2016 IEEE 57th Annual Symposium on Foundations of Computer Science

# Indistinguishability Obfuscation from DDH-like Assumptions on Constant-Degree Graded Encodings

Huijia Lin
University of California Santa Barbara
Email: rachel.lin@cs.ucsb.edu

Vinod Vaikuntanathan
Massachusetts Institute of Technology
Email: vinodv@csail.mit.edu

***Abstract*—All constructions of general purpose indistinguishability obfuscation (IO) rely on either meta-assumptions that encapsulate an exponential family of assumptions (e.g., Pass, Seth and Telang, CRYPTO 2014 and Lin, EUROCRYPT 2016), or polynomial families of assumptions on graded encoding schemes with a high polynomial degree/multilinearity (e.g., Gentry, Lewko, Sahai and Waters, FOCS 2014).**

**We present a new construction of IO, with a security reduction based on two assumptions: (a) a *DDH-like* assumption — called the *joint-SXDH assumption* — on *constant degree* graded encodings, and (b) the existence of polynomial-stretch pseudorandom generators (PRG) in $\mathsf{NC}^0$. Our assumption on graded encodings is simple, has constant size, and does not require handling composite-order rings. This narrows the gap between the mathematical objects that exist (bilinear maps, from elliptic curve groups) and ones that suffice to construct general purpose indistinguishability obfuscation.**

***Index Terms*—Cryptography; Program Obfuscation; Graded Encodings.**

## I. Introduction

Indistinguishability obfuscation (IO) is a probabilistic polynomial-time algorithm $\mathcal{O}$ that takes as input a circuit $C$ and outputs an (obfuscated) circuit $C' = \mathcal{O}(C)$ satisfying two properties:

(a) *functionality*: $C$ and $C'$ compute the same function; and
(b) *security*: for any two circuits $C_1$ and $C_2$ that compute the same function (and have the same size), $\mathcal{O}(C_1)$ and $\mathcal{O}(C_2)$ are computationally indistinguishable.

IO is a surprisingly powerful cryptographic notion. Defined first in the seminal work of Barak, Goldreich, Impagliazzo, Rudich, Sahai, Vadhan and Yang [15], it was largely unnoticed (with the singular exception of [37]) until the recent work of Garg, Gentry, Halevi, Raykova, Sahai and Waters [31] who demonstrated a *candidate* construction of indistinguishability obfuscation, and the work of Sahai and Waters [50] who showed that despite appearing somewhat useless to an untrained eye, IO has enormous power, so much so that it is virtually "crypto-complete". Starting from [50], we now know that IO gives us a treasure-chest of cryptographic constructions, solutions to a number of open problems (see, e.g., [50], [31], [30], [25], [17] and many more) and even has implications in complexity theory [18], [39].

In the tradition of theoretical cryptography, one *defines* a useful cryptographic object (a definition involves a notion of functionality and one of security), demonstrates a *construction* of this object using mathematics, and finally, *proves its security* from computational hardness assumptions (such as the hardness of factoring, discrete logarithms, or learning with errors). Garg et al. [31] showed a construction of IO using ideal lattices (which they abstracted into a framework called cryptographic multilinear maps [29]) but the construction came *as-is* with *no* security proof. There have since been a series of attempts at security proofs for IO under assumptions of varying complexity (which we will review in detail in the sequel). This state of affairs motivates one of the most important questions in cryptography today:

*Does indistinguishability obfuscation exist, and under which cryptographic assumptions?*

Let us cut to the chase: in this work, we show a construction of IO from the *joint SXDH* assumption on prime-order multilinear maps with *constant multilinearity*. This narrows the gap between the mathematical objects that exist (bilinear maps) and ones that suffice for IO (O(1)-linear maps). We now describe what each of these terms mean through the lens of existing IO constructions.

**Constructions and Proofs of Indistinguishability Obfuscation** Over the last three years, there has been a great deal of work trying to construct IO schemes and prove their security. The mathematics underlying all IO constructions, broadly speaking, arises from the geometry of numbers (or the theory of integer lattices), but this has been abstracted out into the framework of *graded encoding schemes* (also called *cryptographic*


DOI 10.1109/FOCS.2016.11

*multilinear maps*) [21], [29].[1]

In a nutshell, a graded encoding scheme for a ring $\mathcal{R}$ provides us with a (potentially exponentially large) collection of groups (written multiplicatively) of order $q$ together with a relation pairable on them such that for any two groups $G_i$ and $G_j$ such that $\mathsf{pairable}(G_i, G_j) = G_k$, we have $g_i^{\alpha} \otimes g_j^{\beta} = g_k^{\alpha\beta}$ where $\otimes$ is a pairing function and $\alpha, \beta \in \mathbb{Z}_q$ are scalars. If $\mathsf{pairable}(G_i, G_j) = \perp$, then we say that $G_i$ and $G_j$ are *not pairable*, and otherwise they are *pairable*. (In the literature, such graded encoding schemes are referred to as *clean* graded encodings, and can be generalized to *noisy* graded encodings. For most part of this introduction, we will use the interface of clean graded encodings; see Section I-C for a discussion on noisy graded encodings towards the end of the introduction.)

We can use these groups to compute multivariate polynomials in the exponent. That is, given a sequence of elements $g_j^{\alpha_j}$, compute $g_k^{p(\alpha_1,\ldots,\alpha_n)}$ where $p$ is an $n$-variate polynomial and $g_k$ is an element in the appropriate group, provided that $p$ can be computed using a sequence of group operations in the same group and pairing operations over different groups as specified by pairable. The maximum degree of a multivariate polynomial that can be computed in the exponent is called the *multilinearity* of the collection. We call the number of groups in the collection the *universe size* (which could be constant, polynomial or exponential in the security parameter). The order of the group is $q$, and we will differentiate between prime-order and composite-order groups. In this language, the well-known bilinear maps have multilinearity 2 and universe size 2 (or 3 in the case of asymmetric pairing groups).

- **Proofs in Ideal Models:** Several works [31], [24], [22], [14], [9], [51] showed proofs of security for obfuscation in the so-called *ideal multilinear group* model. Roughly speaking, these models postulate that the only way an adversary can operate on group elements is through the legal group interface (namely, group operations between two elements in the same group and the pairing operation between two elements in pairable groups). Restricting the power of the adversary in such a way is unrealistic, and is underscored by the fact that in this model, one can actually get virtual black-box obfuscation (which by [15] does not exist for general programs).
- **Concrete Assumptions on Graded Encodings (GES):** Pass, Seth and Telang [48] postulated an *uber-assumption* on multilinear maps which, roughly speaking, say that any attack on a collection of group elements can be translated into an attack in the *ideal multilinear group model*. Gentry, Lewko, Sahai and Waters [36], following the work of Gentry, Lewko and Waters [35], took the first step in simplifying the assumption and came up with a construction under the *multilinear subgroup elimination* assumption on *composite-order groups*. Bitansky and Vaikuntanathan [19] and Ananth and Jain [5], showed how to convert any functional encryption scheme into an IO scheme. Together with the FE construction of Garg, Gentry, Halevi and Zhandry [32], this gives us an IO scheme based on similar assumptions on composite-order groups. The main deficiency of all these constructions is that they require graded encoding schemes with large multilinearity – either $\mathrm{poly}(|C|, n, \lambda)$ standalone, or at least $\mathsf{poly}(n, \lambda)$ after applying the bootstrapping theorem of Canetti, Lin, Tessaro and Vaikuntanathan [23].[2] In addition, they all either rely on a very complicated uber-assumption, or rely on composite-order graded encodings. Furthermore, the universe size in all these cases is at least $\mathsf{poly}(n, \lambda)$.
- **Towards Constant Multilinearity:** The closest in spirit to this work is the recent result of Lin [42] who showed that constant-degree multilinear maps suffice for IO. (This is in spite of recent implausibility results [49], [43], [19], [44] showing that construction of IO in the *ideal constant-degree multilinear map* model, implies construction of IO in the plain model; in other words, constant-degree multilinear map does not "help" black-box construction of IO. [42] circumvents this by making non-black-box use of the graded encodings.) Unfortunately, her work has the following drawbacks: First, the concrete complexity assumption was a complicated über-assumption (borrowed from [48]); and secondly, her graded encoding collection has composite order and large, polynomial, universe size, namely $\mathsf{poly}(n, \lambda)$.

See Figure I for a summary.

In short, all assumptions used in the construction of IO are either a complicated uber-assumption that encodes the computation in the assumption itself, or an assumption on composite order graded encodings (GES) with polynomial universe size. The gap between bilinear maps and these objects is rather large. Even the construction of Lin [42] requires a collection of $\mathsf{poly}(n, \lambda)$ groups with a complex interaction between

[1] In reality, we do not have any instantiations of graded encoding schemes, but rather only *noisy* graded encodings which turn out to suffice for functionality.

[2] There are other IO bootstrapping theorems in the literature, notably that of [31]. However, they do not appear to help in reducing the multilinearity.

| | Assumption | Multilinearity | Universe | Composite Order? |
|---|---|---|---|---|
| [22]<br>[14], [7]<br>[9], [51] | ideal multilinear model | $\mathsf{poly}(n,\lambda)^\ddagger$ | $\mathsf{poly}(n,\lambda)^\ddagger$ | no |
| [48] | über-multilinear | $\mathsf{poly}(n,\lambda)^\ddagger$ | $\mathsf{poly}(n,\lambda)^\ddagger$ | no |
| [36] | multilinear subgroup elimination | $\mathsf{poly}(n,\lambda)^\ddagger$ | $\mathsf{poly}(n,\lambda)^\ddagger$ | yes |
| [19], [5], [6] +[32] | similar to multilinear subgroup elimination | $\mathsf{poly}(n,\lambda)$ | $\mathsf{poly}(n,\lambda)$ | yes |
| [42] | über-multilinear | $O(1)$ | $\mathsf{poly}(\lambda)$ | yes |
| This Work | Joint SXDH | $O(1)$ | $O(1)$ | no |

Fig. 1. A Summary of Known IO Constructions. $\ddagger$ denotes the fact that the complexity (multilinearity or universe size) was originally $\mathsf{poly}(|C|,n,\lambda)$ but can be brought down to $\mathsf{poly}(n,\lambda)$ using bootstrapping.

them (through the uber-assumption), even though the multilinearity is constant.

With the aim of narrowing the gap between the mathematical objects that exist (bilinear maps) and ones that suffice for IO, we seek to:

> *Construct IO from a simple assumption on prime-order GES with $O(1)$ multilinearity and universe size.*

### A. Our Results

**Joint-SXDH Assumption on Graded Encodings.** Our joint-SXDH assumption on graded encodings is a natural generalization of the standard symmetric external Diffie-Hellman (SXDH) assumption on (asymmetric) bilinear pairing groups. In short, SXDH states that the decisional Diffie-Hellman assumption holds in every source group. That is, let $G_0$ and $G_1$ be a pair of source groups, whose elements can be paired to produce elements in a target group $G_T$.

SXDH over bilinear maps:

$$\forall l \in \{0,1\},\ \{g_0,\ g_1,\ g_T,\ g_l^a,\ g_l^b,\ g_l^{ab}\} \approx \{g_0,\ g_1,\ g_T,\ g_l^a,\ g_l^b,\ g_l^r\}$$

where $g_i$ is the generator for group $G_i$ and $a, b, r$ are random exponents. Note that SXDH possibly holds in asymmetric bilinear groups because elements in the same source group do not pair; otherwise, one can easily distinguish the above distributions by checking the equality $e(g_l^a, g_l^b) = e(g_l^{ab}, g_l)$.

The SXDH assumption naturally generalizes to graded encodings with a collection of groups $\{G_l\}_l$: It postulates that the distribution of $g_l^a, g_l^b, g_l^{ab}$ in any group $l$ should be indistinguishable to that of $g_l^a, g_l^b, g_l^r$, provided that elements in $G_l$ cannot be paired with themselves. Here, because graded encodings allow for a richer computation structure, it is not only necessary that elements cannot be paired directly (*i.e.* $\mathbf{pairable}(G_l, G_l) = \perp$), but they also do not pair "indirectly", via a sequence of pairings with elements in other groups. This leads to the notion of the closure of the $\mathbf{pairable}$ function, denoted as $\mathbf{pairable}^*$, which roughly speaking indicates whether two groups $G_{l_1}, G_{l_2}$ can ever be paired via any sequence of pairing. More precisely, $\mathbf{pairable}^*(G_{l_1}, G_{l_2}) = 1$ if there are groups $G_{l_3}$ and $G_{l_4}$ such that $\mathbf{pairable}^*(G_{l_1}, G_{l_3}) = 1$, $\mathbf{pairable}^*(G_{l_2}, G_{l_4}) = 1$, and $\mathbf{pairable}(G_{l_3}, G_{l_4}) \neq \perp$; otherwise, $\mathbf{pairable}^*(G_{l_1}, G_{l_2}) = 0$. Then,

SXDH over Graded Encodings:

$$\forall l \text{ s.t. } \mathbf{pairable}^*(G_l, G_l) = 0,$$
$$\{\{g_i\},\ g_l^a,\ g_l^b,\ g_l^{ab}\} \approx \{\{g_i\},\ g_l^a,\ g_l^b,\ g_l^r\}$$

where $\{g_i\}$ is the set of generators of all groups.

Finally, joint-SXDH further generalizes SXDH. It considers the joint distribution of elements $(g_l^a, g_l^b, g_l^{ab})_{l\in S}$ in a set $S$ of groups, with the *same* exponents $a, b, ab$. By the same argument above, if any two groups $G_{l_1}, G_{l_2}$ in the set are pairable, directly or indirectly, one can distinguish the joint distribution from the distribution of $(g_l^a, g_l^b, g_l^r)_{l\in S}$ with random exponents $a, b, r$. Otherwise, in the same spirit as SXDH, the distributions are possibly indistinguishable — this is exactly our joint-SXDH assumption.

Joint-SXDH over Graded Encodings: For all Set $S$ that satisfies $\forall l_1, l_2 \in S$, $\mathbf{pairable}^*(G_{l_1}, G_{l_2}) = 0$:

$$\left\{\{g_i\},\ \{g_l^a,\ g_l^b,\ g_l^{ab}\}_{l\in S}\right\} \approx \left\{\{g_i\},\ \{g_l^a,\ g_l^b,\ g_l^r\}_{l\in S}\right\}$$

Furthermore, the subexponential joint-SXDH assumption requires the above distributions to have subexponentially small distinguishing gap to all polynomial time distinguishers.

**IO from joint-SXDH on Constant-Degree Graded Encodings.** We are now ready to state our main theorem.

**Theorem 1** (Main Theorem, Informal). *Assume the existence of a sub-exponentially secure $n^{1+\alpha}$-stretch pseudorandom generator (PRG) in $\mathsf{NC}^0$ for any positive constant $\alpha > 0$. Then, IO for $\mathsf{P}/\mathsf{poly}$ is implied by the sub-exponential joint-SXDH assumption on a constant-degree graded encoding scheme, with prime order and constant universe size.*

<u>*Tree-GES:*</u> The graded encoding scheme that our main theorem relies on has a specific pairable function that allows computing arithmatic circuits of layers of additions and multiplications; we refer to such a scheme a *tree-structured* graded encoding scheme, or *tree-GES* for short. Roughly speaking, a tree-GES consists of a set of groups arranged at the nodes of a 4-ary tree, together with a pairable function defined in the following way. If $G_{l_0}, G_{l_1}, G_{l_2}$ and $G_{l_3}$ are the (groups in the) four children of a node $G_l$, then $\mathbf{pairable}(G_{l_0}, G_{l_1}) = \mathbf{pairable}(G_{l_2}, G_{l_3}) = G_l$, whereas all other combinations are not pairable (*e.g.*, $\mathbf{pairable}(G_{l_0}, G_{l_2}) = \perp$, and so on). Naturally, given $g_{l_i}^{a_i}$ for $i \in \{0, 1, 2, 3\}$, one can compute $g_l^{a_0 a_1 + a_2 a_3}$ (one layer of multiplications followed by additions), and cannot compute (via the honest interface) any quadratic polynomial containing monomials $a_i a_j$ for $i = j$ or $i \leq 1 < j$.

One of the nice features of this general interface is that it captures directly the computation structure we need from GES, and it can be instantiated from both set-based and graph-based (prime- as well as composite-order) multilinear maps, which gives it a great deal of flexibility. For example, while none of the previous abstract constructions [48], [36], [32], [42] can be instantiated based on the graph-based multilinear maps of [34], our IO scheme will admit such an instantiation.

In the language of tree-GES, our IO construction relies on the joint-SXDH assumption on tree-GES with a tree of constant depth. Carrying this over to the set-multilinear map setting, this translates to constant multilinearity and constant universe size. Our main theorem also relies on a sub-exponentially secure polynomial stretch PRG. See Section I-D for a discussion on this assumption.

**Our Approach via Bootstrapping FE for $\mathsf{NC}^0$ to IO:** Our approach towards constructing IO from constant-depth tree-GES is through a bootstrapping step showing that assuming the existence of polynomial-stretch pseudorandom generators (PPRG) in $\mathsf{NC}^0$, a (collusion-resistant) Functional Encryption (FE) scheme for $\mathsf{NC}^0$ implies indistinguishability obfuscation for all of $\mathsf{P}$; the FE scheme for $\mathsf{NC}^0$ needs to have linear efficiency, in the sense that, encryption time depends linearly in the message length. Then, we use constant-depth tree-GES to implement a such FE scheme.

We invite the reader to pause for a moment and note that while common sense would dictate that obfuscation is more powerful than functional encryption, obfuscation for $\mathsf{NC}^0$ circuits is completely trivial (namely, for each output bit of the circuit, publish the truth table of the circuit that generates it, and the constant number of input bits that the output bit depends on) and yet, FE for $\mathsf{NC}^0$ is far from trivial. Indeed, the theorem below says that FE for $\mathsf{NC}^0$ is powerful enough to imply indistinguishability obfuscation for all of $\mathsf{P}$.

**Theorem 2** (Informal, following [19], [6], [42]). *Assume the existence of a sub-exponentially secure $n^{1+\alpha}$-*stretch *pseudorandom generator in $\mathsf{NC}^0$ for any positive constant $\alpha > 0$. Then, IO for $\mathsf{P}/\mathsf{poly}$ is implied by FE schemes for $\mathsf{NC}^0$ with encryption time linear in the input length.*

The theorem follows from combining observations in previous works [19], [6], [42], in particular, combining randomized encoding in $\mathsf{NC}^0$ and PRG in $\mathsf{NC}^0$ to transform any $\mathsf{NC}^1$ computation into an $\mathsf{NC}^0$ computation.

**Constructing FE for $\mathsf{NC}^0$ from Constant-Degree GES:** Our main technical contribution is the construction of an FE scheme for $\mathsf{NC}^0$ with linear efficiency under the joint-SXDH assumption on tree-GES for constant-depth trees.

**Theorem 3** (Informal). *Assuming the existence of a prime-order tree-GES for depth-$O(1)$ trees with the joint-SXDH assumption, there is a (collusion-resistant) FE scheme for all $\mathsf{NC}^0$ circuits, with encryption time linear in message length.*

Thus, put together, we get IO for $\mathsf{P}/\mathsf{poly}$, assuming joint-SXDH and the existence of polynomial-stretch PRGs in $\mathsf{NC}^0$. We now proceed to describe the techniques behind the FE construction.

### *B. Technical Overview*

Since Theorem 2 follows from observations in previous works, we focus on the question:

> *How does one construct a collusion-resistant functional encryption scheme for $\mathsf{NC}^0$, with linear efficiency?*

**The State-of-the-Art of Collusion Resistant FE.** In the literature, the only constructions of collusion-resistant FE from standard assumptions are for computing inner products, referred to as Inner Product Encryption (IPE). Roughly speaking, a (public key or secret key) IPE scheme allows to encode vectors $\mathbf{y}$ and $\mathbf{x}$ in a ring $\mathcal{R}$, in a function key $\mathsf{sk}_{\mathbf{y}}$ and ciphertext $\mathsf{ct}_{\mathbf{x}}$ respectively, and decryption computes the inner product $\langle \mathbf{y}, \mathbf{x} \rangle \in \mathcal{R}$. Abdalla, Bourse, De Caro and Pointcheval (ABCP) [2], [1] came up with a public key IPE scheme based on one of a variety of assumptions, such as the decisional Diffie-Hellman assumption, the Paillier assumption and the learning with errors assumption. Following that, Bishop, Jain and Kowalczyk [16] (BJK) constructed a secret-key scheme based on the SXDH assumption over asymmetric bilinear maps; their scheme achieves the stronger security property of *weak function-hiding* (explained below). Both the ABCP and BJK schemes do not compute the inner product $\langle \mathbf{y}, \mathbf{x} \rangle$ in the clear, but computes it in the exponent $g^{\langle \mathbf{y}, \mathbf{x} \rangle}$; the BJK scheme in fact computes the inner product $\theta \langle \mathbf{y}, \mathbf{x} \rangle$ masked by a scalar $\theta$ in the exponent; see more discussion later.[3]

Given IPE schemes, it is trivial to implement FE for quadratic polynomials: Simply write a quadratic function $f$ as a linear function over quadratic monomials $f(x) = \Sigma_{i,j} c_{i,j} x_i x_j = \langle \mathbf{c}, \mathbf{x} \otimes \mathbf{x} \rangle$, where $\otimes$ is tensor product. Then, use an IPE scheme to generate a ciphertext $\mathsf{ct}_{\mathbf{x} \otimes \mathbf{x}}$ and a function key $\mathsf{sk}_{\mathbf{c}}$, which produce $f(x)$. However, the ciphertext size scales *quadratically* in $n = |\mathbf{x}|$. This idea easily generalizes and gives a FE scheme for $\mathsf{NC}^0$ with encryption time $n^d$, where $d$ is the degree of the computation. Unfortunately, improving these FE schemes to have encryption time linear in the input length under standard assumptions (*e.g.* bilinear maps) has proved elusive.

Coming from the "other side", Garg, Gentry, Halevi and Zhandry (GGHZ) [32] proposed a general-purpose FE scheme from polynomial-degree GES (with composite-order). A natural next attempt would be to try to specialize their FE scheme to $\mathsf{NC}^0$ circuits, in the hope that we can pull off the construction using only constant-degree GES. This wishful thinking runs into trouble. Very roughly speaking, the GGHZ construction works with a universal branching program and requires GES with multilinearity that is $O(\ell)$ where $\ell$ is the length of the branching program. Now, even if we only want to handle $\mathsf{NC}^0$ circuits that take $n$ bits of input, converting them into a universal branching program results in a program of size $\Omega(n)$.

One might hope to get around this problem by representing the $\mathsf{NC}^0$ circuit for each output bit as a constant-sized branching program; however, in this case, it is not clear how each function key can "index" the right input bits in the $n$-bit input to compute on. This "indexing problem" prevents us from tweaking the construction to support $\mathsf{NC}^0$ circuits with constant multilinearity.

In this work, we come up with a completely different FE construction that not only gives us constant multilinearity, but also relies on GES with prime order and constant universe size, and the simple joint-SXDH assumption.

**Overview** Towards constructing FE for $\mathsf{NC}^0$ with linear efficiency from constant-degree GES, our first observation is that functionality is easy to achieve, since $\mathsf{NC}^0$ circuits $f$ can be represented as constant-degree arithmetic circuits or polynomials, and constant-degree GES supports evaluating constant-degree polynomials in the exponent. Once the output $y = f(x)$ is computed in the exponent $g^y$, it can be extracted as it is Boolean. Thus, the main challenge lies in achieving security, ensuring that the input and all intermediate computation results are hidden.

To hide the input and computation, the first tool that comes in mind is Randomized Encodings (RE). An RE scheme allows one to use randomness to encode a function $f$ and an input $x$, $\Pi \xleftarrow{\$} \mathbf{RE}(f, x; r)$, so that:

1) The encoding algorithm is simple: Each element of $\Pi$ is of the form $x_{\pi(i)} \cdot p_i(r) + q_i(r)$, where $\pi$ is an input-mapping function, and $p_i, q_i$ are polynomial functions of the randomness $r$. That is, a linear function of a single input bit (and a polynomial function of the randomness $r$);
2) The encoding $\Pi$ reveals the output $z = f(x)$ of the computation and nothing more.

The key difference of RE from FE in that RE cannot be reused, whereas the ciphertexts (respectively, function keys) of a FE scheme can be reused across an unbounded number of function keys (respectively, ciphertexts).

*The First Idea and Challenges.* Our first and foremost idea is *to combine the re-usability of IPE schemes with the capability of hiding inputs and computations of RE schemes*, by designing techniques to use an IPE scheme to compute randomized encodings. More specifically,

**Outline of Our FE scheme**

- *Key Generation:* To create a key $\mathsf{sk}_f$ for $f \in \mathsf{NC}^0$, first encode $f$ in a set of vectors $\{\mathbf{u}_k\}$, and then publish IPE function keys $\mathsf{sk}_f = \{\mathsf{sk}^k_{\mathbf{u}_k}\}$ for these vectors, using independently sampled master keys.
- *Encryption:* Similarly, to encrypt an input $x \in \{0,1\}^n$, encode $x$ in a set of vectors $\{\mathbf{v}_k\}$, and encrypt them in IPE ciphertexts $\mathsf{ct}_x = \{\mathsf{ct}^k_{\mathbf{v}_k}\}$ with corresponding master keys.

The vectors $\mathbf{u}_k$ and $\mathbf{v}_k$ are set up in a way so that their inner products $\langle \mathbf{u}_k, \mathbf{v}_k \rangle = \Pi_k$ produce exactly the $k^{\text{th}}$

[3] There has been a long line of work on "inner product *testing* functional encryption" or "zero-testing IPE" (see, e.g., [40], [41] and many others) which is different from what we need here. In IPE, we require that function key $\mathsf{sk}_{\mathbf{y}}$ and ciphertext $\mathsf{ct}_{\mathbf{x}}$ produce the inner product in $\mathcal{R}$ in the exponent. In contrast, in zero-testing IPE, one can only compute whether $\langle \mathbf{x}, \mathbf{y} \rangle \stackrel{?}{=} 0$ in $\mathcal{R}$. In particular, they do not produce the inner product in the exponent, in a way that allows for further computation. Hence, they are insufficient for our construction of FE for $\mathsf{NC}^0$.

element $\Pi_k$ in the randomized encoding for $f, x$. Thus, the IPE scheme ensures that evaluating $\mathsf{sk}^k_{\mathbf{u}_k}$ and $\mathsf{ct}^k_{\mathbf{v}_k}$ produces $\Pi_k$ in the exponent $g_{l_k}^{\Pi_k}$ in some group $G_{l_k}$.

In the literature, the idea of using FE for a weak function class, to compute the randomized encodings of a stronger function class has been used in bootstrapping FE for $\mathsf{NC}^1$ to FE for $\mathsf{P/poly}$ [4]. In some sense, our construction can be viewed as bootstrapping FE for inner products to FE for $\mathsf{NC}^0$. Here, unique challenges arise due to the fact that we can only compute inner products.

- **Challenge 1:** *How to generate randomized encodings using only inner products?*
  To do so, we crucially rely on *affine* randomized encodings, where each element $\Pi_k$ in the encoding of a computation $f, x$ depends *linearly* on each bit in $x$. The idea is then to represent each element $\Pi_k$ as the inner product between some coefficient vectors (depending on $f$) and input vectors (depending on $x$), so that, $\Pi_k$ can be computed using IPE.
  In particular, we will use the arithmetic randomized encodings for $\mathsf{NC}^1$ of Applebaum, Ishai and Kushilevitz [12], which is affine and has many other useful properties.
- **Challenge 2:** *How to generate the randomness for randomized encodings?*
  Consider a scenario where our FE for $\mathsf{NC}^0$ scheme is used to publish $m$ function keys $\{\mathsf{sk}_{f_j}\}$ and $m$ ciphertexts $\{\mathsf{ct}_{x_i}\}$. Every pair of key and ciphertext $\mathsf{sk}_{f_j}, \mathsf{ct}_{x_i}$ computes a randomized encoding $\Pi_{j,i} \in \mathbf{RE}(f_j, x_i)$ (in the exponent), which requires using fresh (at least, "computationally fresh") randomness $r_{ji}$. Note that we need in total $m^2$ "pieces" of randomness, but has only $m$ function keys and ciphertexts — $r_{ji}$'s can only be pseudorandom.
  In the case of bootstrapping FE for $\mathsf{NC}^1$ to FE for $\mathsf{P/poly}$, this problem is easily resolved using Pseudo Random Functions (PRFs): One can simply encrypt a PRF seed $s$ together with the input $x$, and the function keys evaluate the PRF on $s$ to expand pseudorandomness for computing the randomized encoding. However, in our case, the functionality of IPE does not support PRF evaluation. Not even extremely strong local PRGs can help here, since any quadratic-stretch PRGs (from $O(m)$ bits to $m^2$ bits) has at least degree 3.
  We resolve this problem by, instead, relying on built-in pseudorandomness assumption, namely *joint-SXDH*, in GES. Indeed, the SXDH assumption w.r.t. a group $G_l$ guarantees that given a set of $2m$ random elements in the exponent $\{g_l^{s_j}, g_l^{t_i}\}_{j,l\in[m]}$, the set of $m^2$ products in the exponent $\{g_l^{s_j t_i}\}$ are indistinguishable to elements $\{g_l^{r_{ji}}\}$ with truly random exponents. The $r_{ji}$'s in the exponent will be the randomness for generating RE. They can be computed from short, length-$2m$, seeds $\{s_j, t_i\}$ in degree 2; just that they must reside in the exponent.

Before going into details on how to resolve the above two challenges, we first complete the construction outline.

*Achieving Functionality.* Given that we can use IPE to compute randomized encodings in the exponent $\{g_{l_k}^{\Pi_k}\}$, it is tempting to think that one can simply extract $\Pi$ if the encodings are binary, and compute the output $y$ in the clear. If this could be done, we would have obtained FE for $\mathsf{NC}^0$ from only bilinear maps, and thus IO from bilinear maps. The catch is that we have to use arithmetic randomized encodings (where the elements that compose the randomized encoding, namely $\Pi_k$, live in a large field) and cannot use binary randomized encodings. Roughly speaking, the culprit is our solution to Challenge 2. As mentioned above, the elements of the randomized encodings are generated pseudo-randomly. The randomness used for generating the randomized encoding lives in the exponent ring $\mathcal{R}$, and can only produce pseudorandomness *in the exponent* through the joint-SXDH assumption. In turn, as a result of this, we need to use *arithmetic* randomized encodings (in particular, [12]), which cannot be extracted from the exponent, unless discrete logarithm is easy. In fact, extracting these arithmetic randomized encodings would lead to attacks on the joint-SXDH assumption.

Therefore, we rely on constant-degree GES to achieve *functionality*, by evaluating the arithmetic randomized encoding $\Pi$ in the exponent. Evaluation produces the output $y$ in the exponent, which can be extracted since it is binary. More specifically, recall that we use a tree-structured GES that supports evaluating arithmetic circuits with a constant number of layers of multiplications and additions, in particular, the RE evaluation circuit for $\mathsf{NC}^0$ computation. We will carefully instantiate different IPE instances $(\mathsf{sk}^k_{\mathbf{u}_k}, \mathsf{ct}^k_{\mathbf{v}_k})$ using different groups in the tree-GES so that IPE evaluation produces the randomized encoding $g_{l_k}^{\Pi_k}$ in appropriate groups $l_k$, on which RE evaluation can be performed.

With these insights, let's now circle back and resolve challenges 1 and 2.

**Resolving Challenge 1.** Our key tool is the *affine* AIK arithmetic randomized encodings (ARE) [12], which depends linearly in the input. More specifically, the AIK arithmetic randomized encoding for an (arithmetic) $\mathsf{NC}^1$ function $f$ and input $\mathbf{x} \in \mathcal{R}^n$ is computed using a set of $m = \mathrm{poly}(n)$ *fixed* linear functions $L_k$ as follows:

$$\left\{\Pi_k = L_k(\mathbf{x}, \mathbf{r}) = p_k(\mathbf{r}) x_{\pi(k)} + q_k(\mathbf{r});\ \pi : [m] \to [n]\right\}$$

Here, each randomized encoding element $\Pi_k$ depends on a single input bit $x_{\pi(k)}$, determined by an input mapping function $\pi$. The coefficients of the linear functions $p_k(\mathbf{r})$ and $q_k(\mathbf{r})$ are *fixed* multi-linear polynomials that act on the randomness $\mathbf{r}$. The only part that depends on the function $f$ is the input mapping function.

To use IPE to compute such arithmetic randomized encodings, the idea is that the FE key generation algorithm encodes the coefficients $p_k(\mathbf{r}), q_k(\mathbf{r})$ and the input mapping function $\pi$, and the FE encryptor encrypts $\mathbf{x}$; they together compute the affine functions $L_k(\mathbf{x}, \mathbf{r})$. More precisely,

**Our FE scheme, version 1**

- *Key Generation:* To generate a key $\mathsf{sk}_f$ for $f$, sample randomness $\mathbf{r}$, and publish IPE keys $\mathsf{sk}_f = \{\mathsf{sk}^k_{\mathbf{u}_k}\}$ for vectors $\mathbf{u}_k = (p_k(\mathbf{r})||q_k(\mathbf{r})) \otimes \mathbf{e}_{\pi(k)}$ (using independently sampled master keys).
- *Encryption:* To encrypt $x$, publish IPE ciphertexts $\mathsf{ct}_x = \{\mathsf{ct}^k_{\mathbf{v}_k}\}$ for vectors $\mathbf{v}_k = (x_i||1)_{i\in[n]}$ (using corresponding master keys).

It is easy to verify that $\langle \mathbf{u}_k, \mathbf{v}_k \rangle = \Pi_k$. In other words, we achieve the goal of computing AIK arithmetic randomized encodings using IPE. The above scheme is, however, insecure: In particular, the randomness $\mathbf{r}$ for generating randomized encodings is hardcoded in the secret key, meaning that the randomized encodings for the same function $f$ and different inputs $x_1, x_2, \cdots$ share the same randomness, which renders them insecure. This leads us back to resolving the second challenge of generating the randomness for randomized encodings.

**Resolving Challenge 2.** We rely on joint-SXDH to generate randomness. What we need is that for every pair of key and ciphertext, the randomized encoding should use fresh (at least, "computationally fresh") randomness. We accomplish this by (re-)writing the affine functions as

$$\left\{ \Pi_k = L_k(\mathbf{x}, \mathbf{r}, \mathbf{s}) = p_k(\mathbf{rs})x_{\pi(k)} + q_k(\mathbf{rs}) \right\}$$

The randomness in use is the coordinate-wise multiplication of $\mathbf{r}$ and $\mathbf{s}$. We will put one multiplicative "share" $\mathbf{r}$ in the key, and the other $\mathbf{s}$ in the ciphertext. To see how to compute such a thing, note that if

$$p_k(\mathbf{r}) = \sum_j M_{kj}(\mathbf{r}) \text{ and } q_k(\mathbf{r}) = \sum_j M'_{kj}(\mathbf{r})$$

where the $M_{kj}$ and $M'_{kj}$ are monomials, then

$$p_k(\mathbf{rs}) = \sum_j M_{kj}(\mathbf{r})M_{kj}(\mathbf{s})$$

and similarly for $q_k$. We modify our FE scheme as below:

**Our FE scheme, version 2**

- *Key Generation:* To generate $\mathsf{sk}_f$ for $f$, sample $\mathbf{r}$ and publish IPE keys $\mathsf{sk}_f = \{\mathsf{sk}^k_{\mathbf{u}_k}\}$ for vectors
$$\mathbf{u}_k = \left(M_{kj}(\mathbf{r}), M'_{kj}(\mathbf{r})\right)_j \otimes \mathbf{e}_{\pi(k)} \qquad (1)$$
- *Encryption:* To encrypt $x \in \{0,1\}^n$, sample $\mathbf{s}$ and publish IPE ciphertexts $\mathsf{ct}_x = \{\mathsf{ct}^k_{\mathbf{v}_k}\}$ for vectors
$$\mathbf{v}_k = \left(M_{kj}(\mathbf{s})x_i, M'_{kj}(\mathbf{s})\right)_{i,j} \qquad (2)$$

Now, the inner product $\langle \mathbf{u}_k, \mathbf{v}_k \rangle$ is the randomized encoding element $\Pi_k$ generated using randomness $\mathbf{rs}$. Moreover, the AIK randomized encoding has the property that the total number of monomials $M_{k,j}, M'_{k,j}$ is bounded by $2^{O(d)}$, where $d$ is the depth of the arithmetic circuit computing $f$. Thus for $\mathsf{NC}^0$ computations, the vectors $\mathbf{u}_k, \mathbf{v}_k$ are of length $O(n)$, linear in the input length, giving us the desired linear efficiency property.

**Overview of Security Proof** FE security states that the ciphertexts $\mathsf{ct}_{x^0}$ and $\mathsf{ct}_{x^1}$ of inputs $x^0$ and $x^1$ should be indistinguishable, even in the presence of keys $\{\mathsf{sk}_{f_j}\}$ as long as they satisfy that $f_j(x^0) = f_j(x^1)$ for every $j$. We want to reduce this indistinguishability to the security of randomized encodings — that encodings $\{\Pi^0_j\}$ for $f_j, x^0$, and encodings $\{\Pi^1_j\}$ for $f_j, x^1$ are indistinguishable. But, before invoking RE security, we must first argue that the input $x^b$ is hidden, and the randomness $\{\mathbf{r}_j\mathbf{s}\}$ for generating $\Pi^b_j$ is jointly pseudorandom. This is certainly not the case w.r.t. honestly generated keys and ciphertexts: First $x^b$ is embeded in the ciphertext, and second it seems impossible to argue that the products $\{\mathbf{r}_j\mathbf{s}\}$ are pseudorandom, when $\mathbf{r}_j$ and $\mathbf{s}$ reside respectively in IPE keys and ciphertexts that can be paired together.

To resolve this conundrum, our idea is leveraging the function hiding property of a secret-key IPE scheme, in order to "move" the input $x^b$ and $\mathbf{s}$ into the function keys in security hybrids. Let us explain. The function hiding property guarantees that IPE keys and ciphertexts for two sets of vectors $\{\mathbf{a}_i, \mathbf{b}_i\}$ and $\{\mathbf{a}'_i, \mathbf{b}'_i\}$ are indistinguishable if they produce identical inner products $\langle \mathbf{a}_i, \mathbf{b}_j \rangle = \langle \mathbf{a}'_i, \mathbf{b}'_j \rangle$. We now further modify the FE scheme to encode vectors with some trailing zeros.

**Our FE scheme, version 3**

- *Key Generation:* To generate $\mathsf{sk}_f$ for $f$, sample $\mathbf{r}$ and publish IPE keys $\mathsf{sk}_f = \{\mathsf{sk}^k_{\mathbf{u}_k}\}$ for vectors $\mathbf{u}'_k = \mathbf{u}_k \,||\, 0$, where $\mathbf{u}_k$ is described in Equation (1).
- *Encryption:* To encrypt $x \in \{0,1\}^n$, sample $\mathbf{s}$ and publish IPE ciphertexts $\mathsf{ct}_x = \{\mathsf{ct}^k_{\mathbf{v}_k}\}$ for vectors $\mathbf{v}'_k = \mathbf{v}_k \,||\, 0$, where $\mathbf{v}_k$ is described in Equation (2).

The trailing zeros do not affect the functionality. But, in the security proof, they provide the crucial "space" for hardwiring the randomized encoding $\Pi^b$ in the function key $\mathsf{sk}_f$, without computing it. More specifically, in the proof, we move to a hybrid, encoding vectors of form $\mathbf{u}''_k = \mathbf{u}_k||\Pi_k$, and $\mathbf{v}''_k = \mathbf{0}||1$. Since $\langle \mathbf{u}''_k, \mathbf{v}''_k \rangle = \langle \mathbf{u}'_k, \mathbf{v}'_k \rangle = \langle \mathbf{u}_k, \mathbf{v}_k \rangle$, by function hiding of IPE, this hybrid is indistinguishable to the honest execution. Notice that in this hybrid, the ciphertext contains no information of the input $x^b$, and the key for a function $f_j$ has the corresponding randomized encoding $\Pi_j^b$ (for $f_j, x^b$) hardwired in. Furthermore, the fact that the randomness share $\mathbf{s}$ disappears eventually allows us to argue that $\{\mathbf{r}_j\mathbf{s}\}$ used for generating $\{\Pi_j^b\}$ are pseudorandom. Then, we can finally invoke the RE security, that $\{\Pi_j^0\}$ and $\{\Pi_j^1\}$ are indistinguishable, to argue that FE security holds.

The proof strategy of using computational assumptions to reduce the FE security to RE security resembles that of many FE and IO schemes in the literature in a high level (e.g., [36]), but the details of how we make this approach go through are very different.

**Additional Challenges** Additional challenges must be addressed in order to make the above security proof overview go through. First, applying joint-SXDH to argue the pseudorandomness of $\{\mathbf{rs}\}$ is tricky. This is because we (have to) compute elements $\Pi_k$ in a randomized encoding $\Pi$ in different groups $g_{l_k}^{\Pi_k}$ in order to further evaluate $\Pi$ in the exponent. But, the collection of elements $\{\Pi_k\}$ are correlated through shared randomness $\mathbf{rs}$. An attacker can potentially leverage this correlation, and through computation over different groups, distinguish $\Pi_k$ generated from $\mathbf{rs}$ and that from true randomness. It turns out that the structure of the tree-GES, together with the join-SXDH assumption is exactly what we need to prevent all attacks that arise out of such correlations.

Second, the above proof relies on a secret key IPE that is fully function hiding. Looking back into the literature, we see that the BJK secret key IPE [16] is only weak function hiding. A followup work [28] constructed fully function hiding secret-key IPE. In this work, we show how to generically transform any weak function hiding IPE to full function hiding IPE; our transformation is black-box, extremely simple and of independent interest.

A further issue is that these function hiding secret-key IPE schemes do not produce the inner product in the exponent directly $g^{\langle \mathbf{u},\mathbf{v}\rangle}$, but produce the inner product masked by a scalar $(g^{\langle \mathbf{u},\mathbf{v}\rangle\theta}, g^{\theta})$, where the scalar $\theta$ is determined by the randomness used in key generation and encryption. This creates the problem that randomized encoding elements computed using different IPE instances are masked by distinct scalars $(g_{l_k}^{\Pi_k\theta_k}, g^{\theta_k})$, preventing RE evaluation in the exponent. To resolve this, in our FE scheme, the secret-key IPE instances $\{\mathsf{sk}^k_{\mathbf{u}_k}, \mathsf{ct}^k_{\mathbf{v}_k}\}$ are generated using different master secret keys, but the *same randomness*. Thus, they produce randomized encoding elements masked by the same scalar $(g_{l_k}^{\Pi_k\theta}, g^{\theta})$ and then evaluation can be done as before. As a result, we need function hiding secret-key IPE that allows sharing randomness among instances generated using different master secret keys. It turns out that our function hiding secret-key IPE derived from the BJK scheme has this property.

Stitching all pieces together, we obtain a secret-key FE for $\mathsf{NC}^0$ with linear efficiency, from constant-depth tree-GES.

**Slotted IPE and Public-key FE.** We have to go one step further to construct a public-key FE for $\mathsf{NC}^0$ with linear efficiency. The natural first idea is to use, instead of a secret-key function-hiding IPE scheme, a public-key function-hiding IPE. However, a moment's reflection tells us that such an object cannot possibly exist: that is, the properties of function-hiding and being public-key do not play well with each other.

Our solution to this issue is to construct a "hybrid" encryption scheme that we call a *slotted inner product encryption* (or slotted IPE) scheme. Roughly speaking, a slotted IPE scheme generates keys for vectors $\mathbf{y}_{pub}||\mathbf{y}_{priv}$, encrypts vectors $\mathbf{x}_{pub}||\mathbf{x}_{priv}$, and given the functional secret key, computes an inner product between them. Crucially, a slotted IPE scheme has the following seemingly contradictory properties:

- *Public Key for the first Slot:* Anyone can encrypt vectors of the form $\mathbf{x}_{pub}||\mathbf{0}$. (However, it is computationally hard to encrypt any vector with a non-zero component in the second slot.) The usual notion of semantic security holds, that is, encryption of $\mathbf{x}_{pub}||\mathbf{x}_{priv}$ and $\mathbf{x'}_{pub}||\mathbf{x'}_{priv}$ are indistinguishable if all published function keys do not separate them.
- *Function Hiding for the second Slot:* We require hiding for the second component of the vector in the secret key. That is, the following two worlds are indistinguishable: In the first world, one gets the secret key for $\mathbf{y}_{pub}||\mathbf{y}_{priv}$ and the ciphertext for $0||\mathbf{x}_{priv}$, and in the second world, one gets the secret key for $\mathbf{y}_{pub}||\mathbf{y}'_{priv}$ and the ciphertext for $0||\mathbf{x}'_{priv}$ such that $\langle \mathbf{x_{priv}}, \mathbf{y_{priv}}\rangle = \langle \mathbf{x_{priv}}', \mathbf{y_{priv}}'\rangle$.

It turns out that this notion is the right combination of public-key and function-hiding FE which is both achievable and useful. Slotted IPE is similar in spirit to objects defined in [40] and also similar to (but simpler than) the slotted FE definition of [32]. Replacing the secret-key IPE with a slotted IPE in our construction yields a public-key FE for $\mathsf{NC}^0$ with linear efficiency.

*Constructing Slotted IPE from Joint SXDH.* The final piece of the puzzle is to construct a slotted IPE scheme. We do this by combining our secret key function hiding IPE scheme, derived from the BJK scheme [16], and the public-key IPE scheme of Abdalla et al. [2].

### C. Noisy Graded Encodings

So far, we described our constructions and security proof in the language of *clean* graded encodings, however all the instantiations [29], [27], [34] are for *noisy* graded encodings. Our constructions of FE for $\mathsf{NC}^0$ and IO can be instantiated with noisy graded encodings. However, the security reduction to the joint-SXDH assumption does not go through, as the reduction performs scalar multiplication in order to "re-purpose" elements in the joint-SXDH assumption, to elements in the security experiments of FE for $\mathsf{NC}^0$. Known noisy instantiations, when modified to support scalar multiplication, succumb to attacks. Nevertheless, our FE for $\mathsf{NC}^0$ scheme when instantiated with *ideal* noisy graded encodings is secure. We leave it as an interesting open question whether our construction of FE or IO is secure (or can be made secure) in the recently proposed *weakly ideal* multilinear model [45], [46], [33] which seems to capture all known attacks against noisy graded encodings.

### D. Local PRGs

We briefly survey constructions of low depth PRGs. See Applebaum's book [8] for more references and discussions. Applebaum, Ishai, and Kushilevitz [10] showed that any PRG in $\mathsf{NC}^1$ can be efficiently "compiled" into a PRG in $\mathsf{NC}^0$ using randomized encodings, but with only *sub-linear* stretch. Unfortunately, to the best of our knowledge, there is no construction of PRG in $\mathsf{NC}^0$ with super-linear stretch from well-known assumptions. But, there are candidate constructions.

The authors of [10] constructed a *linear-stretch* PRG in $\mathsf{NC}^0$ under a specific intractability assumption related to the hardness of decoding "sparsely generated" linear codes [11], previously conjectured by Alekhnovich [3]. Goldreich's one-way functions $f : \{0,1\}^n \to \{0,1\}^m$ where each bit of output is a fixed predicate $P$ of a constant number $d$ of input bits chosen at random, is also a candidate PRG when $m > n$. Several works investigated the (in)security of Goldreich's OWFs and PRGs: So far, there are no successful attacks when the choice of the predicate $P$ avoids certain degenerate cases [26], [20], [47], [13].

We refer the reader to the full version of this paper for detailed constructions and proofs.

*Acknowledgements.* The authors thank Hoeteck Wee for being the intellectual inspiration behind this work, Stefano Tessaro for many helpful inputs and insights, and Benny Applebaum for being a quick and reliable oracle. Huijia Lin was partially supported by NSF grants CNS-1528178 and CNS-1514526. Vinod Vaikuntanathan was supported by NSF Grants CNS-1350619 and CNS-1414119 and by the U.S. Army Research Office under contract W911NF-15-C-0226.

## References

[1] M. Abdalla, F. Bourse, A. D. Caro, and D. Pointcheval, "Better security for functional encryption for inner product evaluations," *IACR Cryptology ePrint Archive*, vol. 2016, p. 11, 2016.

[2] M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval, "Simple functional encryption schemes for inner products," in *PKC 2015*, vol. 9020 of *LNCS*, (Gaithersburg, MD, USA), pp. 733–751, Mar. 30 – Apr. 1, 2015.

[3] M. Alekhnovich, "More on average case vs approximation complexity," in *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pp. 298–307, 2003.

[4] P. Ananth, Z. Brakerski, G. Segev, and V. Vaikuntanathan, "From selective to adaptive security in functional encryption," in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, vol. 9216 of *Lecture Notes in Computer Science*, pp. 657–677, 2015.

[5] P. Ananth and A. Jain, "Indistinguishability obfuscation from compact functional encryption," in *CRYPTO 2015, Part I*, vol. 9215 of *LNCS*, (Santa Barbara, CA, USA), pp. 308–326, Aug. 16–20, 2015.

[6] P. Ananth, A. Jain, and A. Sahai, "Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption," *IACR Cryptology ePrint Archive*, vol. 2015, p. 730, 2015.

[7] P. V. Ananth, D. Gupta, Y. Ishai, and A. Sahai, "Optimizing obfuscation: Avoiding Barrington's theorem," in *ACM CCS 14*, (Scottsdale, AZ, USA), pp. 646–658, Nov. 3–7, 2014.

[8] B. Applebaum, *Cryptography in Constant Parallel Time*. Information Security and Cryptography, Springer, 2014.

[9] B. Applebaum and Z. Brakerski, "Obfuscating circuits via composite-order graded encoding," in *TCC 2015, Part II*, vol. 9015 of *LNCS*, (Warsaw, Poland), pp. 528–556, Mar. 23–25, 2015.

[10] B. Applebaum, Y. Ishai, and E. Kushilevitz, "Cryptography in $\mathrm{nc}^0$," in *FOCS*, pp. 166–175, 2004.

[11] B. Applebaum, Y. Ishai, and E. Kushilevitz, "On pseudorandom generators with linear stretch in $\mathrm{nc}^0$," *Computational Complexity*, vol. 17, no. 1, pp. 38–69, 2008.

[12] B. Applebaum, Y. Ishai, and E. Kushilevitz, "How to garble arithmetic circuits," *SIAM J. Comput.*, vol. 43, no. 2, pp. 905–929, 2014.

[13] B. Applebaum and S. Lovett, "Algebraic attacks against random local functions and their countermeasures," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 22, p. 172, 2015.

[14] B. Barak, S. Garg, Y. T. Kalai, O. Paneth, and A. Sahai, "Protecting obfuscation against algebraic attacks," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, vol. 8441 of *Lecture Notes in Computer Science*, pp. 221–238, 2014.

[15] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," in *Advances in Cryptology CRYPTO 2001*, pp. 1–18, Springer, 2001.

[16] A. Bishop, A. Jain, and L. Kowalczyk, "Function-hiding inner product encryption," in *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, vol. 9452 of *Lecture Notes in Computer Science*, pp. 470–491, 2015.

[17] N. Bitansky, S. Goldwasser, A. Jain, O. Paneth, V. Vaikuntanathan, and B. Waters, "Time-lock puzzles from randomized encodings," in *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pp. 345–356, 2016.

[18] N. Bitansky, O. Paneth, and A. Rosen, "On the cryptographic hardness of finding a nash equilibrium," in Guruswami [38], pp. 1480–1498.

[19] N. Bitansky and V. Vaikuntanathan, "Indistinguishability obfuscation from functional encryption," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pp. 171–190, 2015.

[20] A. Bogdanov and Y. Qiao, "On the security of goldreich's one-way function," *Computational Complexity*, vol. 21, no. 1, pp. 83–127, 2012.

[21] D. Boneh and A. Silverberg, "Applications of multilinear forms to cryptography," *IACR Cryptology ePrint Archive*, vol. 2002, p. 80, 2002.

[22] Z. Brakerski and G. N. Rothblum, "Virtual black-box obfuscation for all circuits via generic graded encoding," in *TCC 2014*, vol. 8349 of *LNCS*, (San Diego, CA, USA), pp. 1–25, Feb. 24–26, 2014.

[23] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan, "Obfuscation of probabilistic circuits and applications," in *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, vol. 9015 of *Lecture Notes in Computer Science*, pp. 468–497, 2015.

[24] R. Canetti and V. Vaikuntanathan, "Obfuscating branching programs using black-box pseudo-free groups," *IACR Cryptology ePrint Archive*, vol. 2013, p. 500, 2013.

[25] A. Cohen, J. Holmgren, R. Nishimaki, V. Vaikuntanathan, and D. Wichs, "Watermarking cryptographic capabilities," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1096, 2015. To Appear in ACM STOC 2016.

[26] J. Cook, O. Etesami, R. Miller, and L. Trevisan, "Goldreich's one-way function candidate and myopic backtracking algorithms," in *TCC*, pp. 521–538, 2009.

[27] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Practical multilinear maps over the integers," in *CRYPTO 2013, Part I*, vol. 8042 of *LNCS*, (Santa Barbara, CA, USA), pp. 476–493, Aug. 18–22, 2013.

[28] P. Datta, R. Dutta, and S. Mukhopadhyay, "Functional encryption for inner product with full function privacy," in *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part I*, vol. 9614 of *Lecture Notes in Computer Science*, pp. 164–195, 2016.

[29] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, vol. 7881 of *Lecture Notes in Computer Science*, pp. 1–17, 2013.

[30] S. Garg, C. Gentry, S. Halevi, and M. Raykova, "Two-round secure MPC from indistinguishability obfuscation," in *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, vol. 8349 of *Lecture Notes in Computer Science*, pp. 74–94, 2014.

[31] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pp. 40–49, 2013.

[32] S. Garg, C. Gentry, S. Halevi, and M. Zhandry, "Functional encryption without obfuscation," in *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, vol. 9563 of *Lecture Notes in Computer Science*, pp. 480–511, 2016.

[33] S. Garg, P. Mukherjee, and A. Srinivasan, "Obfuscation without the vulnerabilities of multilinear maps," *IACR Cryptology ePrint Archive*, vol. 2016, p. 390, 2016.

[34] C. Gentry, S. Gorbunov, and S. Halevi, "Graph-induced multilinear maps from lattices," in *TCC 2015, Part II*, vol. 9015 of *LNCS*, (Warsaw, Poland), pp. 498–527, Mar. 23–25, 2015.

[35] C. Gentry, A. B. Lewko, and B. Waters, "Witness encryption from instance independent assumptions," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, vol. 8616 of *Lecture Notes in Computer Science*, pp. 426–443, 2014.

[36] C. Gentry, A. B. Lewko, A. Sahai, and B. Waters, "Indistinguishability obfuscation from the multilinear subgroup elimination assumption," in Guruswami [38], pp. 151–170.

[37] S. Goldwasser and G. N. Rothblum, "On best-possible obfuscation," in *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, vol. 4392 of *Lecture Notes in Computer Science*, pp. 194–213, 2007.

[38] V. Guruswami, ed., *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, IEEE Computer Society, 2015.

[39] P. Hubácek and E. Yogev, "Hardness of continuous local search: Query complexity and cryptographic lower bounds," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 23, p. 63, 2016.

[40] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT 2008*, pp. 146–162, 2008.

[41] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT 2010*, vol. 6110 of *LNCS*, (French Riviera), pp. 62–91, May 30 – June 3, 2010.

[42] H. Lin, "Indistinguishability obfuscation from constant-degree graded encoding schemes," 2016. To Appear in Eurocrypt'16.

[43] M. Mahmoody, A. Mohammed, and S. Nematihaji, "On the impossibility of virtual black-box obfuscation in idealized models," in *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pp. 18–48, 2016.

[44] M. Mahmoody, A. Mohammed, S. Nematihaji, R. Pass, and A. Shelat, "Lower bounds on assumptions behind indistinguishability obfuscation," in *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pp. 49–66, 2016.

[45] E. Miles, A. Sahai, and M. Zhandry, "Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13," *IACR Cryptology ePrint Archive*, vol. 2016, p. 147, 2016.

[46] E. Miles, A. Sahai, and M. Zhandry, "Secure obfuscation in a weak multilinear map model: A simple construction secure against all known attacks," *IACR Cryptology ePrint Archive*, vol. 2016, p. 588, 2016.

[47] R. O'Donnell and D. Witmer, "Goldreich's PRG: evidence for near-optimal polynomial stretch," in *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pp. 1–12, 2014.

[48] R. Pass, K. Seth, and S. Telang, "Indistinguishability obfuscation from semantically-secure multilinear encodings," in *CRYPTO 2014, Part I*, vol. 8616 of *LNCS*, (Santa Barbara, CA, USA), pp. 500–517, Aug. 17–21, 2014.

[49] R. Pass and A. Shelat, "Impossibility of VBB obfuscation with ideal constant-degree graded encodings," in *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pp. 3–17, 2016.

[50] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more," in *46th ACM STOC*, (New York, NY, USA), pp. 475–484, May 31 – June 3, 2014.

[51] J. Zimmerman, "How to obfuscate programs directly," in *EUROCRYPT 2015, Part II*, vol. 9057 of *LNCS*, (Sofia, Bulgaria), pp. 439–467, Apr. 26–30, 2015.