the view of $\mathcal{S}$ is independent of b and the claim follows. □

Combining Equations 2,3 and Claim 7.3 we get as required:

$$\Pr\left[\begin{array}{l} b \leftarrow \{0,1\} \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}', \vec{m}'_0, \vec{m}'_1) \leftarrow \mathbb{M}(1^n, p); \\ \mathcal{A}^{\mathcal{M}(p,k,\vec{S}',\vec{m}'_b)}(n) = b \end{array}\right] \leq \frac{1}{2} + \mu_1(n) + \mu_2(n) \ .$$

□

We continue to show that there exist a PPT adversary $\mathcal{A}$ contradicting the semantic security of $\mathcal{E}$. $\mathcal{A}$ gets as input the public parameters pp and the following vectors of encodings:

$$\begin{array}{l} (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}_b[i], \vec{S}_b[i]) : i \in [\ell(n)]) \\ \vec{w} = (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{z}[i], [k(n)]) : i \in [\ell(n)]) \\ \vec{u} = (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{r}[i], [k(n)]) : i \in [n]) \end{array}$$

Using the correctness of the graded encoding scheme $\mathcal{E}$ we have that $\mathcal{A}$ can perfectly emulate the oracle $\mathcal{M}(p, k, \vec{S}, \vec{m}_b)$ by applying the functions $\mathsf{Add}_{\mathcal{E}}, \mathsf{Sub}_{\mathcal{E}}, \mathsf{Mult}_{\mathcal{E}}, \mathsf{isZero}_{\mathcal{E}}$ to the encoding it is given. Therefore $\mathcal{A}$ can emulate:

$$\mathsf{Eval}_{\mathcal{O}}^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(pp, \vec{u}, \vec{w}) \ ,$$

which, by the correctness of $\mathcal{O}$ outputs $f_{\vec{r},\vec{r}',b}(pp, \vec{u}, \vec{w})$. By the definition of the function $\mathsf{Test}_{\mathcal{E}}$ we have that for every $i \in [n]$:

$$\Pr\left[(\mathsf{Test}_{\mathcal{E}}(pp, \vec{r}[i], \vec{u}[i], \vec{w}) = 1) \wedge (\mathsf{Test}_{\mathcal{E}}(pp, \vec{r}'[i], \vec{u}[i], \vec{w}) = 0)\right] \geq \frac{9}{10}$$

And therefore, by the definition of $f_{\vec{r},\vec{r}',b}$, $f_{\vec{r},\vec{r}',b}(pp, \vec{u}, \vec{w})$ outputs b with overwhelming probability, contradicting the unbounded semantic security of $\mathcal{E}$. □

Acknowledgements

We are grateful to Rafael Pass for enlightening discussions and valuable comments.

References

[BBC+14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In *TCC*, pages 26–51, 2014.

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In *CRYPTO*, pages 520–537, 2010.

[BCC+14] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. *CoRR*, abs/1401.0348, 2014.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 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. `http://eprint.iacr.org/`.

[BR13] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. Cryptology ePrint Archive, Report 2013/563, 2013. `http://eprint.iacr.org/`.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *CRYPTO*, pages 455–469, 1997.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pages 449–460, 2008.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In *TCC*, pages 72–89, 2010.

[CV13] Ran Canetti and Vinod Vaikuntanathan. Obfuscating branching programs using black-box pseudo-free groups. Cryptology ePrint Archive, Report 2013/500, 2013. `http://eprint.iacr.org/`.

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

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562, 2005.

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

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In *TCC*, pages 194–213, 2007.

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

[PTS13] Rafael Pass, Sidharth Telang, and Karn Seth. Obfuscation from semantically-secure multi-linear encodings. Cryptology ePrint Archive, Report 2013/781, 2013. `http://eprint.iacr.org/`.

[SW13] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. *IACR Cryptology ePrint Archive*, 2013:454, 2013.

[Wee05] Hoeteck Wee. On obfuscating point functions. *IACR Cryptology ePrint Archive*, 2005:1, 2005.

A Missing Proofs from Section 7.2

A.1 Proof of Lemma 7.1

We prove Part 1 of the lemma. Then we give the proof intuition for Part 2, extending the proof for bounded and entropic semantic security.

Let $\mathcal{E}$ be a graded encoding scheme. Clearly, if it is semantically-secure then it is also semantically-secure for a single message. We thus focus on proving the other direction.

Suppose $\mathcal{E}$ is not semantically-secure. Namely, there exist polynomials $k(\cdot)$ and $\ell(\cdot)$, an admissible (k,ℓ)-message sampler $(\mathbb{Z},\mathbb{M})$, and a poly-size adversary $\mathcal{A}$, such that $\mathcal{A}$ wins in the semantic security game with $(\mathbb{Z},\mathbb{M})$ with non-negligible probability. Namely, there exists a non-negligible function δ such that

$$\Pr\left[\begin{array}{c} b \leftarrow U_1 \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p); \\ \mathcal{A}\left(pp, (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}_b[i], \vec{S}[i]))_{i\in[\ell(n)]}\right) = b \end{array}\right] \geq \frac{1}{2} + \delta(n),$$

where p is described in pp.

We next construct an admissible single-message sampler $(\mathbb{Z}', \mathbb{M}')$ and a poly-size adversary $\mathcal{A}'$ that together break single-message semantic-security. The unbounded (auxiliary input) algorithm $\mathbb{Z}'$ is identical to $\mathbb{Z}$. The admissible single-message sampler $\mathbb{M}'$ takes as input the pair $(\mathbb{Z}(1^n), p)$, and does the following:

1. Run $(\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(\mathbb{Z}(1^n), p)$.
2. Let $r_0 = 0 \in \mathbb{Z}_p$ be the zero element, and let $r_1 \leftarrow \mathbb{Z}_p \setminus \{0\}$ be a uniformly random non-zero element.
3. Choose a random bit $b \in \{0,1\}$.
4. Let $v_0 = r_b$ and let $v_1 = r_{1-b}$.
5. Output $(\vec{S}', \vec{m}'_0, \vec{m}'_1)$ where $\vec{S}' = (\vec{S}, [k])$, $\vec{m}'_0 = (\vec{m}_b, v_0)$, $\vec{m}'_1 = (\vec{m}_b, v_1)$.

The poly-size algorithm $\mathcal{A}'$, on input

$$(pp, e_1, \ldots, e_\ell, e_{\ell+1}) = \left(pp, (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}_b[i], \vec{S}[i]) : i \in [\ell(n)]), \mathsf{Encode}_{\mathcal{E}}(sk, v_{b^*}, [k]\right),$$

does the following:

1. Compute $b' \leftarrow \mathcal{A}(pp, e_1, \ldots, e_\ell)$.

2. If $\mathsf{isZero}_{\mathcal{E}}(pp, e_{\ell+1}) = b'$ then output $b^* = 1$. Otherwise, output $b^* = 0$.

Claim A.1.

$$\Pr[\mathcal{A}'(pp, e_1, \ldots, e_{\ell+1}) = b^*] \geq \frac{1}{2} + \delta.$$

Proof.

$$\Pr[\mathcal{A}'(pp, e_1, \ldots, e_{\ell+1}) = b^*] =$$
$$\frac{1}{2}(\Pr[\mathcal{A}'(pp, e_1, \ldots, e_{\ell+1}) = 0 |\ b^* = 0] + \Pr[\mathcal{A}'(pp, e_1, \ldots, e_{\ell+1}) = 1 |\ b^* = 1]) =$$
$$\frac{1}{2}(\Pr[\mathcal{A}'(pp, e_1, \ldots, e_\ell, \mathsf{Encode}_{\mathcal{E}}(sp, r_b, [k])) = 0] + \Pr[\mathcal{A}'(pp, e_1, \ldots, e_\ell, \mathsf{Encode}_{\mathcal{E}}(sp, r_{1-b}, [k])) = 1]) =$$
$$\frac{1}{2}(\Pr[A(pp, e_1, \ldots, e_\ell) \neq \mathsf{isZero}(\mathsf{Encode}_{\mathcal{E}}(sp, r_b, [k]))] + \Pr[A(pp, e_1, \ldots, e_\ell) = \mathsf{isZero}(\mathsf{Encode}_{\mathcal{E}}(sp, r_{1-b}, [k]))]) =$$
$$\frac{1}{2}(\Pr[A(pp, e_1, \ldots, e_\ell) \neq 1 - b] + \Pr[A(pp, e_1, \ldots, e_\ell) = b]) \geq$$
$$\frac{1}{2} + \delta \ .$$

□

Claim A.2. $(\mathbb{Z}, \mathbb{M}')$ *is an admissible single-message message sampler.*

Proof sketch. Let $\mathcal{A}$ be any unbounded oracle machine making at most $q(n)$ oracle queries, that has an ϵ advantage in distinguishing ideal oracles induced by $\mathbb{M}'$, in the single message setting (see Definition 7.6). In other words, by the definition of $\mathbb{M}'$:

$$\Pr\left[\begin{array}{c} b^* \leftarrow \{0,1\}, b \leftarrow \{0,1\} \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \\ r_0 = 0, r_1 \leftarrow \mathbb{Z}_p \setminus \{0\} \end{array} ; \mathcal{A}^{\mathcal{M}(p,k,(\vec{S},[k]),(\vec{m}_b, r_{b\oplus b^*}))}(n,p) = b^* \right] \geq \frac{1}{2} + \epsilon.$$

We construct B that contradicts the admissibility of $(\mathbb{Z}, \mathbb{M})$. That is, B makes at most q queries, and has an $\epsilon - O(\frac{q(n)}{p})$ advantage in distinguishing between the two ideal oracles induced by $\mathbb{M}$ in the multiple-message setting (Definition 6.6). That is:

$$\Pr\left[\begin{array}{c} b \leftarrow \{0,1\} \\ (sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p) \end{array} ; B^{\mathcal{M}(p,k,\vec{S},\vec{m}_b)}(n,p) = b \right] \geq \frac{1}{2} + \epsilon - O(\frac{q(n)}{p}) \ .$$

Towards constructing B, recall that $\mathcal{A}$ produces set-respecting arithmetic circuits C_A to be evaluated on the field elements $(\vec{m}, r)$, where $r \in \{r_0, r_1\}$ and the set corresponding to r is the entire universe $[k]$. This means that we can think of any such C_A, as $C'_A(\vec{m}) + c \cdot r$, where $c \in \mathbb{Z}$. Namely, we can think of C_A as an arithmetic circuit that is first evaluated on $\vec{m}$ and then r is either added (or subtracted) c times.

Now, B emulates $\mathcal{A}$ as follows. B first chooses a random bit $b' \leftarrow \{0,1\}$ (as a guess for $b^* \oplus b$). It then runs $\mathcal{A}$ by translating any circuit C_A produced by the emulated $\mathcal{A}$ into a circuit C_B to be evaluated on the input $\vec{m}_b$ corresponding to its own oracle. For this purpose it first decomposes C_A into $C'_A + c \cdot r$ as above. If $c = 0$ then B translates C_A to $C_B(\vec{m}) = C'_A(\vec{m})$. If $c \neq 0$ but $b' = 0$, then it also it

translates C_A to $C_B(\vec{m}) = C'_A(\vec{m})$. In case $c \neq 0$ and $b' = 1$, then B does not make any query at all, and just answers "non-zero". Eventually, when $\mathcal{A}$ outputs a bit b^*_A, B outputs $b_B = b' \oplus b^*_A$.

To deduce B's distinguishing advantage, first consider an alternative experiment, where B sets $r_0 = 0$, and samples $r_1 \leftarrow \mathbb{Z}_p \setminus \{0\}$. In addition, B has the liberty of actually "hardwiring" $r_{b'}$ in C_A to produce $C_B(\vec{m}) = C_A(\vec{m}, r_{b'})$. We claim that in this experiment B guesses b, i.e. $b_B = b$, with probability $1/2 + \epsilon$. Indeed, in this experiment the view of $\mathcal{A}$, conditioned on any (b, b^*) where $b^* = b' \oplus b$, is identical to its view in the single message game conditioned on the same (b, b^*). Moreover, choosing b, b' independently at random and setting $b^* = b \oplus b'$, induces the same (uniform) distribution on (b, b^*) as in the single message game.

To conclude the proof, we observe that the statistical distance between the view of the emulated $\mathcal{A}$ in the above alternative experiment, and its view in the actual emulation, is at most $\frac{q(n)}{p-1}$. Indeed, the only difference occurs when $C'_A(\vec{m}) = -c \cdot r_{b'}$, which occurs, in any one of the $q(n)$ queries with the probability at most $\frac{1}{p-1}$. □

Proof intuition for Part 2. The proof of Part 2 is identical to the proof of Part 1. Recall that in the proof, if the messages generated by the entropic (and bounded) message sampler are $\vec{m}_0, \vec{m}_1$ with sets $\vec{S}$ then the messages generated by the single-message sampler are $\vec{m}, \vec{m}_1$ with set $\vec{S}$, and messages r_0, r_1 with the set $[k]$. The fact that the distribution $((\vec{m}_0, r_0), (\vec{m}_1, r_1))$ is entropic follows from the fact that the distribution $\vec{m}_0, \vec{m}_1$ is entropic, and from the fact that r_0, r_1 cannot affect the entropic requirement since they are encoded with the universal set $[k]$. Moreover, the fact that the boundness property is maintained follows from the fact that the single-message sampler increases the number of messages generated exactly by one.

A.2 Proof of Lemma 7.2

Let $\mathcal{E}$ be a graded encoding scheme. Clearly, if it is concentrated semantically-secure for composable message samplers then it is also single-message concentrated semantically-secure for composable message samplers. We thus focus on proving the other direction.

Let $\mathcal{E}$ be a graded encoding scheme that is *single-message* concentrated semantically-secure for composable message samplers. Suppose for the sake of contradiction that $\mathcal{E}$ is not *many-message* concentrated semantically-secure for composable message samplers. Namely, there exists polynomials $k(\cdot)$ and $\ell(\cdot)$, a concentrated composable (k, ℓ)-message sampler $(\mathbb{Z}, \mathbb{M})$, and a poly-size adversary $\mathcal{A}$, such that $\mathcal{A}$ wins in the concentrated composable semantic security game (with $(\mathbb{Z}, \mathbb{M})$) with non-negligible probability. Namely, there exists a non-negligible function δ such that

$$\Pr\left[\begin{array}{c}(sp, pp) \leftarrow \mathsf{InstGen}_{\mathcal{E}}(1^n, 1^{k(n)}) \\ (\vec{S}_i, \vec{m}_{i,0}, \vec{m}_{i,1}) \leftarrow \mathbb{M}(1^n, \mathbb{Z}(n), p); \\ A\left(pp, (\mathsf{Encode}_{\mathcal{E}}(sp, \vec{m}_{i,b}, \vec{S}_i)_{i\in[\ell(n)]}\right) = b\end{array}\right] \geq \frac{1}{2} + \delta(n), \tag{4}$$

where p is described in pp.

We next construct a single-message concentrated composable message sampler $(\mathbb{Z}', \mathbb{M}')$ and a poly-size adversary $\mathcal{A}'$ that together break single-message concentrated composable semantic-security.

High-level idea. We would like to use the same proof as the proof of Lemma 7.1. However, in that proof the single-message sampler is not concentrated even if the underlying many-message sampler is concentrated. The reason is that the messages v_0, v_1 do not have high min-entropy (recall that $v_b = 0$). Instead, we need to choose v_0, v_1 so that each has high min-entropy.

The basic idea is to run $(\mathbb{Z}, \mathbb{M})$ many times, say n times (and not only once as in the proof of Lemma 7.1). Then choose n random bits $(b_1, \ldots, b_n) \leftarrow \{0,1\}^n$, let $v_0 = (b_1, \ldots, b_n)$,[11] let $v_1 \leftarrow \mathbb{Z}_p$

[11] Think of $(b_1, \ldots, b_n)$ as the binary representation of a field element.

be a random field element, and let the auxiliary messages be $\vec{z} = (\vec{m}_{1,b_1}, \ldots, \vec{m}_{n,b_n})$, where each $\vec{m}_{i,b_i}$ is the b_i vector outputted by the i'th run of $(\mathbb{Z}, \mathbb{M})$. Intuitively, $(\mathbb{Z}', \mathbb{M}')$ is a concentrated message sampler since $(\mathbb{Z}, \mathbb{M})$ is concentrated and composable, and since in the ideal world each b_i looks random. It is also composable since $(\mathbb{Z}, \mathbb{M})$ is composable. On the other hand, we would like to argue that there exists a poly-size adversary $\mathcal{A}'$ that uses $\mathcal{A}$ to guess each b_i, and thus can distinguish between v_0 and v_1. However, recall that $\mathcal{A}$ can guess each b_i only with probability $\frac{1}{2} + \delta$, and so $\mathcal{A}'$ will be able to guess $(b_1, \ldots, b_n)$ only with negligible probability.

To fix this, we amplify the success probability by using repetitions. Namely, for each $i \in [n]$ we run $(\mathbb{Z}, \mathbb{M})$ many times, say $\ell = \mathrm{poly}(k)$ times, and for each run we take the b_i vector output by $(\mathbb{Z}, \mathbb{M})$. Now $\mathcal{A}'$ will guess b_i in each of the ℓ runs and will take majority. We can then use the Chernoff bound, together with the union bound, to claim that $\mathcal{A}'$ will guess all the b_i's correctly with high probability, and thus $\mathcal{A}'$ can be used to break the single-message concentrated semantic security of $\mathcal{E}$.

There is one additional technicality. Even if the adversary $\mathcal{A}'$ guesses $\vec{b} = (b_1, \ldots, b_n)$ correctly, in order to distinguish between $\mathsf{Encode}_{\mathcal{E}}(sp, \vec{b})$ and encoding of a random field element, the adversary will need to generate an encoding of $\vec{b}$ on his own, subtract it from the given encoding, and run the zero-test. However, for the adversary to generate an encoding of $\vec{b}$ we need to assume public encoding. To avoid this assumption, in the actual protocol we take $v_0 = \langle \vec{r}, \vec{b} \rangle$, where $\vec{r} \leftarrow \mathbb{Z}_p^n$ is a vector of n random field elements, and where $\langle \cdot, \cdot \rangle$ is the inner product operator. We add $\vec{r}$ to the vector of auxiliary messages. We then claim that since in the ideal world each b_i looks random, and since the inner product is a two-source extractor, then a handle to v_0 is indistinguishable from a handle to a random element, even given handles to $\vec{r}$. However, in the real world, $\mathcal{A}'$ will use $\mathcal{A}$ to predict each b_i with very high probability, and thus will be able to use the encodings of $\vec{r}$ to compute an encoding of $\langle \vec{r}, \vec{b} \rangle$, and will compare it to v_{b^*} (using the zero-test) in order to predict b^*.

We now proceed with the formal proof. The single-message concentrated message sampler $(\mathbb{Z}', \mathbb{M}')$ is defined as follows: $\mathbb{Z}'$ is identical to $\mathbb{Z}$ and $\mathbb{M}'$ takes as input a pair $(1^n, z, p)$, and does the following:

1. Choose randomly $\vec{b} = (b_1, \ldots, b_n) \leftarrow \{0,1\}^n$,
2. Let $\ell = \ell(n) = \frac{n}{\delta^2}$.
3. For every $i \in [n]$ and for every $j \in [\ell]$, run $(\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(\mathbb{Z}(1^n), z, p)$, and denote $(\vec{S}_{i,j}, \vec{m}_{i,j}) = (\vec{S}, \vec{m}_{b_i})$. Namely, for every $i \in [n]$ run $\mathbb{M}(1^n, z, p)$ sequentially ℓ times, and in each run take the message vector corresponding to bit b_i.
4. Let $\vec{m} = (\vec{m}_1, \ldots, \vec{m}_n)$ where each $\vec{m}_i = (\vec{m}_{i,1}, \ldots, \vec{m}_{i,\ell})$, and let $\vec{S} = (\vec{S}_1, \ldots, \vec{S}_n)$ where each $\vec{S}_i = (\vec{S}_{i,1}, \ldots, \vec{S}_{i,\ell})$.
5. Sample $r_1, \ldots, r_n \leftarrow \mathbb{Z}_p$ independent random field elements, and denote $\vec{r} = (r_1, \ldots, r_n)$.
6. Let $v_0 \leftarrow \mathbb{Z}_p$ be a random field element, and let $v_1 = \langle \vec{r}, \vec{b} \rangle \in \mathbb{Z}_p$, where $\langle \vec{r}, \vec{b} \rangle$ denotes the inner product.
7. Output $(\vec{S}^*, \vec{m}_0^*, \vec{m}_1^*)$, where $\vec{S}^* = (\vec{S}, [k]^{n+1})$, $\vec{m}_0^* = (\vec{m}, \vec{r}, v_0)$ and $\vec{m}_1^* = (\vec{m}, \vec{r}, v_1)$.

The poly-size algorithm $\mathcal{A}'$, on input

$$(pp, \{\vec{e}_{i,j}\}_{i \in [n], j \in [\ell]}, \{e'_i\}_{i \in [n]}, e'')$$

where $\vec{e}_{i,j} = \mathsf{Encode}_{\mathcal{E}}(sp, \vec{z}_{i,j}, \vec{S}_{i,j})$, $e'_i = \mathsf{Encode}_{\mathcal{E}}(sp, r_i, [k])$, and $e'' = \mathsf{Encode}(sp, v_b, [k])$, does the following:

1. For each $i \in [n]$ and each $j \in [\ell]$, compute $b'_{i,j} \leftarrow A(pp, \vec{e}_{i,j})$.

2. For each $i \in [n]$ let $b'_i = \mathsf{majority}\{b'_{i,1}, \ldots, b'_{i,\ell}\}$. Let $e' = \sum_{i:b'_i=1} e'_i$.

   (e' is supposedly an encoding of $\langle \vec{r}, \vec{b} \rangle$.)

3. If $\mathsf{isZero}_{\mathcal{E}}(pp, e' - e'') = 0$ then output $b^* = 1$. Otherwise, output $b^* = 0$.

Claim A.3.

$$\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = b^*] = 1 - \mathrm{negl}(n).$$

Proof. Let GOOD denote the event that $(b'_1, \ldots, b'_n) = (b_1, \ldots, b_n)$. The Chernoff bound and the union bound, together with Equation (4), imply that

$$\Pr[\mathsf{GOOD}] \geq 1 - n \cdot 2^{-\Omega(\delta^2 \cdot \ell)} = 1 - \mathrm{negl}(n).$$

$$\begin{aligned}
&\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = b^*] \geq \\
&\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = b^* |\ \mathsf{GOOD}] \cdot \Pr[\mathsf{GOOD}] \geq \\
&\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = b^* |\ \mathsf{GOOD}] - \mathrm{negl}(n) \geq \\
&\frac{1}{2}\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = 0 |\ \mathsf{GOOD} \wedge (b^* = 0)] + \\
&\frac{1}{2}\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, e'') = 1 |\ \mathsf{GOOD} \wedge (b^* = 1)] - \mathrm{negl}(n) = \\
&\frac{1}{2}\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, \mathsf{Encode}_{\mathcal{E}}(sp, v_0, [k])) = 0 |\ \mathsf{GOOD}] + \\
&\frac{1}{2}\Pr[\mathcal{A}'(pp, \{\vec{e}_{i,j}\}_{i\in[n],j\in[\ell]}, \{e'_i\}_{i\in[n]}, \mathsf{Encode}_{\mathcal{E}}(sp, \langle \vec{r}, \vec{b} \rangle, [k])) = 1 |\ \mathsf{GOOD}] - \mathrm{negl}(n) = \\
&\frac{1}{2}\Pr[\mathsf{isZero}(\mathsf{Encode}_{\mathcal{E}}(sp, v_0, [k]) - e') \neq 0 |\ \mathsf{GOOD}] + \\
&\frac{1}{2}\Pr[\mathsf{isZero}(\mathsf{Encode}_{\mathcal{E}}(sp, \langle \vec{r}, \vec{b} \rangle, [k]) - e') = 0 |\ \mathsf{GOOD}] - \mathrm{negl}(n) = \\
&\frac{1}{2} - \mathrm{negl}(n) + \frac{1}{2} - \mathrm{negl}(n) = \\
&1 - \mathrm{negl}(n) \ .
\end{aligned}$$

□

Claim A.4. *$\mathbb{M}'$ is a single-message concentrated composable message sampler.*

The claim follows immediately from the fact that $\mathbb{M}$ is a concentrated composable message sampler, and from the fact that $\langle \cdot, \cdot \rangle$ is a strong 2-source extractor.

A.3 Proof of Lemma 7.3

We start by describing the high level idea behind the proof. To prove that an admissible message sampler $(\mathbb{Z}, \mathbb{M})$ is also composable, we need to show that an algebraic adversary with access to *many* element vectors sampled independently from either $\vec{m}_0, \vec{m}_1$ cannot distinguish the two cases, where $\vec{m}_0, \vec{m}_1$ are the distributions defined by $(\mathbb{Z}, \mathbb{M})$. To do so, we hope to apply a hybrid argument: change the distribution from $\vec{m}_0$ to $\vec{m}_1$, by changing the element vectors one by one and reduce to the case where the algebraic adversary gets access to only a single element vector. Then invoke the admissibility property of the sampler.

The above strategy is problematic since it only works when the algebraic adversary accesses every element vectors independently. However, the adversary may ask to evaluate a query C that compute

on all the element vectors together. The idea is to transform the sampler $(\mathbb{Z}, \mathbb{M})$ into a new sampler $(\mathbb{Z}, \mathbb{M}')$ that samples essentially the same elements as $(\mathbb{Z}, \mathbb{M})$, however the sets that correspond to the elements are different. The modified sets are sampled based on the original sets and the randomness of the sampler $\mathbb{M}'$ and they are chosen such that any query C on the element vector respects the modified set structure if and only if it respects the original set structure. However, the sets that are sampled by two independent invocations of $\mathbb{M}'$ will be "incompatible" with each other. We will show that any query C that operates on many element vectors and respects the set structure can be decomposed into small queries on a single element vector. This will allow us to reduce the composability of the transformed sampler $(\mathbb{Z}, \mathbb{M}')$ to the admissibility of the original sample $(\mathbb{Z}, \mathbb{M})$ via a hybrid argument.

In more details, the new sampler $\mathbb{M}'$ will execute the original sampler $\mathbb{M}$ and obtain a vector of messages $\vec{m} \in \mathbb{Z}_p^\ell$ and a vector of corresponding sets $\vec{S} \in (2^{[k]})^\ell$. Next, $\mathbb{M}'$ samples a random set $M \subseteq [n]$ and computes the modified sets vector $\vec{S}'$ such that for every $i \in [\ell]$, $\vec{S}'[i] = \vec{S}[i] \times M \subseteq [k] \times [n]$ (we think of elements in $[k] \times [n]$ as representing elements in our new universe set $[k \cdot n]$). Finally $\mathbb{M}'$ samples and additional random element $\alpha \in \mathbb{Z}_p$ together with the set $[k] \times ([n] \setminus M)$. We have that as long as $M \neq \emptyset$, for any query C that respects the set vector $\vec{S}$, the query C' that computes $\alpha \cdot C(\vec{m})$ respects the set vector $\vec{S}'$, and as long as $\alpha \neq 0$ we have that $C(\vec{m}) = 0$ iff $C'(\vec{m}, \alpha) = 0$. Additionally, we show that with overwhelming probability over the coins of sampler $\mathbb{M}'$, any set-respecting query C over two independent samples $(\vec{m}, \alpha)$ and $(\vec{m}', \alpha')$ can be decomposed into two set-respecting queries D, D' such that:

$$C(\vec{m}, \alpha, \vec{m}', \alpha') = \alpha \cdot D(\vec{m}) + \alpha' \cdot D'(\vec{m}') \ .$$

Since α, α' are random, we have that with overwhelming probability C evaluates to 0 iff both D and D' evaluate to 0.

Next, we give a formal description of the transformation T:

1. Let $\mathcal{E}$ be a graded encoding scheme. For every security parameter $n \in \mathbb{N}$, and for every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, the transformation T takes as input n, p, and a pair of vectors $(\vec{S}, \vec{m})$ of length ℓ such that $\vec{S}$ is a vector of sets over some universe set $[k]$, and m is a vector of elements in $\mathbb{Z}_p$.

2. T samples a random subset $M \subseteq [n]$ (every element in [n] is in M with probability $\frac{1}{2}$) and a random non-zero element $\alpha \in \mathbb{Z}_p$.

3. T outputs the pair of vectors $(\vec{S}^*, \vec{m}^*)$ of length $\ell + 1$ where:

$$\vec{S}^* = ([k] \times ([n] \setminus M) \, , \, \vec{S}[1] \times M, \ldots, \vec{S}[\ell] \times M) \quad , \quad \vec{m}^* = (\alpha \, , \, \vec{m}[1], \ldots, \vec{m}[\ell]) \ ,$$

   where we think of elements of $[k] \times [n]$ as representing elements of the universe set $[k \cdot n]$. Note that the set vector $\vec{S}^*$ depends only on $\vec{S}$ and the random set M and not on $\vec{m}$, as required.

The proof of the lemma is split into the following two claims.

Claim A.5. *For every single-message concentrated admissible (k, ℓ)-message sampler $(\mathbb{Z}, \mathbb{M})$, let $\mathbb{M}'$ be the sampler that on input $(1^n, z = \mathbb{Z}(n), p)$:*

1. *Samples $(\vec{S}, \vec{m}_0, \vec{m}_1) \leftarrow \mathbb{M}(z, p)$.*

2. *Samples randomness r for T.*

3. *Obtains:*

$$(\vec{S}^*, \vec{m}_0^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}_0); \mathsf{r}) \quad , \quad (\vec{S}^*, \vec{m}_1^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}_1); \mathsf{r}) \ .$$

4. *Outputs $(\vec{S}^*, \vec{m}_0^*, \vec{m}_1^*)$.*

Then $(\mathbb{Z}, \mathbb{M}')$ is a single-message composable $(k \cdot n, \ell + 1)$-message sampler.

Claim A.6. *There exists an oracle machine $\mathcal{S}$ such that for every oracle machine $\mathcal{A}$, every $n \in \mathbb{N}$, every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, k)$ including a prime p, and every pair of vectors $(\vec{S}, \vec{m})$ we have:*

$$\mathcal{A}^{\mathcal{M}(p,k,\vec{S},\vec{m})}(1^n, p, \vec{S}^*) = \mathcal{S}^{\mathcal{M}(p,k\cdot n,\vec{S}^*,\vec{m}^*)}(1^n, A, p, \vec{S}^*) \ .$$

where $(\vec{S}^*, \vec{m}^*) \leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}))$. Additionally, the running time of $\mathcal{S}$ is polynomially related to the running time of $\mathcal{A}$. $\mathcal{S}$ and $\mathcal{A}$ make the same number of queries, and the sizes of the queries $\mathcal{S}$ makes is polynomially related to the sizes of the queries $\mathcal{A}$ makes.

A.3.1 Proof of Claim A.5

Let $(\mathbb{Z}, \mathbb{M})$ be a single-message concentrated admissible (k, ℓ)-message sampler and let $\mathbb{M}'$ be a sampler defined as in the claim statement. First note that since $(\mathbb{Z}, \mathbb{M})$ is a single-message sampler, and since $\mathbb{M}'$ uses the same randomness r and the same auxiliary input z in in both executions of T, it follows from the construction of T that $(\mathbb{Z}, \mathbb{M}')$ is also a single-message sampler.

For a prime p, s polynomial $q = q(n)$, security parameter n, and for every $i \in [q]$ and $b \in \{0, 1\}$, let $\vec{S}^*_i, \vec{m}^*_{i,b}$ be vectors sampled by the following process:

$$\begin{aligned} \mathsf{r}_i &\leftarrow U_{\mathrm{poly}(n)} \ , \\ (\vec{S}, \vec{m}_{i,0}, \vec{m}_{i,1}) &\leftarrow \mathbb{M}(z, p) \ , \\ (\vec{S}^*_i, \vec{m}^*_{i,b}) &\leftarrow \mathsf{T}(1^n, p, (\vec{S}, \vec{m}_{i,b}); \mathsf{r}_i) \ . \end{aligned}$$

For a bit vector $\vec{b} \in \{0, 1\}^q$, let $\vec{S}^*, \vec{m}^*(\vec{b})$ denote the vectors:

$$\vec{S}^* = (\vec{S}^*_1, \ldots, \vec{S}^*_q) \quad , \quad \vec{m}^*(\vec{b}) = (\vec{m}^*_{1,b_1}, \ldots, \vec{m}^*_{q,b_q}) \ .$$

To show that $(\mathbb{Z}, \mathbb{M}')$ is a composable message sampler we need to show that for every polynomials $s = s(n)$ and $q = q(n)$ there exists a negligible function μ such that for every $n \in \mathbb{N}$, every public parameters pp in the support of $\mathsf{InstGen}_{\mathcal{E}}(1^n, 1^k)$ including a prime p, and every arithmetic circuit C of size at most s, there exists a bit $c \in \{0, 1\}$ such that for every bit vector $\vec{b} \in \{0, 1\}^q$,

$$\Pr\left[\mathcal{M}(p, k, \vec{S}^*, \vec{m}^*(\vec{b}))(C) \notin \{c, \perp\}\right] \leq \mu(n) \ . \tag{5}$$

For $i \in [q]$, let M_i and α_i be the variables sampled by the execution of T with randomness r_i. Let G be the event that for every distinct $i, j \in [q]$:

$$M_i \setminus M_j \neq \emptyset \quad \wedge \quad M_i \cap M_j \neq \emptyset \quad \wedge \quad M_i \cup M_j \neq [n] \ .$$

Note that $\neg G$ occurs only with negligible probability.

Let C be an arithmetic circuit of size s and let R be the event that C is $\vec{S}^*$-respecting. The key step in the proof of Claim A.5 is the following decomposition claim.

Claim A.7. *There exist circuits $\{C_i\}_{i\in[q]}$ such that for every $i \in [q]$, $|C_i| < |C|^2$ and conditioned on G and R:*

1. *For every $i \in [q]$, C_i is $\vec{S}$-respecting.*
2. *There exists constants $\{\beta_i\}_{i\in[q]}$ such that for every bit vector $\vec{b} \in \{0, 1\}^q$:*

$$\Pr\left[C(\vec{m}^*(\vec{b})) = \sum_{i\in[q]} \beta_i \cdot \alpha_i \cdot C_i(\vec{m}_{i,\vec{b}_i})\right] = 1 \ .$$

The proof of the claim is given in Section A. Next we finish the proof of Claim A.5. Note that that when R occurs we have :

$$\Pr\left[\mathcal{M}(p,k,\vec{S}^*,\vec{m}^*(\vec{b}))(C)\neq\perp\right]\leq\mu(n)\ ,$$

and (5) holds. Additionally, since the event G occurs with overwhelming probability it suffices to prove (5) conditioned on $R\wedge G$. In the rest of the proof we condition on these events.

Let $\{C_i\}_{i\in[q]}$ be the circuits given in Claim A.7. Conditioned on G and R, we have that C_i is $\vec{S}$-respecting for every $i\in[q]$. By the concentrated admissibility of $(\mathbb{Z},\mathbb{M})$ for every $i\in[q]$ there exist a a negligible function μ_i and a constant $c_i\in\{0,1\}$ such that for every $b\in\{0,1\}$:

$$\Pr\left[\mathcal{M}(p,k,\vec{S},\vec{m}_{i,b})(C)\neq c_i\right]\leq\mu_i(n)\ .$$

Fix a bit vector $\vec{b}\in\{0,1\}^q$. In the rest of the proof we sperate between the case where $c_i=1$ for every $i\in[q]$ and the case where there exists $i\in[q]$ such that $c_i=0$.

The case were $c_i=1$ for every $i\in[q]$. We have that:

$$\Pr\left[\forall_{i\in[q]}\mathcal{M}(p,k,\vec{S},\vec{m}_{i,\vec{b}_i})(C)=1\right]=\Pr\left[\forall_{i\in[q]}C_i(\vec{m}_{i,\vec{b}_i})=0\right]\geq1-\mu(n)\ ,$$

where $\mu=\sum_{i\in[q]}\mu_i$ is a negligible function. By Claim A.7 we have that:

$$\begin{aligned}\Pr\left[\mathcal{M}(p,k,\vec{S}^*,\vec{m}^*(\vec{b}))(C)\neq1\right]&=\Pr\left[C(\vec{m}^*(\vec{b}))\neq0\right]\\&=\Pr\left[\sum_{i\in[q]}\beta_i\cdot\alpha_i\cdot C_i(\vec{m}_{i,\vec{b}_i})\neq0\right]\leq\mu(n)\ ,\end{aligned}$$

and (5) holds for $c=1$.

The case were $c_i=0$ for some $i\in[q]$. we have that:

$$\Pr\left[\mathcal{M}(p,k,\vec{S}_{i,\vec{b}_i},\vec{m}_{i,\vec{b}_i})(C)=0\right]=\Pr\left[C_i(\vec{m}_{i,\vec{b}_i})\neq0\right]\geq1-\mu_i(n)\ .$$

Therefore, with probability at least $1-\mu_i(n)$ we have that the expression:

$$\sum_{i\in[q]}\beta_i\cdot\alpha_i\cdot C_i(\vec{m}_{i,\vec{b}_i})\ ,$$

is a non-zero linear function in the formal variable α_i. Since α_i is a random element in $\mathbb{Z}_p$ (independent of $m_{j,\vec{b}_j}$ for $j\in[q]$ and of α_j for $j\in[q],j\neq i$) we have that:

$$\Pr\left[\sum_{i\in[q]}\beta_i\cdot\alpha_i\cdot C_i(\vec{m}_{i,\vec{b}_i})=0\right]\leq\mu_i(n)+\frac{1}{p}\ .$$

Therefore, by Claim A.7 we have that:

$$\begin{aligned}\Pr\left[\mathcal{M}(p,k,\vec{S}^*,\vec{m}^*(\vec{b}))(C)\neq0\right]&=\Pr\left[C(\vec{m}^*(\vec{b}))=0\right]\\&=\Pr\left[\sum_{i\in[q]}\beta_i\cdot\alpha_i\cdot C_i(\vec{m}_{i,\vec{b}_i})\neq0\right]\leq\mu_i(n)+\frac{1}{p}\ ,\end{aligned}$$

and (5) holds for $c=0$.

A.3.2 Proof of Claim A.7

The claim follows from the next two claims.

Claim A.8. *For every circuit C There exist circuits $\{C_j\}_{j\in[\ell]}$ such that $\ell \leq |C|$, for every $j \in [\ell]$, $|C_j| < |C|$ and conditioned on G and on C being $\vec{S}^*$-respecting:*

1. *For every $j \in [\ell]$ there exists $i_j \in [q]$ such that C_j is $\vec{S}^*_{i_j}$-respecting.*
2. *There exists a circuit L taking ℓ inputs and contains only addition and subtraction gates such that for every bit vector $\vec{b} \in \{0,1\}^q$:*

$$\Pr\left[C(\vec{m}^*(\vec{b})) = L(C_1(\vec{m}^*_{i_1,\vec{b}_{i_1}}), \ldots, C_\ell(\vec{m}^*_{i_\ell,\vec{b}_{i_\ell}}))\right] = 1 \ .$$

Claim A.9. *For every $i \in [q]$ and for every circuit C there exists a circuit C' such that $|C'| < |C|^2$ and conditioned on G and on C being $\vec{S}^*_i$-respecting we have that C' is $\vec{S}$-respecting (over the universe set $[k]$) and for every bit $b \in \{0,1\}$:*

$$C(\vec{m}^*_{i,b}) = \alpha_i \cdot C'(\vec{m}_{i,b}) \ .$$

Proof of Claim A.8. We say that D is a sub-circuit of C if D is the circuit computing one of the wires of C. We say that D is good if it only takes inputs from $\vec{m}^*_{i,b}$ for some $i \in [q]$. We say that D is maximal good if there is no other good sub-circuit D' such that D is a sub-circuit of D'. Let $\{C_j\}_{j\in[\ell]}$ be all maximal good sub-circuits of C. Clearly, $\ell \leq |C|$. Let L be the circuit C where the sub-circuit C_j is replaced with an input wire for all $j \in [\ell]$. Clearly, for every bit vector $\vec{b} \in \{0,1\}^q$:

$$\Pr\left[C(\vec{m}^*(\vec{b})) = L(C_1(\vec{m}^*_{i_1,\vec{b}_{i_1}}), \ldots, C_\ell(\vec{m}^*_{i_\ell,\vec{b}_{i_\ell}}))\right] = 1 \ .$$

It is left to show that conditioned on G and on C being $\vec{S}^*$-respecting, C_j is $\vec{S}^*_{i_j}$-respecting for every $j \in [\ell]$ and the circuit L contains only addition and subtraction gates. For the rest of the proof we assume that G holds and that C is $\vec{S}^*$-respecting. Since C is $\vec{S}^*$-respecting, there exist a function Tag as in Definition 6.2.

Claim A.10. *If D is a maximal good sub-circuits of C taking inputs from $\vec{m}^*(\vec{b})_i$, then the output wire w_D of D satisfies $\mathsf{Tag}(w_D) = [k] \times [n]$.*

Proof. In what follows we use the following simple claim (that we state without a proof):

Claim A.11. *Let C be a set-respecting circuit and let Tag be the function defined in Definition 6.2. Let D be a sub-circuit of C with inputs wires $w^1_{in}, \ldots, w^\ell_{in}$ and output wire w_{out}. We have that:*

1. $$\mathsf{Tag}(w_{out}) = \bigcup_{j\in[\ell]} \mathsf{Tag}(w^j_{in}) \ .$$
2. *For every intermediate wire w of D there exist indexes $j_1, \ldots, j_{\ell'} \in [\ell]$ such that:*

$$\mathsf{Tag}(w_{out}) \setminus \mathsf{Tag}(w) = \bigcup_{k\in[\ell']} \mathsf{Tag}(w^{j_k}_{in}) \ ,$$

and $\mathsf{Tag}(w), \mathsf{Tag}(w^{j_1}_{in}), \ldots, \mathsf{Tag}(w^{j_{\ell'}}_{in})$ are pairwise disjoint.

Next we prove that $\mathsf{Tag}(w_D) = [k] \times [n]$ by case analysis. If w_D is also the output wire of C than $\mathsf{Tag}(w_D) = [k] \times [n]$ since C is $\vec{S}^*$-respecting. Else, there exist some sub-circuit D' of C with output wire $w_{D'}$ such that w_D and $w_{D'}$ are connected to the same gate in C.

Since C is set-respecting, we have that either $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$ (when w_D and $w_{D'}$ are connected by an addition or subtraction gate) or $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$ (when w_D and $w_{D'}$ are connected by a multiplication gate). Additionally, the circuit D can either take the input $m^*(\vec{b})_i[1] = \alpha_i$ or not take it. The rest of the analysis explore these four cases.

D does not take the input α_i and $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$. If the circuit D does not take the input $m^*(\vec{b})_i[1] = \alpha_i$, then by Claim A.11 and the construction of T, there exist a set $B \subseteq [k]$ such that $B \times M_i = \mathsf{Tag}(w_D)$. Since D is maximal, D' must take an input from $\vec{m}^*(\vec{b})_j$ for $j \neq i$. Therefore, by Claim A.11 and the construction of T, either $\mathsf{Tag}(w_{D'}) = [k \times ([n] \setminus M_j)]$ or there exist a set $A \subseteq [k]$, such that $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$. If $\mathsf{Tag}(w_{D'}) = [k] \times ([n] \setminus M_j)$, since G holds $M_i \cap M_j \neq \emptyset$ and $M_i \neq [n] \setminus M_j$ contradicting the fact that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$. If $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$, since G holds $M_j \setminus M_i \neq \emptyset$ and $M_j \not\subseteq M_i$ contradicting the fact that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$.

D does not take the input α_i and $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$. If the circuit D does not take the input $m^*(\vec{b})_i[1] = \alpha_i$, then by Claim A.11 and the construction of T, there exist a set $B \subseteq [k]$ such that $B \times M_i = \mathsf{Tag}(w_D)$. Since D is maximal, D' must take an input from $\vec{m}^*(\vec{b})_j$ for $j \neq i$. Therefore, by Claim A.11 and the construction of T, either $\mathsf{Tag}(w_{D'}) = [k \times ([n] \setminus M_j)])$ or there exist a set $A \subseteq [k]$, such that $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$. If $\mathsf{Tag}(w_{D'}) = [k] \times ([n] \setminus M_j)$, since G holds $M_i \cap M_j \neq \emptyset$ contradicting the fact that $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$. If $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$, since G holds $M_j \cap M_i \neq \emptyset$ and therefore, since $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$, it must be that $A \cap B = \emptyset$. Let $x \in B \times (M_j \setminus M_i)$. Since $x \in [k] \times [n] \setminus (\mathsf{Tag}(w_D) \cup \mathsf{Tag}(w_{D'}))$ by Claim A.11 there exist an input wire w_{in} such that $x \in \mathsf{Tag}(w_{in})$ and $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_D) = \mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_{D'}) = \emptyset$. By the construction of T, $\mathsf{Tag}(w_{in}) = E \times M_\ell$ or $\mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_\ell)$ for some $E \subseteq [k]$ and $k \in [n]$. Assume $\mathsf{Tag}(w_{in}) = E \times M_\ell$. Since $x \in \mathsf{Tag}(w_{in}) = E \times M_\ell$ we have that $E \cap B \neq \emptyset$. Since G holds, $M_i \cap M_\ell \neq \emptyset$, contradicting the fact that $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_D) = \emptyset$. Therefore it must be that $\mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_\ell)$. If $\ell \neq i$, again since G holds $M_\ell \setminus M_i \neq \emptyset$ and $M_i \cap ([n] \setminus M_\ell) \neq \emptyset$ contradicting the fact that $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_D) = \emptyset$. Therefore it must be that $\ell = i$ and $\mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_i)$. However, since G holds $M_j \setminus M_i \neq \emptyset$ and $M_j \cap ([n] \setminus M_i) \neq \emptyset$ contradicting the fact that $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_{D'}) = \emptyset$.

D takes the input α_i and $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$. Since D only takes inputs from $\vec{m}^*(\vec{b})_i$ and it take as input $m^*(\vec{b})_i[1] = \alpha_i$, by the construction of T there exist a set $B \subseteq [k]$ such that $\mathsf{Tag}(w_D) = B \times M_i \cup [k] \times ([n] \setminus M_i)$. If $B = [k]$ then $\mathsf{Tag}(w_D) = [k] \times [n]$ as required. For the rest of the proof we assume that $[n] \setminus B \neq \emptyset$. Since D is maximal, D' must take an input from $\vec{m}^*(\vec{b})_j$ for $j \neq i$. Therefore, by Claim A.11 and the construction of T, either $\mathsf{Tag}(w_{D'}) = [k \times ([n] \setminus M_j)])$ or there exist an input wire w_A of D' such that $\mathsf{Tag}(w_A) = A \times M_j$ for some $A \subseteq [k]$. If $\mathsf{Tag}(w_{D'}) = [k] \times ([n] \setminus M_j)$, since G holds $M_i \setminus M_j \neq \emptyset$. We have that $[k] \setminus B \times M_i \setminus M_j \subseteq \mathsf{Tag}(w_{D'})$ while $[k] \setminus B \times M_i \setminus M_j \not\subseteq \mathsf{Tag}(w_D)$ contradicting the fact that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$. Therefore, there exist an input wire w_A of D' such that $\mathsf{Tag}(w_A) = A \times M_j$. In particular, by Claim A.11, $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$. If $A \setminus B \neq \emptyset$, since G holds $M_j \cap M_i \neq \emptyset$. We have that $A \setminus B \times M_j \cap M_i \subseteq \mathsf{Tag}(w_{D'})$ while $A \setminus B \times M_j \cap M_i \not\subseteq \mathsf{Tag}(w_D)$ contradicting the fact that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$. Therefore, $A \subseteq B$. Let $x \in A \cap B \times (M_i \setminus M_j)$. Since $x \in B \times M_i \subset \mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$, by Claim A.11 there exists an input wire w_{in} of D' such that $x \in \mathsf{Tag}(w_{in})$ and $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_A) = \emptyset$. By the construction of T, $\mathsf{Tag}(w_{in}) = E \times M_\ell$ or $\mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_\ell)$ for some $E \subseteq [k]$ and $k \in [n]$. Assume $\mathsf{Tag}(w_{in}) = E \times M_\ell$. Since $x \in \mathsf{Tag}(w_{in}) = E \times M_\ell$ we have that $E \cap A \neq \emptyset$. Since G holds, $M_j \cap M_\ell \neq \emptyset$, contradicting the fact that $\mathsf{Tag}(w_{in}) \cap \mathsf{Tag}(w_A) = \emptyset$. Therefore it must be that $\mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_\ell)$. Since $x \in \mathsf{Tag}(w_{in}) = [k] \times ([n] \setminus M_\ell)$ we have that $M_i \setminus M_\ell \neq \emptyset$. We have that $[k] \setminus B \times M_i \setminus M_j \subseteq \mathsf{Tag}(w_A) \subseteq \mathsf{Tag}(w_{D'})$ while $[k] \setminus B \times M_i \setminus M_j \not\subseteq \mathsf{Tag}(w_D)$ contradicting the fact that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_{D'})$.

D takes the input α_i and $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$. Since D only takes inputs from $\vec{m}^*(\vec{b})_i$ and it

take as input $m^*(\vec{b})_i[1] = \alpha_i$, by the construction of T there exist a set $B \subseteq [k]$ such that $\mathsf{Tag}(w_D) = B \times M_i \cup [k] \times ([n] \setminus M_i)$. If $B = [k]$ then $\mathsf{Tag}(w_D) = [k] \times [n]$ as required. For the rest of the proof we assume that $[n] \setminus B \neq \emptyset$. Since D is maximal, D' must take an input from $\vec{m}^*(\vec{b})_j$ for $j \neq i$. Therefore, by Claim A.11 and the construction of T, either $\mathsf{Tag}(w_{D'}) = [k \times ([n] \setminus M_j)])$ or there exist a set $A \subseteq [k]$, such that $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$. If $\mathsf{Tag}(w_{D'}) = [k] \times ([n] \setminus M_j)$, since G holds $M_i \cup M_j \neq [n]$, and $([n] \setminus M_i) \cap ([n] \setminus M_j) \neq \emptyset$ contradicting the fact that $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$. If $A \times M_j \subseteq \mathsf{Tag}(w_{D'})$, since G holds $M_i \setminus M_j \neq \emptyset$ and $([n] \setminus M_i) \setminus M_j \neq \emptyset$ contradicting the fact that $\mathsf{Tag}(w_D) \cap \mathsf{Tag}(w_{D'}) \neq \emptyset$.

Overall, in all cases either $\mathsf{Tag}(w_D) = [k] \times [n]$ as required or we get a contradiction. □

We can now proceed with the proof of Claim A.8. By Claim A.10 we have that for every $j \in [\ell]$, C_j is a sub-circuit of C taking inputs from $\vec{m}^*(\vec{b})_i$ for some $i \in [n]$ and the output wire w_{C_j} of C_j satisfies:

$$\mathsf{Tag}(w_{C_j}) = [k] \times [n] = \bigcup_{S \in \vec{S}^*_i} S \ .$$

Since C is $\vec{S}^*$-respecting, C_j must be $\vec{S}^*_i$-respecting.

Since every input wire w of L is actually the output wire of the sub-circuit C_j for some $j \in [\ell]$, it follows that $\mathsf{Tag}(w) = [k] \times [n]$. Since C is $\vec{S}^*$-respecting L must set respecting where all its input wires are tagged with the universe set $[k] \times [n]$. Therefore, the circuit L must contain only addition and subtraction gates. This concludes the proof of Claim A.8. □

Proof of Claim A.9. Let $b \in \{0,1\}$ be a bit and let C be a circuit that takes inputs from $\vec{m}^*_{i,b}$ for some $i \in [q]$. Since i, b are fixed for the rest of the proof we write $\vec{S}^*$ instead of $\vec{S}^*_i$, $\vec{m}$ instead of $\vec{m}_{i,b}$, and $\vec{m}^*$ instead of $\vec{m}^*_{i,b}$. Let $\vec{S}^*_{-1}$,$\vec{m}^*_{-1}$ denote the vectors:

$$\vec{S}^*_{-1} = (\vec{S}^*[2], \ldots, \vec{S}^*[\ell+1]) \quad , \quad \vec{m}^*_{-1} = (\vec{m}^*[2], \ldots, \vec{m}^*[\ell+1]) \ .$$

By the definition of the transformation T we have that $\vec{m}^*_{-1} = \vec{m}$ and $\vec{S}^*_{-1}[j] = \vec{S}[j] \times M_i$ for all $j \in [\ell]$.

We show how to transform C into a circuit C' that only takes inputs from $\vec{m}$ (it does not take the input $\vec{m}^*[1] = \alpha_i$) and satisfy the requirements of the claim.

We say that D is a sub-circuit of C if D is the circuit computing one of the wires of C. We say that a sub-circuit D of C is good if it satisfies the following properties:

1. The top gate of D has two input wires computed by sub-circuits U, V.
2. U only takes inputs from $\vec{m}^*_{-1}$.
3. V only takes the input α_i.

Let $D_1, \ldots, D_{\ell'}$ be all the good sub-circuits of C. Note that by definition there are no distinct good sub-circuits D, D' such that D is a sub-circuit of D'. Let L be the circuit C where the sub-circuit D_j is replaced with an input wire for all $j \in [\ell']$ (Note that L may also takes all the original input wires of C except for α_i namely, L may also take inputs from $\vec{m}^*_{-1}$). By the definition of L we have that:

$$C(\vec{m}^*) = L(\vec{m}^*_{-1}, (D_1(\vec{m}^*), \ldots, D'_\ell(\vec{m}^*))) \ .$$

For every $j \in [\ell']$ let U_j, V_j be the sub-circuits computing the input wires of the top gate in D_j such that U_j only takes inputs from $\vec{m}^*_{-1} = \vec{m}$ and V_j only takes the input α_i. Let D'_j be a new circuit computing $V_j(U_j(\vec{m}))$. Let the output circuit C' be the circuit:

$$C'(\vec{m}) = L(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

Note that since $\ell' < |C|$ we have that $|C'| \leq |C|^2$.

The intuition behind this transformation is that if C is $\vec{S}_i^*$-respecting, then in every good sub-circuit D_j, U_j and V_j must be connected by a multiplication gate, and V_j must consist only of addition and subtraction gates. It follows that the circuit V_j computes the function $\alpha_i \cdot \beta$ for some constant β and the circuit D_j computes the function $\alpha_i \cdot \beta \cdot U_j(\vec{m})$. Therefore, we have that:

$$\alpha_i \cdot D'_j(\vec{m}) = \alpha_i \cdot V_j(U_j(\vec{m})) = \alpha_i \cdot \beta \cdot U_j(\vec{m}) = D_j(\vec{m}^*) \ .$$

We show that the circuit L must compute a linear function of D'_j and therefore we get that $C(\vec{m}^*) = \alpha_i \cdot C'(\vec{m})$ as required.

We continue with the formal proof showing that conditioned on G and on C being $\vec{S}_i^*$-respecting we have that C' is $\vec{S}$-respecting (over the universe set $[k]$) and $C(\vec{m}^*) = \alpha_i \cdot C'(\vec{m})$. Assume that G holds and that C is $\vec{S}_i^*$-respecting.

We say that a circuit D is partially $\vec{T}$-respecting for a vector of sets $\vec{T}$ of length ℓ if there exists a set of indexes $j_1, \ldots j_{\ell'} \in [\ell]$ and function Tag_D that satisfy that conditions in Definition 6.2, except that the input wires of D are required to be tagged with the sets $\vec{T}[j_i], \ldots, \vec{T}[j_{\ell'}]$ and we do not require that the output wire w_{out} of D is tagged with the universe set. For a partially $\vec{T}$-respecting circuit we have that $\mathsf{Tag}_D(w_{out}) = \bigcup_{k \in [\ell']} \vec{T}[j_k]$ and we denote the set $\mathsf{Tag}_D(w_{out})$ by $\mathsf{Tag}_{\vec{T}}(D)$.

We will use the following simple claim about partially set-respecting circuits (that we bring here without a proof).

Claim A.12. *Let $\vec{S}$ be a vector of sets of length ℓ and let T be a set. Let C be an arithmetic circuit. We have that:*

1. *Let $\vec{S}'$ be the vector such that $\vec{S}'[i] = \vec{S}[i] \setminus T$ for all $i \in [\ell]$. If C is partially $\vec{S}$-respecting then it is also partially $\vec{S}'$-respecting and $\mathsf{Tag}_{\vec{S}'}(C) = \mathsf{Tag}_{\vec{S}}(C) \setminus T$.*

2. *Let $\vec{S}'$ be the vector such that $\vec{S}'[i] = \vec{S}[i] \times T$ for all $i \in [\ell]$. If C is partially $\vec{S}'$-respecting then it is also partially $\vec{S}$-respecting and $\mathsf{Tag}_{\vec{S}'}(C) = \mathsf{Tag}_{\vec{S}}(C) \times T$.*

Since C is set-respecting, we have that for all $j \in [\ell']$, D_j is partially $\vec{S}^*$-respecting, U_j is partially $\vec{S}^*_{-1}$-respecting, and V_j is partially $\vec{S}^*[1]$-respecting. It follows that V_j must consist of only addition and subtraction gates and therefore, the circuit $D'_j = V_j(U_j(\vec{m}))$ is also partially $\vec{S}^*_{-1}$-respecting. Since $\vec{S}^*_{-1}[j] = \vec{S}[j] \times M_i$ for all $j \in [\ell]$ it follows from Claim A.12 that D'_j is also partially $\vec{S}$-respecting. By the definition of the transformation T we have that for all $j \in [\ell']$:

$$\mathsf{Tag}_{\vec{S}^*}(D_j) \setminus \vec{S}^*[1] = \mathsf{Tag}_{\vec{S}}(D'_j) \times M_i \ . \tag{6}$$

Since C is $\vec{S}^*$-respecting, we have that the circuit L taking the inputs $(\vec{m}^*_{-1}, (D_1(\vec{m}^*), \ldots, D'_\ell(\vec{m}^*)))$ must respect the sets vector S_L:

$$\vec{S}_L = (\vec{S}^*_{-1}, (\mathsf{Tag}_{\vec{S}^*}(D_1), \ldots \mathsf{Tag}_{\vec{S}^*}(D_{\ell'})))\ .$$

In particular, it partially respects $\vec{S}_L$ and $\mathsf{Tag}_{\vec{S}_L}(L) = [k] \times [n]$.

Let $\vec{S}'_L$ be a vector such that for every $j[\ell + \ell']$ we have:

$$\vec{S}_L[j] \setminus \vec{S}^*[1] = \vec{S}'_L[j] \ . \tag{7}$$

Recall that for every $j \in [\ell]$:

$$\vec{S}^*_{-1}[j] \setminus \vec{S}^*[1] = \vec{S}^*_{-1}[j] = \vec{S}[j] \times M_i \ .$$

And together with (6) we get that:

$$\vec{S}'_L = (\vec{S}, (\mathsf{Tag}_{\vec{S}}(D'_1), \ldots \mathsf{Tag}_{\vec{S}}(D'_{\ell'}))) \ .$$

By (7) together with Claim A.12 we get that L is also partially $\vec{S}'_L$-respecting and:

$$\mathsf{Tag}_{\vec{S}'_L}(L) \times M_i = \mathsf{Tag}_{\vec{S}_L}(L) \setminus \vec{S}^*[1] = [k] \times [n] \setminus ([k] \times [n] \setminus M_i) = [k] \times M_i \ ,$$

implying that $\mathsf{Tag}_{\vec{S}}(L) = [k]$. Combining the above with (6), we get that the circuit:

$$C'(\vec{m}) = L(\vec{m}, D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m})) \ ,$$

is partially $\vec{S}$-respecting and that $\mathsf{Tag}_{\vec{S}}(C') = [k]$. Therefore, the circuit C' is $\vec{S}$-respecting (over the universe set $[k]$).

It is left to show that $C(\vec{m}^*) = \alpha_i \cdot C'(\vec{m})$. We start by showing that for every $j \in [\ell']$, $D_j(\vec{m}^*) = \alpha_i \cdot D'_j(\vec{m})$. Since the circuit V_j is partially $\vec{S}^*[1]$-respecting it must consist of only addition and subtraction gates and therefore, there exists a constant $\beta \in \mathbb{Z}_p$ such that $V_j(\alpha) = \beta \cdot \alpha$ for every $\alpha \in \mathbb{Z}_p$. Therefore we have that $D'_j(\vec{m}) = V_j(U_j(\vec{m})) = \beta \cdot U_j(\vec{m})$. Since the circuit D_j is partially $\vec{S}^*$-respecting, U_j is partially $\vec{S}^*_{-1}$-respecting, and V_j is partially $\vec{S}^*[1]$-respecting it must be that:

$$D_j(\vec{m}^*) = U_j(\vec{m}^*_{-1}) \cdot V_j(\vec{m}^*[1]) = U_j(\vec{m}) \cdot V_j(\alpha_i) = U_j(\vec{m}) \cdot \beta \cdot \alpha_i = D'_j(\vec{m}) \cdot \alpha_i \ .$$

Recall that the circuit L is $\vec{S}_L$-respecting where:

$$\vec{S}_L = (\vec{S}^*_{-1}, (\mathsf{Tag}_{\vec{S}^*}(D_1), \ldots \mathsf{Tag}_{\vec{S}^*}(D_{\ell'}))) \ .$$

Therefore, there exists a function Tag as in Definition 6.2.

Claim A.13. *for every sub-circuit D of L computing the wire w_D, if $\vec{S}^*[1] \in \mathsf{Tag}w_D$ then:*

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$

and if if $\vec{S}^*[1] \notin \mathsf{Tag}w_D$ then:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

Before proving the claim we show that Claim A.13 completes the proof of Claim A.9. Since the output wire w_L of L satisfies $\vec{S}^*[1] \in \mathsf{Tag}w_L = [k] \times [n]$ we have that:

$$\begin{aligned} C(\vec{m}^*) &= \\ L(\vec{m}, (D_1(\vec{m}^*), \ldots, D'_\ell(\vec{m}^*))) &= \\ L(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) &= \alpha_i \cdot L(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \\ &= \alpha_i \cdot C'(\vec{m}) \end{aligned}$$

Proof of Claim A.13. If D is in input wire of L, then either $D = \vec{m}[j]$ for some $j \in [\ell]$ or that $D = D_j(\vec{m}^*)$ for some $j \in [\ell']$. If $D = \vec{m}[j]$ for some $j \in [\ell]$ we have that $\vec{S}^*[1] \notin \mathsf{Tag}w_D = \vec{S}^*_{-1}[j]$, and that:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = D(\vec{m}[j]) = D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

If $D = D_j(\vec{m}^*)$ for some $j \in [\ell']$ we have that $\vec{S}^*[1] \in \mathsf{Tag}w_D = \mathsf{Tag}_{\vec{S}^*}(D_1)$, and that:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = D_j(\vec{m}^*) = \alpha_i \cdot D'_j(\vec{m}) = \alpha_i \cdot D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

In any case the claim holds where D is an input wire.

Let D be a sub-circuit of L such that the top gate in D is adding two wires w_U, w_V computed by the sub-circuits U, V respectively where U, V satisfy the claim. (The case of subtraction gate is analogues.) Since D is partially $\vec{S}_L$-respecting we have that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_U) = \mathsf{Tag}(w_V)$. If $\vec{S}^*[1] \in \mathsf{Tag}(w_D) = \mathsf{Tag}(w_U) = \mathsf{Tag}(w_V)$ we have that:

$$U(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot U(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$
$$V(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot V(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$

and since $D = U + V$:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

Similarly, if $\vec{S}^*[1] \notin \mathsf{Tag}(w_D) = \mathsf{Tag}(w_U) = \mathsf{Tag}(w_V)$ we have that:

$$U(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = U(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$
$$V(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = V(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$

and since $D = U + V$:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

Let D be a sub-circuit of L such that the top gate in D is multiplying two wires w_U, w_V computed by the sub-circuits U, V respectively where U, V satisfy the claim. Since D is partially $\vec{S}_L$-respecting we have that $\mathsf{Tag}(w_U) \cap \mathsf{Tag}(w_V) = \emptyset$ and that $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_U) \cup \mathsf{Tag}(w_V)$. If $\vec{S}^*[1] \in \mathsf{Tag}(w_D)$ since the set $\vec{S}^*[1]$ is disjoint from all sets in $\vec{S}^*_{-1}$ and since $\mathsf{Tag}(w_U) \cap \mathsf{Tag}(w_V) = \emptyset$, the set $\vec{S}^*_{-1}$ must be contained in exactly one of the sets $\mathsf{Tag}(w_U), \mathsf{Tag}(w_V)$. Assume w.l.o.g that $\vec{S}^*[1] \notin \mathsf{Tag}(w_U)$ and $\vec{S}^*[1] \in \mathsf{Tag}(w_V)$. Therefore:

$$U(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot U(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$
$$V(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = V(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$

and since $D = U \cdot V$:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = \alpha_i \cdot D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

Similarly, if $\vec{S}^*[1] \notin \mathsf{Tag}(w_D)$, then since $\mathsf{Tag}(w_D) = \mathsf{Tag}(w_U) \cup \mathsf{Tag}(w_V)$ we have that $\vec{S}^*[1] \notin \mathsf{Tag}(w_U)$ and $\vec{S}^*[1] \notin \mathsf{Tag}(w_V)$. Therefore:

$$U(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = U(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$
$$V(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = V(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ ,$$

and since $D = U \cdot V$:

$$D(\vec{m}, (\alpha_i \cdot D'_1(\vec{m}), \ldots, \alpha_i \cdot D'_{\ell'}(\vec{m}))) = D(\vec{m}, (D'_1(\vec{m}), \ldots, D'_{\ell'}(\vec{m}))) \ .$$

We have that the claim holds in all cases where D is computing an intermediate wire of L. □

□

A.3.3 Proof of Claim A.6

The simulator $\mathcal{S}$ first recovers the original vector of sets $\vec{S}$ from $\vec{S}^*$. $\mathcal{S}$ then emulates the execution of $A(1^n, p, \vec{S})$. Let M and α be the variables sampled by T and let ℓ be the length of $\vec{S}$ and $\vec{m}$. Whenever $\mathcal{A}$ make an oracle call C to $\mathcal{M}(p, k, \vec{S}, \vec{m})$, $\mathcal{S}$ translates it to a query C' to its oracle $\mathcal{M}(p, k, \vec{S}^*, \vec{m}^*)$ with the same answer. The circuit C' evaluates the circuit C on the elements $\vec{m}^*[2], \ldots, \vec{m}^*[\ell + 1]$ and multiplies the result by $m^*[1] = \alpha$. $\mathcal{S}$ forwards all answers to $\mathcal{A}$ and outputs the same as $\mathcal{A}$. It is left to show $\mathcal{S}$ perfectly simulates the oracle $\mathcal{M}(p, k, \vec{S}, \vec{m})$. That is, we show that C' is $\vec{S}^*$-respecting iff C' is $\vec{S}$-respecting and that C' outputs 0 iff C outputs 0.

If C is $\vec{S}$-respecting there exist a function Tag as in Definition 6.2. Let Tag' be a function from the wires of C' to $2^{[k]\times[n]}$ defined as follows. For every wire w in C' that corresponds to a wire in C, $\mathsf{Tag}'(w) = \mathsf{Tag}(w) \times M$. The wire $w_{in}^{\ell+1}$ for the input α satisfies $\mathsf{Tag}'(w_{in}^{\ell+1}) = [n] \setminus M$ and for the new output wire, $\mathsf{Tag}'(w_{out}) = [k] \times [n]$. It is straight forward to verify that Tag' satisfies Definition 6.2 iff Tag does. Specifically note that if $\mathsf{Tag}(w_{out}) = [k]$ for the output wire of C, then $\mathsf{Tag}'(w_{out}) = [k] \times M \cup [k] \times ([n] \setminus M) = [k] \times [n]$. Since C' evaluates the circuit C on the elements $\vec{m}^*[1], \ldots, \vec{m}^*[\ell] = \vec{m}[1], \ldots, \vec{m}[\ell]$ and $\alpha \neq 0$ we have that C outputs $\beta \neq 0$ iff C' outputs $\alpha \cdot \beta \neq 0$.