Limits on Low-Degree Pseudorandom Generators (Or: Sum-of-Squares Meets Program Obfuscation)

Boaz Barak[1]*, Zvika Brakerski[2]**,
Ilan Komargodski[3]***, and Pravesh K. Kothari[4]†

[1] Harvard University, Cambridge, MA, USA.
b@boazbarak.org
[2] Weizmann Institute of Science, Rehovot, Israel.
zvika.brakerski@weizmann.ac.il
[3] Cornell Tech, New York, NY, USA.
komargodski@cornell.edu
[4] Princeton University and IAS, Princeton, NJ, USA.
kothari@cs.princeton.edu

Abstract. An m output pseudorandom generator $\mathcal{G}\colon (\{\pm 1\}^b)^n \to \{\pm 1\}^m$ that takes input n blocks of b bits each is said to be ℓ-block local if every output is a function of at most ℓ blocks. We show that such ℓ-block local pseudorandom generators can have output length at most $\tilde{O}(2^{\ell b} n^{\lceil \ell/2 \rceil})$, by presenting a polynomial time algorithm that distinguishes inputs of the form $\mathcal{G}(x)$ from inputs where each coordinate is sampled from the uniform distribution on m bits.

As a corollary, we refute some conjectures recently made in the context of constructing provably secure indistinguishability obfuscation (iO). This includes refuting the assumptions underlying Lin and Tessaro's [47] recently proposed candidate iO from bilinear maps. Specifically, they assumed the existence of a secure pseudorandom generator $\mathcal{G}\colon \{\pm 1\}^{nb} \to \{\pm 1\}^{2^{cb} n}$ as above for large enough $c > 3$ and $\ell = 2$. (Following this work, and an independent work of Lombardi and Vaikuntanathan [49], Lin and Tessaro retracted the bilinear maps based candidate from their manuscript.)

* Supported by NSF awards CCF 1565264 and CNS 1618026, and the Simons Foundation. Work done while the author visited Weizmann Institute of Science during Spring 2017.

** Supported by the Israel Science Foundation (Grant No. 468/14) and Binational Science Foundation (Grants No. 2016726, 2014276), ERC Project 756482 REACT and European Union PROMETHEUS Project (Horizon 2020 Research and Innovation Program, Grant 780701).

*** Supported in part by a Packard Foundation Fellowship. Most work done while the author was a Ph.D. student at the Weizmann Institute of Science, supported in part by a grant from the Israel Science Foundation (no. 950/16) and by a Levzion Fellowship.

† Work done while the author visited Weizmann Institute of Science in March 2017.

Our results actually hold for the much wider class of low-degree, non-binary valued pseudorandom generators: if every output of $\mathcal{G}\colon \{\pm 1\}^n \to \mathbb{R}^m$ ($\mathbb{R}$ = reals) is a polynomial (over $\mathbb{R}$) of degree at most d with at most s monomials and $m \geq \tilde{\Omega}(sn^{\lceil d/2 \rceil})$, then there is a polynomial time algorithm for distinguishing the output $\mathcal{G}(x)$ from z where each coordinate z_i is sampled independently from the marginal distribution on $\mathcal{G}_i$. Furthermore, our results continue to hold under arbitrary *pre-processing* of the seed. This implies that any such map $\mathcal{G}$, with arbitrary seed pre-processing, cannot be a pseudorandom generator in the mild sense of fooling a product distribution on the output space. This allows us to rule out various natural modifications to the notion of generators suggested in other works that still allow obtaining indistinguishability obfuscation from bilinear maps.

Our algorithms are based on the Sum of Squares (SoS) paradigm, and in most cases can even be defined more simply using a canonical semidefinite program. We complement our algorithm by presenting a class of candidate generators with block-wise locality 3 and constant block size, that resists both Gaussian elimination and sum of squares (SOS) algorithms whenever $m = n^{1.5-\varepsilon}$. This class is extremely easy to describe: Let $\mathbb{G}$ be any simple non-abelian group with the group operation "$*$", and interpret the blocks of x as elements in $\mathbb{G}$. The description of the pseudorandom generator is a sequence of m triples of indices (i, j, k) chosen at random and each output of the generator is of the form $x_i * x_j * x_k$.

1 Introduction

Understanding how "simple" a pseudorandom generator can be has been of great interest in cryptography and computational complexity. In particular, researchers have studied the question of whether there exist pseudorandom generators with *constant input locality*, in the sense that every output bit only depends on a constant number of the input bits. Applebaum, Ishai and Kushilevitz [9] showed that, assuming the existence of one-way functions computable by log-depth circuits, there is such a generator mapping n bits to $n + n^{\varepsilon}$ bits for a small constant $\varepsilon > 0$. Goldreich [36] gave a candidate pseudorandom generator of constant locality that could potentially have even *polynomially large* stretch (e.g. map n bits to n^s bits for some $s > 1$).[5] The possibility of such "ultra simple" high-stretch pseudorandom generators has attracted significant attention recently with applications including:

- Public key cryptography from "combinatorial" assumptions [8].
- Highly efficient multiparty computation [40].
- Reducing the assumptions needed for constructing *indistinguishability obfuscators* (iO) [4,45,48,46,5,47].

[5] While Goldreich originally only conjectured that his function is a one-way function, followup work has considered the conjecture that it is a pseudorandom generator, and also linked the two questions (see e.g., [6,11]; see also Applebaum's survey [7]).

The last application is perhaps the most exciting, as it represents the most promising pathway for basing this important cryptographic primitive on more standard assumptions. Furthermore, this application provides motivation for considering qualitatively different notions of "simplicity" of a generator. For example, it is possible to relax the condition of having small input locality to that of just having small algebraic *degree* (over the rationals), as well as allow other features such as preprocessing of the input and admitting non-Boolean outputs.

At the same time, the application to obfuscation emphasizes a fine-grained understanding of the quantitative relationship between the "simplicity" of a generator (such as its locality, or algebraic degree) and its *stretch* (i.e., ratio of output and input lengths). For example, works of Lin and Ananth and Sahai [46,5] show that a generator mapping n bits to $n^{1+\varepsilon}$ bits with locality 2 implies an obfuscation candidate based on standard cryptographic assumptions – a highly desired goal, but it is known that it is impossible to achieve super-linear stretch with locality four (let alone two) generator [52].

Very recently, Lin and Tessaro [47] proposed bypassing this limitation by considering a relaxation of locality to a notion they referred to as *block locality*. They also proposed a candidate generator with the required properties. If such secure PRGs exist, this would imply obfuscators whose security is based on standard cryptographic assumptions, a highly desirable goal. Ananth et al. [3] observed that the conditions can be relaxed further to allow generators without a block structure, and even allow non-Boolean outputs, but their method requires (among other restrictions) that each output is computed by a sparse polynomial of small degree.

In this paper we give strong limitations on this approach, in particular giving negative answers to some of the questions raised in prior works. While a priori, questions of algebraic flavor, such as the difference between the power of bilinear vs trilinear maps, and those of combinatorial essence such as the difficulty of refuting random constraint satisfaction instances might seem unrelated, it turns out that techniques useful in the study of CSP refutation yield a barrier that, somewhat surprisingly, seems to exactly correspond to what is needed to bypass the "trilinear map barrier" for obfuscation constructions.

We complement our negative results with a simple construction of a candidate degree *three* pseudorandom generator which resists known attacks (Gaussian elimination and sum-of-squares algorithms) even for output length $n^{1+\Omega(1)}$.

1.1 Our Results

To state our results, let us define the notion of the *image refutation problem* for a map $\mathcal{G}$ that takes n inputs into m outputs (e.g., a purported pseudorandom generator). Looking ahead, we will allow maps to have non-Boolean outputs.[6]

[6] Allowing non-Boolean output can make a significant difference. For example, [50, Theorem 6.1] show that every degree two Boolean-valued function on $\{\pm 1\}^n$ depends on at most four variables, which in particular means that it cannot be used as the

Informally, the image refutation problem asks for a efficiently computable certificate for a random string *not* being in the image of a purported generator $\mathcal{G}$.

Definition 1.1 (Refutation problem). *Let $\mathcal{G}\colon \{\pm 1\}^n \to \mathbb{R}^m$ and Z be a distribution over $\mathbb{R}^m$. An algorithm A is said to solve the $\mathcal{G}$-*image refutation problem w.r.t Z *if on input $z \in \mathbb{R}^m$, A outputs either* "`refuted`" *or* "?" *and satisfies:*

- *If $z = \mathcal{G}(x)$ for some $x \in \{\pm 1\}^n$ then $A(z) =$ "?".*
- $\mathbb{P}_{z\sim Z}[A(z) = \text{"}\texttt{refuted}\text{"}] \geq 0.5$

Note that in particular if Z is the uniform distribution over $\{0,1\}^m$, then the existence of an efficient algorithm that solves the $\mathcal{G}$ image refutation problem with respect to Z means that $\mathcal{G}$ is not a pseudorandom generator - in fact, an image refutation algorithm, with probability at least $1/2$, shows that a random string from $\{\pm 1\}^m$ is not in the image of $\mathcal{G}$.

Remark 1.2 (Refutation vs Distinguishing). It is instructive to contrast the algorithmic tasks of image refutation with the easier task of distinguishing the output of a pseudorandom generator from a uniformly random string. In the latter case, we are typically concerned with distinguishing the output distribution of a generator $\mathcal{G}\colon \{\pm 1\}^n \to \{\pm 1\}^m$ when the input is chosen according to the uniform distribution on $\{\pm 1\}^m$. It's easy to see that a refutation algorithm immediately yields a distinguisher. In general, refutation, however can be more powerful. For example, a refutation algorithm can distinguish between the uniform distribution on $\{\pm 1\}^m$ from the output distribution of the generator even under *arbitrary* distributions on the seed. Thus, an image refutation algorithm rules out not only the natural PRG construction but also natural modifications that involve using some non-trivial pre-processing on the seed before inputting it into the generator, thus modifying the input distribution. Such modifications were in fact suggested for candidate constructions of iO from bilinear maps in the concurrent work of [49]. While a distinguisher for the original PRG may fail after this modification, a refutation algorithm continues to work. As we discuss later, this is one of the key differences in our approach from that of [49].

Our first result is a limitation on generators with "block locality" two:

Theorem 1.3 (Limitations of two block local generators). *For every n, b, let $\mathcal{G}\colon \{\pm 1\}^{nb} \to \{\pm 1\}^m$ be such that, if we partition the input into n blocks of size b, then every output of G depends only on variables inside two blocks. Then, there is an absolute constant K such that if $m > K \cdot 2^{2b} n \log^2 n$, then there is an efficient algorithm for the $\mathcal{G}$-image-refutation problem w.r.t. the uniform distribution over $\{\pm 1\}^m$.*

basis for a pseudorandom generator with super-linear output length. It also allows us to consider polynomials that only take the values in $\{\pm 1\}$ on a subset of their inputs.

Theorem 1.3 yields an attack on the aforementioned candidate pseudorandom generator proposed by Lin and Tessaro [47] towards basing indistinguishability obfuscator on bilinear maps, as well as any other candidate of block-locality 2 compatible with their construction.

A special case that has been of considerable interest in literature is one where all outputs of the PRG are computed by the same two-block-local predicate $P\colon \{\pm 1\}^b \to \{\pm 1\}^b \to \{\pm 1\}$. For this case, we give an image refutation algorithm that works whenever the stretch $m = \tilde{\Omega}(n2^b)$. [7]

Theorem 1.4 (Limitations of two block local generators with a single predicate, Theorem 5.3). *For every n, b, let $\mathcal{G}\colon \{\pm 1\}^{nb} \to \{\pm 1\}^m$ be such that, if we partition the input into n blocks of size b, then every output of $\mathcal{G}$ is the same predicate P applied to two b-bit blocks. Then, there is an absolute constant K such that if $m > K \cdot 2^b n \log^2 n$, then there is an efficient algorithm for the $\mathcal{G}$-image-refutation problem w.r.t. the uniform distribution over $\{\pm 1\}^m$.*

Yet another special case of interest is where the candidate generator obtained is chosen at random: that is, the m pairs of blocks used to compute the output are chosen at random and, further, each predicate computing an output is chosen randomly and independently conditioned on being balanced. For this case, we show (in Theorem 5.4, Section 5.3) that we can again improve our bound on the output length from $\tilde{O}(2^{2b}n)$ to $\tilde{O}(2^b n)$:

Our next result applies to any degree d map, and even allows maps with non-Boolean output. For the refutation problem to make sense, the probability distribution Z must be non-degenerate or have large entropy, as otherwise it may well be the case that $z \sim Z$ is in the image of $\mathcal{G}$ with high probability. For real-valued distributions, a reasonable notion of non-degeneracy is that the distribution does not fall inside any small interval with high probability. Specifically, if we consider *normalized* product distributions (where $\mathbb{E}\, Z_i = 0$ and $\mathbb{E}\, Z_i^2 = 1$ for every i and the Z_i are independent), then we say that Z is *c-spread* (see Definition 4.1) if it is a product distribution and $\mathbb{P}[Z_i \notin I] \geq 0.1$ for every interval $I \subseteq \mathbb{R}$ of length at most $1/c$ (where we can think of c as a large constant or even a poly-logarithmic or small polynomial factor).

If Z is supposed to be indistinguishable from $\mathcal{G}(U)$, where U is the uniform distribution over $\{\pm 1\}^n$, then these two distributions should agree on the marginals and in particular at least on their first and second moments. Hence, we can assume that the map $\mathcal{G}$ has the same normalization as Z, meaning that $\mathbb{E}\, \mathcal{G}(U)_i = 0$ and $\mathbb{E}\, \mathcal{G}(U)_i^2 = 1$.[8] Our result for general low degree generators is the following:

Theorem 1.5 (Limitations on degree d generators). *Suppose that $\mathcal{G}\colon \{\pm 1\}^n \to \mathbb{R}^m$ is such that for every $i \in [m]$ the map $x \mapsto \mathcal{G}(x)_i$ is a normalized polynomial of degree at most d with at most s monomials. Let Z be a*

[7] Unlike the other results in this paper, Theorem 1.4 builds upon the concurrent work [49]. See Section 1.3 for a detailed comparison between this work and [49].

[8] We say that $\mathcal{G}$ is *normalized* if it satisfies these conditions. Clearly, any map can be normalized by appropriate shifting and scaling.

c-spread product distribution over $\mathbb{R}^m$. Then, there is some absolute constant K such that if $m \geq Kc^2 s n^{\lceil d/2 \rceil} \log^2 n$, then there is an efficient algorithm for the $\mathcal{G}$-image-refutation problem w.r.t. Z.

We believe the dependence on the degree d can be improved in the odd case from $\lceil d/2 \rceil$ to $d/2$. Resolving this is related to some problems raised in the CSP refutation literature (e.g., see [60, Questions 5.2.3,5.2.7,5.2.8]).

While for arbitrary polynomials we do not know how to remove the restriction on sparsity (i.e., number of non-zero monomials s), we show in Section 4 that we can significantly relax it in several settings. Moreover, the applications to obfuscation require generators that are both low degree and sparse; see Section 2. Nevertheless, we view eliminating the dependence on the sparsity as the main open question left by this work. We conjecture that this can be done, at least in the pseudorandom generator setting, as paradoxically, it seems that the only case where our current algorithm fails is when the pseudorandom generator exhibits some "non-random" behavior. Improving this is related to obtaining better upper bound on the stretch of block-local generators.

Up to the dependence on sparsity, Theorem 1.5 answers negatively a question of Lombardi and Vaikuntanathan [50, Question 7.2], who asked whether it is possible to have a degree d pseudorandom generator with stretch $n^{\lceil \frac{3}{4} d \rceil + \varepsilon}$. It was already known by the work of Mossel et al. [52] that such output length cannot be achieved by *d-local* generators; our work shows that, at least for $n^{o(1)}$-sparse polynomials, relaxing locality to the notion of algebraic degree does not help achieve a better dependency .

All of our results are based on the same algorithm: the *sum of squares* (SOS) semidefinite program ([59,55,44]; see the lecture notes [16]). This is not surprising as for refuting CSPs, semidefinite programs in general and the sum-of-squares semi-definite programming hierarchy in particular are the strongest known general tools [56,43]. This suggests that for future candidate generators, it will be useful to prove resilience at least with respect to this algorithm. Fortunately, there is now a growing body of techniques to prove such lower bounds.

Here, we establish that the sum-of-squares algorithm cannot be used to give an attack on PRGs with stretch $O(n2^b)$. Note that the sum of squares algorithm captures all the techniques in literature for efficiently refuting (non-linear) random CSPs including the algorithms in this paper and the work of [49]. Our lower bound on the sum of squares algorithm below shows that using such techniques, one cannot hope to attack two-block-local PRGs with stretch at most $O(n2^b)$ - for the case of identical predicates computing all outputs of the generator, this, in particular, establishes the optimality of our analyis of any technique captured by the sum of squares framework.

Concretely, in Section 6, we show that there is a natural sum-of-squares resistant construction with a stretch of $\tilde{\Theta}(n2^b)$. We stress that this PRG is only secure against a sum-of-square algorithm, and is actually *insecure* outside the sum-of-squares framework.

Theorem 1.6 (See Theorem 6.1 for a formal version). *For any $b \geq 10 \log\log(n)$, there is a construction of a two-block-local PRG $\mathcal{G} \colon (\{\pm 1\}^b)^n \to \{\pm 1\}^m$ for $m = \Omega(n2^b)$ such that degree-$\Theta(n/2^{4b})$ sum of squares algorithm cannot solve the refutation problem for $\mathcal{G}$.*

For example, for $b < \varepsilon/4 \log(n)$, the above results rules out an attack on $\Omega(n2^b)$-stretch PRGs using SoS algorithm that runs in time $\sim 2^{n^{1-\varepsilon}}$.

While our results give strong barriers for degree *two* pseudorandom generators, they do not rule out a degree *three* pseudorandom generator with output length $n^{1+\Omega(1)}$. Indeed, we show a very simple candidate generator that might satisfy this property. This is the generator $\mathcal{G}$ mapping $\mathbb{G}^n$ to $\mathbb{G}^m$ where $\mathbb{G}$ is some finite *non-abelian* simple group (e.g., the size 60 group A_5), where for every $\ell \in [m]$, the ℓ^{th} output of $\mathcal{G}(x)$ is obtained as

$$\mathcal{G}(x)_\ell = x_i * x_j * x_k$$

for randomly chosen indices i, j, k and $*$ is the group operation. This generator has block locality three with constant size blocks and also (using the standard representation of group elements as matrices) has algebraic degree three as well. Yet, it is a hard instance for the SOS algorithm which encapsulates all the techniques used in this paper. While more study of this candidate's security is surely needed, there are results suggesting that it resists algebraic attacks such as Gaussian elimination [35]. See Section 7 for details.

1.2 Prior Work

Most prior work on limitations of "simple" pseudorandom generators focused on providing upper bounds on the output length in terms of the *locality*. Cryan and Miltersen [27] observe that there is no PRG with locality 2 and proved that there is no PRG with locality 3 achieving super linear stretch (i.e., having input length n and output length $n + \omega(n)$ bits). Mossel, Shpilka, and Trevisan [52] extended this result to locality 4 PRGs and constructed (non-cryptographic) small-biased locality 5 generators with linear stretch and exponentially-small bias. They also showed that a k local generator cannot have output length better than $O(2^k n^{\lceil k/2 \rceil})$. Applebaum, Ishai, and Kushilevitz [9] showed that, under standard cryptographic assumptions, there are locality 4 PRGs with sublinear-stretch. Applebaum and Raykov [6,11] related the pseudorandomness and one-wayness of Goldreich's proposed one-way function [36] in some regime of parameters.

We focus on (algebraic) *degree* instead of locality of the predicate that is used. There were few works in the past with this property (for example [31,10]). Apart from this, another feature that distinguishes our work from much of the prior works on pseudorandom generators is the focus on the *refutation* problem (certifying that a random string is *not* in the image of the generator) as opposed to the *decision* problem (given the output of a uniformly random seed, distinguish from a random string) or the *search* problem (given the output of a uniformly random seed, recover the seed). This is important for us since we do not want to make the typical assumption that the input (i.e., seed) to the pseudorandom generator is uniformly distributed, as to allow the possibility of preprocessing for it.

The refutation problem was extensively studied in the context of random *constraint satisfaction problems* (CSPs). The refutation problem for a k-local generator with n inputs and m outputs corresponds to refuting a CSP with n variables and m constraints. Thus, the study of limitations for local generators is tightly connected to the study of refutation algorithm for CSPs. Most well studied in this setting is the problem of refuting random CSPs - given a random CSP instance with a predicate P, certify that it is far from satisfiable with high probability. There is a large body of works on the study of refuting *random* and *semirandom* CSPs, starting with the work of Feige [28].[9]

In particular, we now know tight relations between the *arity* (or locality) of the predicates and the number of constraints required to refute random instances [1,56,43] using the sum-of-squares semidefinite programming hierarchy - the algorithm of choice for the problem.

Most relevant to the current paper are works from this literature that deal with predicates that have large arity but have small degree d (or the related notion of not supporting $(d+1)$-wise independent distribution). Allen, O'Donnell, and Witmer [1] showed that random instances of such predicates can be refuted when the number of constraints m is larger than $\tilde{O}(k^d n^{d/2})$. In his thesis proposal, Witmer [60] sketched how to generalize this to the *semirandom* setting, though only for the case of *even* degree d. This is related to the questions considered in this work for higher degree, though our model is somewhat more general, considering not just CSPs but arbitrary low-degree maps.

The notion of ℓ *block locality* is equivalent to the notion of CSPs of arity ℓ over a *large alphabet* (specifically, exponential in the block size). Though much of the CSP refutation and approximation literature deals with CSPs over a binary alphabet, there have been works dealing with larger alphabet (see e.g., [1]). The work of [15] gives an SOS based algorithm for 2-local CSPs over large alphabet (or equivalently, 2 block-local CSPs) as long as the underlying constraint graph is a sufficiently good expander. However, their algorithm (at least their analysis) has an *exponential* dependence in the running time on the alphabet size which is unsuitable for our applications.

The main technical difference between our work and prior results in the CSP literature, is that since for CSPs we often think as the arity as constant, these works often had poor dependence on this parameter, whereas we want to handle the case that it can be as large as n^{ε} or in some cases even unrestricted. Another difference is that in the cryptographic setting, we wish to allow the designer of a pseudorandom generator significant freedom, and this motivates studying more

[9] In a *random* CSP the graph of dependence between variables and constraints is random, and we also typically consider adding a random pattern of negations or shifts to either the inputs or the outputs of the predicates. In *semirandom* instances [29,30], the graph is arbitrary and only this pattern of negations or shifts is random.

challenging semirandom models than those typically used in prior works. We discuss these technical issues in more depth in Section 3.

The algorithms in almost all the refutation works in the CSP literature can be encapsulated by the *sum of squares* semidefinite programming hierarchy. Some lower bounds for this hierarchy, showing tightness of these analysis, were given in [13,54,43]. For the alphabet-size sensitive setting of block-local PRGs, we give a lower bound in Section 6.

1.3 Comparison with [49]

In a concurrent and independent work, Lombardi and Vaikuntanathan [49] also analyzed the possibility of a secure block-wise local PRG motivated by the work of Lin and Tessaro [47]. They show that there exists an efficient polynomial-time distinguisher with the following property: for any $m \geq \tilde{\Omega}(n2^b)$ and any predicate $P\colon \{\pm 1\}^b \times \{\pm 1\}^b \to \{\pm 1\}$ in two blocks of size b, there's an efficient distinguishing algorithm for the following two distributions over $\{\pm 1\}^m$: 1) the uniform distribution on $\{\pm 1\}^m$ and 2) the output distribution of Goldreich's PRG $G_H\colon (\{\pm 1\}^b)^n \to \{0,1\}^m$ instantiated with a random graph H and the single predicate P computing all m outputs when given a uniformly random nb bit string as input. [10]

We point out the major differences between our results on block-local PRGs and that of [49] here.

1. *Distinguishing vs Refutation:* As discussed in Remark 1.2, our approach yields the stronger refutation guarantees while that of [49] yields a distinguisher. This allows us to show that reinforcing the block-local (or low-degree, more generally) PRGs by allowing arbitrary input preprocessing cannot lead to a larger stretch. This is important, as preprocessing is OK to do in the context of the applications for obfuscation, and in fact this was one of the avenues suggested for bypassing these general type of negative results.
2. *Single Predicates vs Multiple Predicates:* The work of [49] only applies to the PRGs where each output is computed using the *same* predicate. Our approach shows that block-local (or low-degree) PRGs cannot achieve large enough stretch even if each output is computed using a different predicate - a priori, one could hope that using different predicates for different outputs could add significantly to the stretch of the PRG. This bottleneck is in fact inherent in the technical approach of [49]. In particular, our approach allows us to analyze the natural candidate for 2-block-local generator obtained by applying independently chosen multiple random predicates to randomly chosen pairs of input blocks and yields an $\tilde{O}(2^b n)$ upper bound on their stretch, see Section 5.3.
3. *Random Graph vs Arbitrary Graphs:* The work of [49] only handles block-local PRGs when the underlying graph G defining the generator is chosen

[10] We learned that in an updated version of [49], they use a refutation algorithm from our work to extend their distinguisher to the case when the graph H is arbitrarily chosen.

at random. This was because [49] relied on CSP refutation results that work under the assumption of the instance being random.

4. *Special Case of Single Predicate Block-Local PRGs:* For the PRGs with all outputs computed by a single predicate, [49] show a distinguisher that works whenever the stretch of the PRG is $\Omega(n2^b)$. For this case, we show that our algorithm in fact guarantees image *refutation* at the same stretch requirement. (A previous version of our work didn't include this result on PRGs with single predicate.) Our refutation algorithm (Theorem 1.4) is in fact inspired by the application of the Chor-Goldreich Lemma in the work of [49].

We note that the three first differences: image refutation as opposed to distinguishing, allowing different predicates as opposed to a single predicate, and using arbitrary graphs as opposed to random graphs, exactly correspond to the open questions raised by [49].[11] Thus, our results block all the approaches that [49] identified as potential strategies for repairing the iO candidate. This suggests that, rather than a "patchable problem", there is perhaps a fundamental barrier to this approach of obtaining iO from bilinear maps.

1.4 Paper Organization

Section 2 explains the connection between simple generators and the construction of indistinguishability obfuscator. This explanation allows us to draw the conclusion that our algorithm renders recently proposed methods ineffective for constructing obfuscation from standard cryptographic assumptions. For those interested in additional details, the full version [12, Appendix B] contains more information about constructing obfuscators and in particular on the result of [47]. In Section 3, we provide a high level overview of our algorithmic techniques. Section 4 contains our main algorithm and analysis, and in particular proves Theorem 1.5. We use standard tools from the SDP/SOS literature that can be found in Appendix A. In Section 5 we focus our attention on pseudorandom generators with small block-locality and show tighter results than those achieved by our general analysis, in particular we prove Theorem 1.3 as well as an even tighter result for generators with single predicates (Theorem 5.3) and random two-block-local PRGs (Theorem 5.4). In Section 6, we show that sum-of-squares algorithm cannot be used to prove sharper upper bounds on the stretch than $\sim n2^b$. Finally, in Section 7 we present our class of candidate block-local generators.

2 Relating Simple Generators and Program Obfuscators

A program obfuscator [38,14] is a compiler that given a program (say represented as a Boolean circuit) transforms it into another "scrambled" program which is functionally equivalent but its implementation details are "hidden", making

[11] See Section 5 on page 12 of https://eprint.iacr.org/2017/301/20170409:183008.

it hard to reverse-engineer. The study of *indistinguishability obfuscation* (iO) stands at the forefront of cryptographic research in recent years due to two main developments. Firstly, Garg et al. [33] suggested that this notion might be achievable given sufficiently strong *cryptographic multilinear maps*, for which a candidate construction was given by [32]. Secondly, it was shown by Sahai and Waters [58] and numerous follow-up works that iO is extremely useful for constructing a wide variety of cryptographic objects, many of which are unknown to exist under any other assumption.

A fundamental question in the construction of iO from multilinear maps is the *level of multilinearity.* Without going into details, this essentially corresponds to the highest degree of polynomials that can be evaluated by this object. Whereas multilinear maps of level 2, a.k.a *bilinear maps*, can be constructed based on pairing on elliptic curves [41,17] and have been used in cryptographic literature for over 15 years, the first obfuscation candidates required *polynomial* level (in the "security parameter" of the scheme). Proposed constructions of multilinear maps for level > 2 have only started to emerge recently [32,25,26,34] and their security is highly questionable. Indeed, many concrete security assumptions were shown to be broken w.r.t all known candidates with level > 2 [18,24,21,39,20,22,51].

A beautiful work of Lin [45], followed by [48,46,5], showed that the required level of multilinearity can be reduced to a constant (ultimately 5 in [46,5]). These works show a relation between the required multilinearity level and the existence of "simple" pseudorandom generators (PRGs). At a rudimentary level, the PRGs are used to "bootstrap" simple obfuscation-like objects into full-fledged obfuscators. This approach requires PRGs mapping $\{0,1\}^n$ to $\{0,1\}^m$ with $m = n^{1+\Omega(1)}$, which can be represented as low-degree polynomials over $\mathbb{R}$.

More accurately, for a security parameter λ and large enough n, the required output length is $m = n^{1+\varepsilon} \cdot \text{poly}(\lambda)$, for some fixed polynomial $\text{poly}(\cdot)$ which is related to the computational complexity of evaluating the underlying cryptographic primitives. One can ensure this condition as long as the output length is at least $n^{1+\Omega(1)}$ by setting n to be a sufficiently large polynomial in λ. The situation complicates further when trying to optimize the concrete constant corresponding to the level of multilinearity by means of preprocessing as in [46,5,47]. The stretch bound needs to hold even with respect to the preprocessed seed length (see the full version [12, Appendix B] for more details).

Lin [46] and Ananth and Sahai [5] instantiated this approach with locality-5 PRGs, which can trivially be represented as degree 5 polynomials. Their main insight was that for constant locality PRGs, preprocessing only blows up the seed by a constant factor. However, even so, the required stretch is impossible to achieve with locality smaller than 5 [52].

Implications of our Work to Candidate Bilinear-Maps-Based Constructions. Very recently, Lin and Tessaro [47] proposed an approach to overcome the locality barrier and possibly get all the way to an instantiation of iO based on bilinear maps. This could be a major breakthrough in cryptographic research, allowing to base "fantasy" cryptography on well studied hardness assumptions. Lin and Tessaro showed that it is sufficient if the PRG has low *block-wise locality* for blocks of logarithmic size. Namely, if we consider the seed of the PRG as an $b \times n$ matrix for $b = O(\log n)$, then each output bit can be allowed to depend on ℓ columns of this matrix. The required output length is $m = 2^{c \cdot b} n^{1+\Omega(1)}$ for some constant c. An explicit value for c is not given, but the construction requires $c > 3$ which seems to be essential for this approach (see the full version [12, Appendix B]).Block-wise locality allows a possible way to bypass the impossibility results for standard (i.e., bitwise) locality, and indeed Lin and Tessaro conjectured that there is a pseudorandom generator with output length $n^{1+\Omega(1)}$ and block-wise locality $\ell = 2$, and proposed a candidate construction.

Theorem 1.3 shows that generators with block-wise locality 2 cannot have the stretch required by the [47] construction, thus suggesting that their current techniques are insufficient for achieving obfuscation from bilinear maps. While our worst-case result leaves a narrow margin for possible improvement of the obfuscation reduction to work with $1 < c < 2$, our improved analysis for random graphs and predicates (see Theorem 5.4 in Section 5.3) suggests that our methods may be effective, at least heuristically, for generators with *any* $c > 1$.

Ananth et al. [3] observed that there is a way to generalize the [47] approach, so that it is sufficient that the range of the PRG is not $\{0, 1\}$, but rather some small specified set, so long as the degree (as a polynomial over the rationals) is bounded by the level of multilinearity. Furthermore, pseudorandomness was no longer a requirement, but rather it is only required that the output of the generator is indistinguishable from some product distribution (in particular, the one where each output entry is distributed according to its marginal). This suggests that perhaps a broader class of generators than ones that have been considered in the literature so far are useful for reducing the degree of multilinearity. However, their approach imposes a number of restrictions on such generators in order to be effective. In particular, it requires preprocessing which increases the seed length by a factor of s^c, for some $c > 1$, where s is the number of monomials in each output coordinate of the generator. Therefore, Theorem 1.5 rules out the applicability of this technique for degree 2 generators, as well.

Supporting Evidence for Block-Wise Locality 3. We show that while the Lin-Tessaro approach might not yet bring us all the way to level 2, it is quite plausible that it implies a construction from tri-linear maps. Namely, that any improvement on the state of the art would imply full-fledged program obfuscators. Specifically, as explained in Section 1.1, we present a candidate generator of block-wise locality 3, with *constant* size blocks. We show that this candidate is robust against algorithms such as ours, as well as other algorithmic methods. See Section 7 for more details.

3 Our Techniques

In this section we give an informal overview of the proof of our main result, Theorem 1.5 (i.e., limitations of low degree generators), focusing mostly on the degree two case, and making some simplifying assumptions. For the full proof see Section 4. We also describe at a high level, the ideas involved in the improved algorithm for the special cases of single-predicate generators (Theorem 1.4), random block-local generators (Theorem 5.4) and sum-of-squares lower bound (Theorem 1.6) that shows a generator with stretch $m = \Omega(n2^b)$ that is resistant to sum-of-squares based attacks (an algorithm that encapsulates all our techniques.)

As we observe in Section 3.1 below, Theorem 1.5 can be used in a black-box way to obtain a slightly weaker variant of Theorem 1.3, showing limitations of two block-local (and more generally ℓ block-local) generators. The full proof of Theorem 1.3, with the stated parameters, appears in Section 5.

Our work builds on some of the prior tools used for analyzing local pseudorandom generators and refuting constraint satisfaction problems, and in particular relies on *semidefinite programming*. The key technical difference is that while prior work mostly focused on generators/predicates with *constant* input locality or arity, we consider functions that could have much larger input locality, but have small degree. The fact that (due to our motivations in the context of obfuscation) we consider mappings with *non-Boolean* output also induces an extra layer of complexity.

We now describe our results in more detail. For simplicity, we focus on the degree two case, which is the case that is of greatest interest in the application for obfuscation. Recall that a *degree-two map of* $\mathbb{R}^n$ *to* $\mathbb{R}^m$ is a tuple of m degree two polynomials $\bar{p} = (p_1, \ldots, p_m)$. We will assume that the polynomials are *normalized* in the sense that $\mathbb{E}\, p_i(U) = 0$ and $\mathbb{E}\, p_i(U)^2 = 1$ for every i. Let Z be some "nice" (e.g., $O(1)$-spread) distribution over $\mathbb{R}^m$. (For starters, one can think of the case that Z is the uniform distribution over $\{\pm 1\}^n$, though we will want to consider more general cases as well.) The *image refutation problem* for the map $\bar{p}$ and the distribution Z is the task of certifying, given a random element z from Z, that $z \notin \bar{p}(\{\pm 1\}^n)$.

A natural approach is to use an approximation or refutation algorithm for the constraint satisfaction problem obtained from the constraints $\{p_i(x) = z_i\}$ for every i. The problem in our case is that while each of these predicates is "simple" in the sense of having quadratic degree, it can have very large locality or arity. In particular, the locality can be as large as s— the number of monomials of p_i— which we typically think of as equal to n^ε for some small $\varepsilon > 0$.

Much of the CSP refutation literature (e.g., see [1]) followed the so called "XOR principle" which reduces the task of refuting a CSP with arbitrary predicates, to the task of refuting a CSP where all constraints involve XORs (or products, when the input is thought of as ±1 valued) of the input variables. Generally, applying this principle to arity s predicates leads to a 2^s multiplicative loss in the number of constraints, and also yields XORs that can involve up to s variables, which is unacceptable in our setting. However, as shown by [1], the situation is much better when the original predicate has small degree d (which, in particular, means it does not support a $(d+1)$-wise-independent distribution). In this case, utilizing the XOR principle results in a d-XOR instance, and only yields roughly an s^d loss in the number of constraints.

However, there are two issues with this approach. First, this reduction is not directly applicable in the non-Boolean setting, which is relevant to potential applications in obfuscation. Second, reducing to an XOR inherently leads to a loss in the output length that is related to the sparsity s, while, as we'll see, it may be sometimes possible to avoid losing such factors altogether.

Thus, our algorithm takes a somewhat different approach. Given the variables $z_1, \ldots, z_m$, we consider the quadratic program

$$\max_{x \in \{\pm 1\}^n} \sum_{i=1}^{m} z_i p_i(x) \ . \tag{3.1}$$

The value of this program can be approximated to within a $O(\log n)$ factor using semidefinite relaxation via the *symmetric Grothendieck inequality* of Charikar and Wirth [19]. Thus, it is sufficient to show a gap in the value of this program between the "planted" case, where there is some x such that $p_i(x) = z_i$ for every i, and the case where the values z_i are sampled from Z.

If there is some x such that $p_i(x) = z_i$ for every i, then the value of the program (3.1) is at least $\sum_{i=1}^m z_i^2$ which (using the fact that $\mathbb{E}\, z_i^2 = 1$ and standard concentration bounds) we can assume to be very close to m.[12]

On the other hand, consider the case where $(z_1, \ldots, z_m)$ is chosen from Z. For every fixed $x \in \{\pm 1\}^n$, we can define m random variables $Y_1^x, \ldots, Y_m^x$ such that $Y_i^x = z_i p_i(x)$ and let $Y^x = \sum_{i=1}^m Y_i^x$. Since Z is a product distribution, the random variables Y_i^x are independent, and hence we can use the Chernoff bound to show that with all but $0.01 \cdot 2^{-n}$ probability, the value of Y^x will be at most $O(\sqrt{nBm})$, where B is a bound on the magnitude of $z_i p_i(x)$. We can then apply the union bound over all possible x's to show that the value of the quadratic program (3.1) is at most $O(\sqrt{nBm})$ with probability 0.99.

For example, if each z_i is a uniform element in $\{\pm 1\}$, and $|p_i(x)| \leq O(1)$ for every x (as is the case when p_i is a *predicate*), then $B = O(1)$ and so in this case the value of (3.1) will be at most m/c as long as $m \gg c^2 n$. Setting c to the aforementioned approximation factor $O(\log n)$, we get a successful refutation.

The resulting algorithm does the following. On input $z_1, \ldots, z_m$, run the SDP relaxation for (3.1) and if the value is smaller than $m/2$, then output "`refuted`" and declare that z is not in the image of G. In the case where $z = \mathcal{G}(x)$ the value of the quadratic program, and so also its SDP relaxation, will be at least $0.9m$.[13] On the other hand, if $m = \omega(n \log n)$, then with high probability the value of the quadratic program will be $o(m/\log n)$ and hence the relaxation will have value $o(m)$.

In the discussion above we made two key assumptions:

- $|p_i(x)| \leq O(1)$ for every $x \in \{\pm 1\}^n$
- $|z_i| \leq O(1)$ for $x \in \{\pm 1\}^n$

[12] Formally, in the case that $p_i(x) = z_i$ we do not assume anything about the distribution of z. However, if $\sum_{i=1}^m z_i^2 < 0.9m$, we can simply choose to output "?".

[13] We ignore here the case where $\sum z_i^2 < 0.9m$, in which case our algorithm will halt with the output "?".

In general both of these might be false. If p_i has at most s non-zero monomials, and satisfies $\mathbb{E}\, p_i(U)^2 = 1$, then we can show that $|p_i(x)| \leq \sqrt{s}$ for every x, using the known relations between the ℓ_1 and ℓ_2 norms of p_i's Fourier transform. The second condition can be a little more tricky. If the z_i's are *subgaussian*, then we can use Hoeffding's inequality in place of the Chernoff bound, but in general we cannot assume that this is the case. Luckily, it turns out that in our application we can use a simple trick of rejecting outputs in which z_i has unusually large magnitude to reduce to the bounded case. The bottom line is that we get an efficient algorithm for the image-refutation problem of an s-sparse quadratic map whenever $m \gg sn \log n$.

The higher degree case reduces to the degree 2 by "quadratisizing" polynomials. That is, we can consider a degree d polynomial on n variables as a degree 2 polynomial on the $n^{\lceil d/2 \rceil}$ variables obtained by considering all degree $\lceil d/2 \rceil$ monomials. Using this approach, we can generalize our results (at a corresponding loss in the bound on the output) to higher degree maps.

3.1 Distinguishing Generators with Block-Locality 2

A priori the notions of *block locality* and *algebraic degree* seem unrelated to one another. After all, a two block local generator on size b blocks could have degree that is as large as $2b$. However, we can *pre-process* a length bn input $x \in \{\pm 1\}^{bn}$, by mapping it to an input $x' \in \{\pm 1\}^{n'}$ for $n' = 2^b n$ where for every $i \in [n]$, the i^{th} block of x' will consist of the values of all the 2^b monomials on the i^{th} block of x. Note that a map of block locality ℓ in x becomes a map of *degree* ℓ in x'. Moreover, since every output bit depends on at most ℓ blocks, each containing 2^b variables, the number of monomials in this degree ℓ polynomial is at most $2^{\ell b}$.

In this way, we can transform a candidate two block-local pseudorandom generator $\mathcal{G}\colon \{\pm 1\}^{bn} \to \{\pm 1\}^m$ into a degree-2 sparsity-2^{2b} map $\mathcal{G}'\colon \{\pm 1\}^{n'} \to \mathbb{R}^m$. Note that even if $\mathcal{G}$ is a secure pseudorandom generator, it is *not* necessarily the case that $\mathcal{G}'$ is also a pseudorandom generator, as the uniform distribution on $x \in \{\pm 1\}^{bn}$ does not translate to the uniform distribution over $x' \in \{\pm 1\}^{2^b n}$. However, the image of $\mathcal{G}'$ contains the image of $\mathcal{G}$, and hence if we can solve the image refutation problem for $\mathcal{G}'$, then we can do so for $\mathcal{G}$ as well. Applying the above result as a black-box gives an efficient algorithm to break a two block-local generator of block size b as long as the output length m satisfies

$$m \gg 2^{2b} n' \log^2 n = 2^{3b} n \log^2 n \ .$$

This is already enough to break the concrete candidate of Lin and Tessaro [47], but a more refined analysis shows that we can improve the 2^{3b} factor to 2^{2b}. Furthermore, if we initialize the construction with a random predicate on an expanding constraint graph we can bring this factor down to 2^b. Both improvements still use the same algorithm, only providing a tighter analysis of it in these cases. We do not know if our analysis can be improved even further. Mapping out the various trade-offs for block-local generators (or, equivalently, refuting very large alphabet CSPs), is a very interesting open question.

The first improvement, described in Section 5.1, yields a better bound on the output of any two-block-wise generator. As mentioned above, it uses the same algorithm. That is, we take a candidate two-block-local generator $\mathcal{G}\colon \{\pm 1\}^{bn} \to \{\pm 1\}^m$ and transform it into a degree two mapping $\mathcal{G}'\colon \{\pm 1\}^{2^b n} \to \mathbb{R}^m$ by "expanding out" the monomials in each block. We then run the same algorithm as before on the generator $\mathcal{G}'$, but the key idea is that because $\mathcal{G}'$ arose out of the expansion of a two-block-local generator, we can show a better upper bound on the objective value of the quadratic program (3.1). Specifically, we can express each of these polynomials as a function of the Fourier transform of the predicate that the original block local generator applied to each pair of blocks. We can then change the order of summations, which enables us to reduce bounding (3.1) to bounding 2^{2b} "simpler" sums, for which we able to obtain, in the random case, tighter bounds with sufficiently high probability that allows to take a union bound over these 2^{2b} options. See Section 5.1 for the full detail.

3.2 Improving the Stretch to $n2^b$ for the Single Predicate Case

The second improvement (Theorem 5.3), considers the special case where each output of the generator is computed using the same predicate (as discussed before, this case is the principle focus of [49]). In this case, we show that our image refutation algorithm works whenever m (the number of outputs) of the generator satisfies $m = \tilde{\Omega}(n2^b)$. This matches the stretch required for the *distinguisher* of [49] to work.

We now describe at a high level, how our refutation algorithm works. The refutation algorithm is given a string $z \in \{\pm 1\}^m$ and description of the generator $\mathcal{G}$ that includes the underlying graph G on n vertices and the predicate $P\colon \{\pm 1\}^b \times \{\pm 1\}^b \to \{\pm 1\}$. As a first step, we will reduce the problem of image refuting $\mathcal{G}$ to image refuting a somewhat simpler $\mathcal{G}'$ where the predicate P will be replaced by a "product-predicate" P'. A predicate $P'\colon [q] \times [q] \to \{\pm 1\}$ is a *product* predicate if it can be written as a product of two functions $f\colon [q] \to \{\pm 1\}$ and $g\colon [q] \to \{\pm 1\}$ applied to each of the inputs to P. In the second step, we will give an efficient algorithm for image-refuting two-block-local, single product predicate PRG.

We now describe the first step. Here, the algorithm wishes to certify that there's no $x \in (\{\pm 1\}^b)^n$ such that $\mathcal{G}(x) = z$. Fix any $x \in (\{\pm 1\}^b)^n$. For this fixed x, consider the distribution $\mathcal{D}$ on inputs to P, generated by taking a random edge $\{i, j\}$ in G and outputting (x_i, x_j). We will show, using a result of Linial and Schraibman shown in the context of relating marginal complexity to various measures of communication complexity, that on $\mathcal{D}$ (more generally, any distribution on inputs to P), there's a product predicate $F(\alpha, \beta) = f(\alpha) \cdot g(\beta)$ such that $\mathbb{E}_{(\alpha,\beta)\sim\mathcal{D}}[P(\alpha, \beta) \cdot F(\alpha, \beta)] \geq \Theta(2^{-b/2})$. Thus, if there is an $x \in (\{\pm 1\}^b)^n$ such that $\mathcal{G}(x) = z$, then for the same x, $\mathbb{E}_{i\sim[m]}[\mathcal{G}'(x)_i \cdot z_i] \geq \Theta(2^{-b/2})$. If we can now certify an upper bound of $\ll 2^{-b/2}$ on $\mathbb{E}_{i\sim[m]}[\mathcal{G}'(x)_i \cdot z_i]$ for every x and with high probability over the draw of z, we'd obtain an image refutation algorithm. This latter question turns out to be simpler because of the product nature of the predicate defining $\mathcal{G}'$.

This step in our algorithm is inspired by the use of a result of Chor-Goldreich in the work of [49]. This lemma says[14] that for the uniform distribution on the inputs to P, there's a product predicate that has a correlation of $\Theta(2^{-b/2})$ with P. In the work of [49] this observation is used to replace P by a *constant-alphabet* predicate (obtained by massaging the constituents of the product predicate given by Chor-Goldreich lemma above) to obtain a simplified PRG on constant-alphabet size such that when the seed is chosen according to the uniform distribution on $(\{\pm 1\}^b)^n$, the modified PRG's output distribution correlates well with that of the original one. Thus, a strong enough refutation algorithm (they use one due to [1]) applied to the modified PRG is enough to give a distinguisher. Observe that this approach doesn't give a refutation algorithm because the key step of replacing P with $f \cdot g$ relies on x being drawn uniformly from $[q]^n$.

Instead of using off-the-shelf refutation algorithms (such as that of [2]), we solve the image refutation problem for single product predicate block-local PRGs by giving a direct, simple algorithm – this algorithm crucially works without the knowledge of the product predicate itself or even the block size parameter b. This is important, as our argument that obtains $\mathcal{G}'$ is not constructive, in particular, the distribution that the product predicate approximates P on is a complicated function of the (purported) arbitrary assignment x and the graph G. Thus, our product-predicate refutation algorithm must work without the explicit knowledge of the underlying product predicate.

Indeed, we show (in the full version [12]) that given a graph G on n vertices with $m \gg n$ edges and any string z, we can (in one shot) show that z (w.h.p) is not in the image of *any* of the (infinitely many!) generators obtained by using any two-block-local product predicate of arbitrarily large block size with the same underlying graph G. In particular, our refutation algorithm does not need to know the predicate itself or even the number of bits in each block of the seed for the generator!

3.3 Random Block Local Generators

We analyze the natural candidate of multiple-predicate, block-local generators, where both the underlying graph and each of the predicates are chosen uniformly at random (conditioned on the predicates being balanced), and show (see Section 5.3) that our refutation algorithm works whenever $m = \Omega(n2^b)$. As before, our idea to consider the problem of maximizing the polynomial $\sum_i z_i p_i(x)$. We work with the *matrix* M such that our target polynomial $\sum_i z_i p_i(x)$ is a bilinear form of M. To obtain a certificate for the upper bound on the polynomial, it then suffices to show a strong enough upper bound on the *spectral* norm of the matrix M – which we show is small enough (w.h.p) because of the randomness involved in defining the generator. M has some dependencies between its various entries that preclude the use of standard bounds to upper bound the spectral

[14] We use a somewhat different way to describe the use Chor-Goldreich lemma by [49] in order to show how it inspires our approach.

norm. So we compute an upper bound on the spectral norm using the standard trace method that reduces the problem to some combinatorial properties that are simple to reason about.

4 Image Refutation for Low Degree Maps

In this section we will prove our main technical theorem, which is an algorithm for the image refutation problem for every low degree map and "nice" or "non-degenerate" product distributions. We start by defining the notion of non-degenerate distributions, which amounts to distributions that do not put almost all their probability mass on a small (compared to their standard deviation) interval.

Definition 4.1 (c-spread distributions). *Let Z be a product distribution over $\mathbb{R}^m$ with $\mathbb{E} Z_i = 0$ and $\mathbb{E} Z_i^2 = 1$ for every i. We say that Z is c-*spread *if for every interval $I \subseteq \mathbb{R}$ of length $1/c$, the probability that $Z_i \in I$ is at most* 0.9*.*

Normalized low-degree maps are polynomials over $\{\pm 1\}^n$ - we use the standard Fourier basis (e.g., see [53]) to represent them:

Definition 4.2 (Fourier notation). *For any $S \subseteq [n]$, let $\chi_S(x) = \Pi_{i \in S} x_i$ for any $x \in \{\pm 1\}^n$. A function $p \colon \{\pm 1\}^n \to \mathbb{R}$ can be uniquely expanded as $\sum_{S \subseteq [n]} \hat{p}(S)\chi_S$ where the "Fourier coefficients" $\hat{p}(S) = \mathbb{E}_{x \sim \{\pm 1\}^n}[\chi_S(x)p(x)]$ and the expectation is over the uniform distribution over the hypercube $\{\pm 1\}^n$. Fourier coefficients satisfy the Parseval's theorem: $\mathbb{E}_{x \sim \{\pm 1\}^n} p(x)^2 = \sum_{S \subseteq [n]} \hat{p}(S)^2$.*

We define a normalized degree d map to be a collection of degree d polynomials $\bar{p} = (p_1, \ldots, p_m)$ mapping $\{\pm 1\}^n$ to $\mathbb{R}^m$ such that $\mathbb{E}\, p_i(U) = 0$ and $\mathbb{E}\, p_i(U)^2 = 1$ for every i where U is the uniform distribution.[15]

Our main technical theorem is the following:

Theorem 4.3 (Main theorem). *There is an efficient algorithm that solves the refutation problem for every normalized degree d map $\bar{p}$ and c-spread probability distribution Z as long as*

$$m > K \cdot c^2 s(\bar{p}) n^{\lceil d/2 \rceil} \log^2(n) \tag{4.1}$$

for some global constant K.

[15] Note that we are using the same normalization for the Z_i's and $p_i(U)$, which makes sense in the context of a pseudorandom generator applied to the uniform distribution over the seed. If we wanted to consider other distributions D over the seed, we would need to require that $\mathbb{E}\, p_i(D)^2$ is not much smaller than $\mathbb{E}\, p_i(U)^2$. This condition is satisfied by many natural distributions.

To state the result in a stronger form, we use a somewhat technical definition for the parameter $s(\bar{p})$, which is deferred till later (see Equation (4.5) and Definition 4.9 below). However, one important property of it is that for every normalized polynomial map $\bar{p} = (p_1, \ldots, p_m)$, $s(\bar{p})$ is smaller than the maximum *sparsity* (i.e., number of monomials) of the polynomials. Hence, Theorem 4.3 implies Theorem 1.5 from Section 1.1. The fact that we only require a factor of $s(\bar{p})$ as opposed to the sparsity makes our result stronger, and in some cases this difference can be very significant.

The algorithm for proving Theorem 4.3 is fairly simple:

```
Refutation algorithm
Input: $z \in \mathbb{R}^m$, $p_1, \ldots, p_m$ normalized polynomials of degree $d$ in $\{\pm 1\}^n$.
Output: ”refuted” or ”?”.
Operation:

  1. Let $I = \{i \in [m] : z_i^2 \leq 100\}$. Let $\mu_i$ be the conditional expectation of $z_i$
     conditioned on $z_i^2 \leq 100$.
  2. If $\sum_{i \in I} (z_i - \mu_i)^2 < m/(10c)$ return ”?”.
  3. Let $\theta$ be the value of the degree $\lceil d/2 \rceil$ SOS relaxation for the degree $d$
     polynomial optimization problem

         $\max_{x \in \{\pm 1\}^n} \sum_{i \in I} (z_i - \mu_i) p_i(x)$        (4.2)

  4. Return ”refuted” if $\theta - \sum_{i \in I} \mu_i (z_i - \mu_i) < m/(10c)$ otherwise return ”?”.
```

The *degree d sum of squares program* is a semidefinite programming relaxation to a polynomial optimization problem, which means that the value θ is always an upper bound on (4.2). The most important fact we will use about this program is the *symmetric Grothendieck Inequality* of Charikar and Wirth [19], which states that in the important case where $d = 2$, the *integrality gap* of this program (i.e., ratio between its value and the true maximum) is $O(\log n)$.

For this case, where $d = 2$, this program is equivalent to the semidefinite program known as the *basic SDP* relaxation for the corresponding quadratic program. This means that θ can also be computed as

$$\max_{\substack{X \in \mathbb{R}^{(n+1)\times(n+1)} \\ X \succeq 0,\ X_{ii} = 1\ \forall i}} \operatorname{tr}(A \cdot X)\ , \tag{4.3}$$

where A is an $(n+1) \times (n+1)$ matrix that *represents* the quadratic polynomial $\sum_{i \in I} (z_i - \mu_i) p_i$, in the sense that for every $i, j \in [n]$, $A_{i,j}$ corresponds to the coefficient of $x_i x_j$ in this polynomial, and for every $i \in [n]$, $A_{i,n+1} = A_{n+1,i}$ is the coefficient of x_i.

We now turn to proving Theorem 4.3. We start by showing the case that $d = 2$. The proof for general degree will follow by a reduction to that case.

4.1 Degree 2 Image Refutation

In this section, we prove Theorem 4.3 for the case $d = 2$, which is restated below as the following lemma:

Lemma 4.4 (Image refutation for degree 2). *There is an efficient algorithm that solves the refutation problem for every normalized degree 2 map $\bar{p}$ and c-spread probability distribution Z as long as*

$$m > K \cdot c^2 s(\bar{p}) n \log^2 n \tag{4.4}$$

for some absolute constant $K > 0$.

In this case, the parameter $s(\bar{p})$ is defined as follows:

$$s(p_1, \ldots, p_m) = \tfrac{1}{m} \max_{x \in \{\pm 1\}^n} \sum_{i=1}^{m} p_i(x)^2 \tag{4.5}$$

By expanding each p_i in the Fourier basis as $p_i = \sum \hat{p_i}(S) \chi_S$, we can see that $\max_{x \in \{\pm 1\}^n} |p_i(x)| \leq \sum |\hat{p_i}|$. Hence, in particular, $s(\bar{p})$ is smaller than the average of the ℓ_1 norm squared of the p_i's Fourier coefficients. Using the fact that $\mathbb{E}\, p_i(U)^2 = 1$, and the standard relations between the ℓ_1 and ℓ_2 norms, we can see that if every one of the p_i polynomials has at most s monomials (i.e., non-zero Fourier coefficients), then $s(\bar{p}) \leq s$.

We now prove Lemma 4.4. To do so, we need to show two statements:

- If $z = \bar{p}(x)$, then the algorithm will never output "refuted".
- If z is chosen at random from Z, then the algorithm will output "refuted" with high probability.

We start with the first and easiest fact, which in fact holds for *every* degree d.

Lemma 4.5. *Let $z \in \mathbb{R}^m$ be such that there exists an x^* such that $p_i(x^*) = z_i$. Then, the algorithm does not output "refuted".*

Proof. Suppose otherwise. We can assume that $\sum_{i \in I}(z_i - \mu_i)^2 \geq m/(10c)$ as otherwise we will output "?". Since the SDP is a relaxation, in particular, the value θ is larger than $\sum_{i \in I}(z_i - \mu_i) p_i(x^*) = \sum_{i \in I}(z_i - \mu_i) z_i$ under our assumption. Hence, $\theta - \sum_{i \in I}(z_i - \mu_i)\mu_i \geq \sum_{i \in I}(z_i - \mu_i)^2 \geq m/(10c)$

We now turn to the more challenging part, which is to show that the algorithm outputs "refuted" with high probability when z is sampled from Z. We start by observing that by Markov's inequality, for every i, the probability that $z_i^2 > 100\, \mathbb{E}\, z_i^2 = 100$ is at most 0.99. Hence, the expected size of the set I defined by the algorithm is at least $0.99m$ and using Chernoff's bound it follows with very high probability that $|I| > 0.9m$. Let Z'_i be the random variable Z_i conditioned on the (probability ≥ 0.99) event that $Z_i^2 \leq 100$, and $\mu_i = \mathbb{E}\, Z'_i$. Note that by definition $(Z'_i)^2 \leq 100$ with probability 1, i.e. $|Z'_i| \leq 10$ with probability

1, which in turn implies that $|\mu_i| \leq 10$. By the “spread-out-ness” condition on Z_i and the union bound, $\mathbb{P}[Z_i' \notin [\mu_i - \frac{1}{2c}, \mu_i + \frac{1}{2c}] \geq 0.1 - 0.01$ and hence, in particular, $\mathbb{E}[(Z_i' - \mu_i)^2] \geq \frac{1}{500c^2}$.

We can consider the process of sampling the z_i values from the algorithm as being obtained by first choosing the set I, and then sampling z_i independently from the random variable Z_i' for every coordinate $i \in I$. The following lemma says that there will not be an *integral* (i.e., $\{\pm 1\}$-valued) solution to the SDP with large value.

Lemma 4.6. *With probability at least* 0.99 *it holds that for every* $x \in \{\pm 1\}^n$*,*

$$\sum_{i \in I} (z_i' - \mu_i) p_i(x) \leq O(\sqrt{nms(\bar{p})}) \tag{4.6}$$

Proof. We use the union bound. For every fixed $x \in \{\pm 1\}^n$, we let $\alpha_i = p_i(x)$. We know that $\sum_{i \in I} \alpha_i^2 \leq \sum_{i=1}^m \alpha_i^2 \leq \max_{x \in \{\pm 1\}^n} \sum p_i(x)^2 = ms(\bar{p})$. Since $|z_i' - \mu_i| \leq 20$, it follows that $(z_i' - \mu_i)$ is sub-gaussian with constant standard deviation. Therefore, $\sum_{i \in I}(z_i' - \mu_i)\alpha_i$ is sub-gaussian with zero expectation standard deviation $O(\sqrt{ms(\bar{p})})$. Therefore, there exists a value $O(\sqrt{nms(\bar{p})})$ s.t. the probability that $\sum_{i \in I}(z_i' - \mu_i)\alpha_i$ exceeds it is smaller than $0.001 \cdot 2^{-n}$. Applying the union bound implies the lemma.

Lemma 4.4 will follow from Lemma 4.6 using the fact that the SDP gives $O(\log n)$ approximation factor for true maximum. In particular the symmetric version of Grothendieck inequality shown by [19] implies that the value θ computed by the algorithm is at most a factor of $O(\log n)$ larger than the true maximum of the integer program (4.2), see Theorem A.3 in Appendix A.

To finish the proof, we need to ensure that (after multiplying by $O(\log n)$) the bound on the RHS of (4.6) will be smaller than $m/(100c) + \sum_{i \in I}(z_i - \mu_i)\mu_i$. Indeed, since $|\mu_i| \leq 10$, with high probability over the choice of the z_i's (which are chosen from Z_i'), the quantity $\sum_i (z_i - \mu_i)\mu_i$ is at most, say, 10 times the standard deviation, which is $O(\sqrt{m}) \ll m/c$. (Here no union bound is needed.) So, by plugging in (4.6) what we really need is to ensure that

$$m/(20c \log n) \geq O(\sqrt{nms(\bar{p})})$$

or that

$$m \geq O(ns(\bar{p})c^2 \log^2 n)$$

which exactly matches the conditions of Lemma 4.4 hence concluding its proof (and hence the proof Theorem 4.3 for the $d = 2$ case).

4.2 Refutation for $d > 2$

In this section, we show how to reduce the general degree d case to the case $d = 2$, hence completing the proof of Theorem 4.3. The main tool we use is the notion of "quadratizing" a polynomial. That is, we can convert a degree d polynomial p on n variables into a degree two polynomial $\tilde{p}$ on $(n+1)^{\lceil d/2 \rceil}$ variables by simply encoding every monomial of degree up to $\lceil d/2 \rceil$ of the input as a separate variable.

Definition 4.7 (Quadratization). *Let p be a degree d polynomial on $\mathbb{R}^n$ which we write in Fourier notation (see Definition 4.2) as $p = \sum_{|S| \le d} \hat{p}(S)\chi_S$. Let $d' = \lceil d/2 \rceil$ Then the quadratization of p is the degree two polynomial q on $\binom{n}{\le d'}$ variables defined as:*

$$q(y) = \sum_{S,T} \hat{p}(S \cup T) y_S y_T,$$

where the elements of the $\binom{n}{\le d'}$ dimensional vector y are indexed by sets of size at most d', and this sum is taken over all sets $S, T \subseteq [n]$ of size at most d' such that every element in S is smaller than every element of T, $|S| = \max\{|S \cup T|, d'\}$.

The following simple properties ensured by quadratization are easy to verify:

Lemma 4.8. *Let q be the quadratization of a degree d polynomial p on $\binom{n}{\le d'}$ variables for $d' = \lceil d/2 \rceil$. Then,*

1. *For any $x \in \{\pm 1\}^n$ there exists $y \in \{\pm 1\}^{\binom{n}{\le d'}}$ such that $q(y) = p(x)$.*
2. $\sum_{S,S'} \hat{q}(\{S, S'\})^2 = \sum_T \hat{p}(T)^2$.
3. $\max_{y \in \{\pm 1\}^{\binom{n}{\le d'}}} q(y) \le \sum_{|T| \le d} |\hat{p}(T)|$.

Proof (sketch). For 1, we let $y_S = \chi_S(x)$ for every $|S| \le d'$. For 2 and 3, we note that the set of nonzero Fourier coefficients of p and q is identical because for every set $|U| \le d$ there is a unique way to split it into disjoint sets S, T of size at most d' where S is the first $\min\{|U|, d'\}$ coordinates of U, and $\hat{q}(\{S, T\}) = \hat{U}$. For all other pairs S, T that do not arise in this manner, it will hold that $\hat{q}(\{S, T\}) = 0$. This means that both the ℓ_1 and ℓ_2 norms of the vector $\hat{q}$ are the same as that of the vector $\hat{p}$, implying both 2 and 3.

We define the complexity of the degree d normalized map $\bar{p}$ as the complexity of the degree 2 normalized map of the quadratizations of p_is:

Definition 4.9 (Complexity of degree d normalized maps). *Let $\bar{p}$ be a normalized degree d map and let $\bar{q}$ be its quadratization. Then, we define $s(\bar{p})$ as $s(\bar{q})$ from (4.5).*

Remark 4.10. Part 2 of Lemma 4.8 shows that if $\bar{p}$ is normalized the so is its quadratization $\bar{q}$. Part 3 of Lemma 4.8 shows that $s(\bar{p}) \le \text{sparsity}(p)$ for any normalized degree d map p.

We can now complete the proof of Theorem 4.3.

Proof (of Theorem 4.3). Let $\bar{p} = (p_1, \ldots, p_m)$ be a normalized degree d polynomial map and let $z_1, \ldots, z_m$ be the inputs given to the algorithm. If there is an x such that $p_i(x) = z_i$ for every i, then by Lemma 4.5 (which did not assume that $d = 2$), the algorithm will return "?".

Suppose otherwise, that $z_1, \ldots, z_m$ are chosen from the distribution Z. Recall that our algorithm computes θ to be the value of the degree $2d'$ SOS relaxation for the quadratic program (4.2). This value satisfies

$$\theta = \max_{\mu(x)} \tilde{\mathbb{E}}_{\mu} \left[\sum_{i \in I} (z_i - \mu_i) p_i(x) \right] ,$$

where the maximum is over all degree $2d'$ pseudo-distributions satisfying $\{x_i^2 = 1\}$ for every $i \leq n$.

If μ is a degree $2d'$ pseudodistribution over $\{\pm 1\}^n$ then we can define a degree 2 pseudodistribution μ' over $\{\pm 1\}^{\binom{n}{d'}}$ by having $y \sim \mu$ be defined as $y_S = \chi_S(x)$ for $x \sim \mu$.[16] Let $\bar{q} = (q_1, \ldots, q_m)$ be the quadratization of $\bar{p} = (p_1, \ldots, p_m)$. Then the distribution μ' above demonstrates that $\theta \leq \theta'$ where

$$\theta' = \max_{\mu'(y)} \tilde{\mathbb{E}}_{\mu'} \left[\sum_{i \in I} (z_i - \mu_i) q_i(x) \right] .$$

But since this is the value of a degree two SDP relaxation for a quadratic program, we know by Theorem A.3 that it provides an $O(\log n)$ approximation factor, or in other words that

$$\theta' \leq O(\log n) \max_{y \in \{\pm 1\}^{\binom{n}{d'}}} \sum_{i \in I} (z_i - \mu_i) q_i(y) \ . \tag{4.7}$$

Since the q_i's are degree two polynomials over $O(n^{d'})$ variables, Lemma 4.6 implies that when $z_1, \ldots, z_m$ are randomly chosen from Z, w.h.p. the RHS of (4.7) is at most $O((\log n)\sqrt{n^{d'} ms(\bar{q})}) = O((\log n)\sqrt{n^{d'} ms(\bar{p})})$. Setting this to be smaller than $(m/10c^2)$ recovers Theorem 4.3.

5 Block Local Generators

Recall that a map $\mathcal{G}\colon \{\pm 1\}^{bn} \to \{\pm 1\}^m$ is ℓ *block-local* if the input can be separated into n blocks of b bits each[17], such that every output of $\mathcal{G}$ depends on at most ℓ blocks.

In this section we will show tighter bounds for block-local generators than those derived from the theorem in Section 4. Of particular interest is the case

[16] While it is clear that this operation makes sense for actual distributions, it turns out to be not hard to verify that it also holds for pseudodistributions, see the lecture notes [16].

[17] Our algorithm works even if the blocks intersect arbitrarily. The construction in [47] uses only non-intersecting blocks.

of block-locality 2 due to its applications for obfuscation from bilinear maps. In Section 5.1 we show a tighter analysis of our algorithm from Section 4 for any block-local generator. This yields a distinguisher for any block-locality 2 generator with $m \gg 2^{2b} n \log n$. In Section 5.3, we analyze a particularly natural instantiation for 2-block-local PRGs - a random predicate and random constraint graph and show that our distinguisher works for an even smaller $m \gg 2^b n$. In fact, we show that one can even use a simpler distinguisher that computes the largest singular value of a certain matrix arising out of the input instead of running a semidefinite program.

5.1 Bounds on General Block-Local Generators

In this subsection we prove the following result:

Theorem 5.1 (Limitations of block local generators). *For every ℓ-block-local $\mathcal{G}\colon \{\pm 1\}^{bn} \to \{\pm 1\}^m$ there is an efficient algorithm for the $\mathcal{G}$ image refutation problem w.r.t. the uniform distribution over $\{\pm 1\}^m$ as long as*

$$m > (K \log n) 2^{\ell b} (n + 2\ell b)^{\lceil \ell/2 \rceil},$$

where K is a constant depending only on ℓ.

If ℓ is constant and $b = o(n)$ (as is mostly the case), the above translates to refutation for $m > (K \log n) 2^{\ell b} n^{\lceil \ell/2 \rceil}$.

The proof of this theorem can be found in the full version [12].

Theorem 1.3 from the introduction is the special case of Theorem 5.1 for the case $\ell = 2$, and so in particular Theorem 5.1 breaks any 2 block local pseudorandom generator with stretch $\tilde{\Omega}(n 2^{2b})$ to instantiate the bilinear-map based construction of iO of [47].

Remark 5.2. A slightly weaker bound can be obtained by a direct application of Theorem 4.3. We sketch the argument in the full version [12].

5.2 Sharper Bounds on the Stretch of Block-Local PRGs with a Single Predicate

Next, we prove a tighter upper bound of $\tilde{\Theta}(n 2^b)$ on the stretch of a block local PRGs with a *single* predicate P (instead of a different predicate for each output) with block-locality 2. The following is the main result of this section:

Theorem 5.3. *For $b \in \mathbb{N}$, let $\mathcal{G}\colon \{\pm 1\}^{bn} \to \{\pm 1\}^m$ be a two block-local PRG defined by an instance graph $G([n], E)$ with $m = |E|$ edges and an arbitrary predicate $P\colon \{\pm 1\}^b \to \{\pm 1\}^b \to \{\pm 1\}$ such that for any seed $x \in (\{\pm 1\}^b)^n$, for every $e \in E$, $\mathcal{G}_e = P(x_{e_1}, x_{e_2})$. Let $z \in \{\pm 1\}^m$.*

Then, for any $m > O(\log^2(n)) n 2^b$, there exists a poly(m, n) *time algorithm that takes input G, z and P and outputs* "`refuted`" *or* "?" *with the following guarantees:*

1. *If the output is* "`refuted`", *then,*

$$\max_{x\in(\{\pm1\}^b)^n} \sum_{(i,j)\in E} P(x_{e_1}, x_{e_2}) z_e < 0.99m.$$

2. *When* $z \in \{\pm1\}^m$ *is chosen uniformly at random, then* $\mathbb{P}[$ *Algorithm outputs* "`refuted`"$] > 1 - 1/n$.

The proof of this theorem can be found in the full version [12].

5.3 Image Refutation for Random Block-Local PRGs

A particularly appealing construction of block local PRGs is obtained by instantiating them with a random graph with $\sim m$ edges and a random and independent predicate for every edge. A priori, the randomness in this construction could appear to *aid* the security of the PRG. Indeed, such instantiations are in fact suggested by [47]. We show that in this case, as in the previous section where all predicates are identical, we can show a *stronger* upper bound on the stretch of the local PRG in terms of the block size b. Whereas in Section 5.1, for general block-local PRGs with non-identical predicates, we lost a factor of $2^{2b}\log(n)$ in the output length, for the special case of a random graphs and random, independent predicates, this can be improved to $\Theta(2^b)$ as we show in this section. We note that the only property of random graphs that we use is expansion.

More concretely, in this section, we analyze the stretch of the following candidate construction of a block-local PRG.

- We choose a graph $G([n], q)$ where every edge is present in G with probability $q = \frac{m}{\binom{n}{2}}$. Thus, with high probability, the number of edges in the graph is $m \pm \sqrt{m}$.
- For every edge $\{i, j\}$ in G, we choose a uniformly random predicate $P_{i,j}(x, y) = \pm1$ conditioned on $P_{i,j}$s being balanced, i.e. $\mathbb{E}_{x,y\sim\{\pm1\}^b} P_{i,j}(x, y) = 0$.
- On input (seed) $x \in \{\pm1\}^{bn}$, which we think of as partitioned into blocks $x_1, \ldots, x_n \in \{\pm1\}^b$, the generator outputs $h_{i,j}(x_i, x_j)$ for every edge (i, j) of G.

Theorem 5.4 (Limitations of random block-local generators). *There is some constant K such that if $\mathcal{G}\colon \{\pm1\}^{bn} \to \{\pm1\}^m$ is a generator sampled according to the above model and $m \geq K2^b n \log^3(n)$, then w.h.p. there is a polynomial-time algorithm for the $\mathcal{G}$ image refutation problem w.r.t. the uniform distribution over $\{\pm1\}^m$.*

The proof of this theorem can be found in the full version [12].

6 Lower Bound for Refuting Two-Block-Local PRGs

In this section, we establish that if $b > 10 \log \log(n)$, then there's no $2^{O(n/2^{4b})}$-time algorithm for image refutation of block-local PRG of stretch $\Omega(n2^b)$ based on the sum-of-squares method.

The main goal of this section is summarized in the following theorem.

Theorem 6.1. *For any $b > 10 \log \log(n)$, there's a construction $\mathcal{G} \colon \{\pm 1\}^n \to \{\pm 1\}^m$ for $m = \Omega(n2^b)$ such that for any $z \in \{\pm 1\}^m$, there's a feasible solution for the degree $\Theta(n/2^{4b})$ sum-of-squares relaxation of the constraints $\{\mathcal{G}_i = z_i\}$. In particular, sum of squares algorithm of degree $\Theta(n/2^{4b})$ cannot accomplish image refutation for $\mathcal{G}$.*

The proof of this theorem can be found in the full version [12].

7 A Class of Block-Local Candidate Pseudorandom Generators

In this section we outline a simple candidate pseudorandom generator of degree d that has potentially output length as large as $n^{d/2-\varepsilon}$. We have not conducted an extensive study of this candidate's security, but do believe it's worthwhile example as a potential counterpoint to our results on limitations for pseudorandom generator, demonstrating that they might be tight.

The idea is simple: for a finite group $\mathbb{G}$ that does not have any abelian quotient group (for example, a non-abelian simple group will do), we choose dm random indices $\{i_{j,k}\}_{j \in [m], k \in [d]}$ and let $\mathcal{G}$ be the generator mapping $\mathbb{G}^n$ to $\mathbb{G}^m$ where

$$\mathcal{G}(x)_j = x_{i_{j,1}} * x_{i_{j,2}} * \cdots * x_{i_{j,d}} \tag{7.1}$$

If want to output m bits rather than m elements of $\mathbb{G}$, then we use a group $\mathbb{G}$ of even order and apply to each coordinate some balanced map $f \colon \mathbb{G} \to \{0, 1\}$. For every group element $g \in \mathbb{G}$, the predicate

$$x_1 * \cdots * x_d = g \tag{7.2}$$

supports a $d - 1$ wise independent distribution. Hence, using the results of [43] we can show that as long $m < n^{d/2-\varepsilon}$, for a random $z \in \mathbb{G}^m$, the SOS algorithm cannot be used to efficiently refute the statement that $z = \mathcal{G}(x)$ for some x.

Ruling out Gaussian-elimination type attacks is trickier. For starters, solving a linear system over a non-abelian group is NP-hard [35,42]. Also, Applebaum and Lovett [10, Theorem 5.5] showed that at least for the large d case, because the predicate (7.2) has rational degree d, the image-refutation problem for this generator is hard with respect to algebraic attacks (that include Gaussian elimination) for $m = n^{\Omega(d)}$. Nevertheless, there are non trivial algorithms in the group theoretic settings (such as the low index subgroup algorithm, see [23] and [57, Sec. 6]). A more extensive study of algebraic attacks against this predicate is needed to get better justifications of its security, and we leave such study for future work.

We remark that the condition that the group $\mathbb{G}$ does not have abelian normal subgroups is crucial. Otherwise, we can write $\mathbb{G}$ as the direct product $\mathbb{H} \times \mathbb{H}'$ where $\mathbb{H}$ is abelian, and project all equations to their component in $\mathbb{H}$. We will get m random equations in n variables over the abelian group $\mathbb{H}$, and hence we can use Gaussian elimination to refute those.

Acknowledgements

We thank Prabhanjan Ananth, Dakshita Khurana and Amit Sahai for discussions regarding the class of generators needed for obfuscation. Thanks to Rachel Lin and Stefano Tessaro for discussing the parameters of their construction with us. We thank Avi Wigderson and Andrei Bulatov for references regarding Gaussian elimination in non-abelian groups.

References

1. Allen, S.R., O'Donnell, R., Witmer, D.: How to refute a random CSP. In: FOCS. pp. 689–708. IEEE Computer Society (2015) 8, 13, 17
2. Allen, S.R., O'Donnell, R., Witmer, D.: How to refute a random CSP. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science—FOCS 2015, pp. 689–708. IEEE Computer Soc., Los Alamitos, CA (2015) 17
3. Ananth, P., Brakerski, Z., Khurana, D., Sahai, A.: Private communication (2017) 3, 12
4. Ananth, P., Jain, A., Sahai, A.: Indistinguishability obfuscation from functional encryption for simple functions. IACR Cryptology ePrint Archive 2015, 730 (2015) 2
5. Ananth, P., Sahai, A.: Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Advances in Cryptology - EUROCRYPT. vol. 10210, pp. 152–181 (2017) 2, 3, 11
6. Applebaum, B.: Pseudorandom generators with long stretch and low locality from random local one-way functions. SIAM J. Comput. 42(5), 2008–2037 (2013) 2, 7
7. Applebaum, B.: Cryptographic hardness of random local functions - survey. Computational Complexity 25(3), 667–722 (2016) 2
8. Applebaum, B., Barak, B., Wigderson, A.: Public-key cryptography from different assumptions. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC. pp. 171–180. ACM (2010) 2
9. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography in NC^0. SIAM J. Comput. 36(4), 845–888 (2006) 2, 7
10. Applebaum, B., Lovett, S.: Algebraic attacks against random local functions and their countermeasures. In: STOC. pp. 1087–1100. ACM (2016) 7, 26
11. Applebaum, B., Raykov, P.: Fast pseudorandom functions based on expander graphs. In: Theory of Cryptography - 14th International Conference, TCC 2016-B. vol. 9985, pp. 27–56 (2016) 2, 7
12. Barak, B., Brakerski, Z., Komargodski, I., Kothari, P.K.: Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). IACR Cryptology ePrint Archive 2017, 312 (2017) 10, 11, 12, 17, 24, 25, 26

13. Barak, B., Chan, S.O., Kothari, P.K.: Sum of squares lower bounds from pairwise independence [extended abstract]. In: STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing, pp. 97–106. ACM, New York (2015) 9
14. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. In: Advances in Cryptology - CRYPTO. Lecture Notes in Computer Science, vol. 2139, pp. 1–18. Springer (2001) 10
15. Barak, B., Raghavendra, P., Steurer, D.: Rounding semidefinite programming hierarchies via global correlation. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011, pp. 472–481. IEEE Computer Soc., Los Alamitos, CA (2011), http://dx.doi.org/10.1109/FOCS.2011.95 8
16. Barak, B., Steurer, D.: Proofs, beliefs, and algorithms through the lens of sum-of-squares (2017), lecture notes, available on http://sumofsquares.org 6, 23, 30
17. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003) 11
18. Brakerski, Z., Gentry, C., Halevi, S., Lepoint, T., Sahai, A., Tibouchi, M.: Cryptanalysis of the quadratic zero-testing of GGH. Cryptology ePrint Archive, Report 2015/845 (2015) 11
19. Charikar, M., Wirth, A.: Maximizing quadratic programs: Extending grothendieck's inequality. In: FOCS. pp. 54–60. IEEE Computer Society (2004) 14, 19, 21, 31
20. Cheon, J.H., Fouque, P., Lee, C., Minaud, B., Ryu, H.: Cryptanalysis of the new CLT multilinear map over the integers. In: Advances in Cryptology - EUROCRYPT. vol. 9665, pp. 509–536. Springer (2016) 11
21. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Advances in Cryptology – EUROCRYPT '15. pp. 3–12 (2015) 11
22. Cheon, J.H., Jeong, J., Lee, C.: An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. Cryptology ePrint Archive, Report 2016/139 (2016) 11
23. Conder, M., Dobcsányi, P.: Applications and adaptations of the low index subgroups procedure. Mathematics of computation 74(249), 485–497 (2005) 26
24. Coron, J., Gentry, C., Halevi, S., Lepoint, T., Maji, H.K., Miles, E., Raykova, M., Sahai, A., Tibouchi, M.: Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In: Advances in Cryptology – CRYPTO '15. pp. 247–266 (2015) 11
25. Coron, J., Lepoint, T., Tibouchi, M.: Practical multilinear maps over the integers. In: Advances in Cryptology - CRYPTO. pp. 476–493 (2013) 11
26. Coron, J., Lepoint, T., Tibouchi, M.: New multilinear maps over the integers. In: Advances in Cryptology - CRYPTO. pp. 267–286 (2015) 11
27. Cryan, M., Miltersen, P.B.: On pseudorandom generators in NC. In: 26th International Symposium on Mathematical Foundations of Computer Science, MFCS. pp. 272–284 (2001) 7
28. Feige, U.: Relations between average case complexity and approximation complexity. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing. pp. 534–543 (electronic). ACM, New York (2002), http://dx.doi.org/10.1145/509907.509985 8
29. Feige, U.: Refuting smoothed 3CNF formulas. In: FOCS. pp. 407–417. IEEE Computer Society (2007) 8

30. Feige, U., Ofek, E.: Easily refutable subformulas of large random 3CNF formulas. Theory Comput. 3, 25–43 (2007), http://dx.doi.org/10.4086/toc.2007.v003a002 8
31. Feldman, V., Perkins, W., Vempala, S.: On the complexity of random satisfiability problems with planted solutions. In: STOC. pp. 77–86. ACM (2015) 7
32. Garg, S., Gentry, C., Halevi, S.: Candidate multilinear maps from ideal lattices. In: Advances in Cryptology - EUROCRYPT. pp. 1–17 (2013) 11
33. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013. pp. 40–49 (2013), http://dx.doi.org/10.1109/FOCS.2013.13 11
34. Gentry, C., Gorbunov, S., Halevi, S.: Graph-induced multilinear maps from lattices. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC. pp. 498–527 (2015) 11
35. Goldmann, M., Russell, A.: The complexity of solving equations over finite groups. Inf. Comput. 178(1), 253–262 (2002), http://dx.doi.org/10.1006/inco.2002.3173 7, 26
36. Goldreich, O.: Candidate one-way functions based on expander graphs. Electronic Colloquium on Computational Complexity (ECCC) 7(90) (2000) 2, 7
37. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1(2), 169–197 (1981), http://dx.doi.org/10.1007/BF02579273 31
38. Hada, S.: Zero-knowledge and code obfuscation. In: Okamoto, T. (ed.) Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1976, pp. 443–457. Springer (2000), https://doi.org/10.1007/3-540-44448-3_34 10
39. Hu, Y., Jia, H.: Cryptanalysis of GGH map. In: Advances in Cryptology - EUROCRYPT. Lecture Notes in Computer Science, vol. 9665, pp. 537–565. Springer (2016) 11
40. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Prabhakaran, M., Sahai, A.: Efficient non-interactive secure computation. In: Advances in Cryptology - EUROCRYPT. pp. 406–425 (2011) 2
41. Joux, A.: A one round protocol for tripartite diffie-hellman. In: Bosma, W. (ed.) Algorithmic Number Theory, 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000, Proceedings. Lecture Notes in Computer Science, vol. 1838, pp. 385–394. Springer (2000), http://dx.doi.org/10.1007/10722028_23 11
42. Klíma, O., Tesson, P., Thérien, D.: Dichotomies in the complexity of solving systems of equations over finite semigroups. Theory of Computing Systems 40(3), 263–297 (2007) 26
43. Kothari, P.K., Mori, R., O'Donnell, R., Witmer, D.: Sum of squares lower bounds for refuting any CSP. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC. pp. 132–145. ACM (2017) 6, 8, 9, 26
44. Lasserre, J.B.: New positive semidefinite relaxations for nonconvex quadratic programs. In: Advances in convex analysis and global optimization (Pythagorion, 2000), Nonconvex Optim. Appl., vol. 54, pp. 319–331. Kluwer Acad. Publ., Dordrecht (2001), http://dx.doi.org/10.1007/978-1-4613-0279-7_18 6, 31
45. Lin, H.: Indistinguishability obfuscation from constant-degree graded encoding schemes. In: Advances in Cryptology - EUROCRYPT. pp. 28–57 (2016) 2, 11

46. Lin, H.: Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In: Advances in Cryptology - CRYPTO. vol. 10401, pp. 599–629. Springer (2017) 2, 3, 11
47. Lin, H., Tessaro, S.: Indistinguishability obfuscation from bilinear maps and block-wise local prgs. IACR Cryptology ePrint Archive p. 250 (2017) 1, 2, 3, 5, 9, 10, 11, 12, 15, 23, 24, 25
48. Lin, H., Vaikuntanathan, V.: Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS. pp. 11–20. IEEE Computer Society (2016) 2, 11
49. Lombardi, A., Vaikuntanathan, V.: Limits on the locality of pseudorandom generators and applications to indistinguishability obfuscation. In: Theory of Cryptography - 15th International Conference, TCC. vol. 10677, pp. 119–137. Springer (2017) 1, 4, 5, 6, 9, 10, 16, 17
50. Lombardi, A., Vaikuntanathan, V.: Minimizing the complexity of Goldreich's pseudorandom generator. IACR Cryptology ePrint Archive p. 277 (2017) 3, 6
51. Miles, E., Sahai, A., Zhandry, M.: Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In: Advances in Cryptology - CRYPTO. vol. 9815, pp. 629–658. Springer (2016) 11
52. Mossel, E., Shpilka, A., Trevisan, L.: On epsilon-biased generators in NC^0. Random Struct. Algorithms 29(1), 56–81 (2006) 3, 6, 7, 11
53. O'Donnell, R.: Analysis of boolean functions. Cambridge University Press (2014) 18
54. O'Donnell, R., Witmer, D.: Goldreich's PRG: evidence for near-optimal polynomial stretch. In: IEEE 29th Conference on Computational Complexity—CCC 2014, pp. 1–12. IEEE Computer Soc., Los Alamitos, CA (2014), http://dx.doi.org/10.1109/CCC.2014.9 9
55. Parrilo, P.A.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, Citeseer (2000) 6, 31
56. Raghavendra, P., Rao, S., Schramm, T.: Strongly refuting random CSPs below the spectral threshold. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017. pp. 121–131. ACM (2017) 6, 8
57. Rozenman, E., Shalev, A., Wigderson, A.: Iterative construction of cayley expander graphs. Theory OF Computing 2(5), 91–120 (2006) 26
58. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Symposium on Theory of Computing, STOC. pp. 475–484. ACM (2014) 11
59. Shor, N.Z.: Quadratic optimization problems. Izv. Akad. Nauk SSSR Tekhn. Kibernet. (1), 128–139, 222 (1987) 6, 31
60. Witmer, D.: On refutation of random constraint satisfaction problems (thesis proposal) (2017), http://www.cs.cmu.edu/~dwitmer/papers/proposal.pdf 6, 8

A Analysis of the Basic SDP Program

The degree d SOS program [16] for a polynomial optimization problem of the form

$$\max_{x\in\{\pm1\}^n} p(x)$$

corresponds to

$$\max_{\mu} \tilde{\mathbb{E}}\, p$$

where $\tilde{\mathbb{E}}$ ranges over the set of degree d expectation operators that satisfy the constraints $\{x_i^2 = 1\}_{i=1}^n$. These are defined as follows:

Definition A.1 (Pseudo-expectation). *Let $\mathcal{P}_{n,d}$ denote the space of all degree $\leq d$ polynomials on n variables. A linear operator $\tilde{\mathbb{E}} : \mathcal{P}_{n,d}$ is a degree d* pseudo-expectation *if it satisfies the following conditions:*

1. $\tilde{\mathbb{E}}[1] = 1$.
2. $\tilde{\mathbb{E}}[p^2] \geq 0$ *for every polynomial p of degree at most $d/2$.*

A pseudo-expectation is said to satisfy a constraint $\{q = 0\}$ if for every polynomial p of degree at most $d - deg(q)$, $\tilde{\mathbb{E}}[pq = 0$. We say that $\tilde{\mathbb{E}}$ satisfies the constraint $\{q \geq 0\}$ if for every polynomial p of degree at most $d/2 - deg(q)/2$, $\tilde{\mathbb{E}}[p^2 q] \geq 0$.

If μ is any distribution on $\mathbb{R}^n$, then the associated expectation is a pseudo-expectation operator of all degrees. The above definition can be thought of as a relaxation of the notion of an actual expectation.

Key to the utility of the definition above is the following theorem that shows one can efficiently search over the space of all degree d pseudo-expectations.

Theorem A.2 ([59,55,44]). *For any n, and integer d, the following set has an $n^{O(d)}$ time weak separation oracle (in the sense of [37]):*

$$\{\tilde{\mathbb{E}}[(1, x_1, x_2, \ldots, x_n,)^{\otimes d}] \mid \tilde{\mathbb{E}} \ \textit{ is a degree } d \textit{ pseudo-expectation}\}$$

In this appendix we expand on how Charikar and Wirth's work [19] implies the the following theorem:

Theorem A.3. *For every degree two polynomial $p \colon \mathbb{R}^n \to \mathbb{R}$ with no constant term, the value of the degree two SOS program for*

$$\max_{x \in \{\pm 1\}^n} p(x) \tag{A.1}$$

is larger than the true value of (A.1) by a factor of at most $O(\log n)$.

Theorem A.3 is a direct implication of the following result of [19]:

Theorem A.4 (Symmetric Grothendieck Inequality, [19], Theorem 1). *Let A be any $m \times m$ matrix such that $A_{i,i} = 0$ for every i. Then,*

$$\max_{X \succeq 0, X_{i,i}=1 \forall i} Tr(AX) \leq O(\log n) \max_{x \in \{\pm 1\}^n} x^\top A x$$

Proof (of Theorem A.3 from Theorem A.4). Suppose that there is a degree 2 pseudo-distribution $\{x\}$ such that $\tilde{\mathbb{E}}\, p(x) \geq \theta$, and let X be the $n+1 \times n+1$ matrix corresponding to $\tilde{\mathbb{E}}(x,1)(x,1)^\top$. That is, $X_{i,j} = \tilde{\mathbb{E}}\, x_i x_j$ and $X_{n+1,i} = X_{i,n+1} = \tilde{\mathbb{E}}\, x_i$. Note that X is a psd matrix with 1's on the diagonal.

Then $\mathrm{Tr}(AX) \geq \theta$ if A be the $(n+1) \times (n+1)$ matrix that represents the polynomial p. In this case Theorem A.4 implies that there is an $n+1$ dimensional vector $(x,\sigma) \in \{\pm 1\}^{n+1}$ such that $(x,\sigma)^\top A(x,\sigma) \geq \Omega(\theta/\log n)$. If we write $p(x) = q(x) + l(x)$, where q is the homogeneous degree two and l is linear, then we can see by direct inspection that

$$(x,\sigma)^\top A(x,\sigma) = q(x) + \sigma l(x) = p(\sigma x)$$

with the last equality following from the fact that $q(-x) = q(x)$ and $l(-x) = -l(x)$. Hence the vector $\sigma x \in \{\pm 1\}^n$ demonstrates that the value of (A.1) is at least $\Omega(\theta/\log n)$.