On Removing Graded Encodings from Functional Encryption

Nir Bitansky[1⋆], Huijia Lin[2⋆⋆], and Omer Paneth[1]

[1] MIT
[2] UCSB

Abstract. Functional encryption (FE) has emerged as an outstanding concept. By now, we know that beyond the immediate application to computation over encrypted data, variants with *succinct ciphertexts* are so powerful that they yield the full might of indistinguishability obfuscation (IO). Understanding how, and under which assumptions, such succinct schemes can be constructed has become a grand challenge of current research in cryptography. Whereas the first schemes were based themselves on IO, recent progress has produced constructions based on *constant-degree graded encodings*. Still, our comprehension of such graded encodings remains limited, as the instantiations given so far have exhibited different vulnerabilities.

Our main result is that, assuming LWE, *black-box constructions* of *sufficiently succinct* FE schemes from constant-degree graded encodings can be transformed to rely on a much better-understood object — *bilinear groups*. In particular, under an *über assumption* on bilinear groups, such constructions imply IO in the plain model. The result demonstrates that the exact level of ciphertext succinctness of FE schemes is of major importance. In particular, we draw a fine line between known FE constructions from constant-degree graded encodings, which just fall short of the required succinctness, and the holy grail of basing IO on better-understood assumptions.

In the heart of our result, are new techniques for removing ideal graded encoding oracles from FE constructions. Complementing the result, for weaker ideal models, namely the generic group model and the random oracle model, we show a transformation from *collusion-resistant* FE in either of the two models directly to FE (and IO) in the plain model, without assuming bilinear groups.

1 Introduction

Functional Encryption (FE) is a fascinating object. It enables fine-grained control of encrypted data, by allowing users to learn only specific functions of the

⋆ Supported by NSF Grants CNS-1350619 and CNS-1414119, and the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226. Part of this research was done while visiting Tel Aviv University and supported by the Leona M. & Harry B. Helmsley Charitable Trust and Check Point Institute for Information Security.

⋆⋆ Partially supported by NSF grants CNS-1528178 and CNS-1514526.

data. This ability is captured trough the notion of *function keys*. A function key SK_f, associated with a function f, allows to partially decrypt a ciphertext CT_x encrypting an input x in a way that reveals $f(x)$ and nothing else.

A salient aspect of FE schemes is their *ciphertext succinctness*. Focusing on the setting of (indistinguishability-based) *single-key* FE where only one function key SK_f is supported, we say that an FE scheme is *weakly succinct* if the ciphertext size scales *sub-linearly* in the size of the circuit f; namely,

$$|\mathsf{CT}_x| \leq |f|^{\gamma} \cdot \mathrm{poly}(|x|), \qquad \text{for } \textit{some} \text{ constant } \textit{compression factor } \gamma < 1.^{3}$$

While non-succinct single-key FE schemes (where we allow the size of ciphertexts to grow polynomially with $|f|$) are equivalent to public-key encryption (or just one-way functions, in the secret-key setting) [42, 29], weakly succinct schemes are already known to be extremely strong. In particular, subexponentially-secure weakly-succinct FE for functions in $\mathbf{NC}^1$ implies indistinguishability obfuscation (IO) [6, 1, 11], and has far reaching implications in cryptography and beyond (*e.g.*, [25, 24, 43, 10, 7, 18]).[4]

Thus, understanding how, and under which assumptions, weakly-succinct FE can be constructed has become a central question in cryptographic research. While schemes for Boolean functions in $\mathbf{NC}^1$ have been constructed from LWE [28], the existence of such FE scheme for non-Boolean functions (which is required for the above strong implications) is still not well-founded, and has been the subject of a substantial body of work. The first construction of general purpose FE that achieves the required succinctness relied itself on IO [25]. Subsequent constructions were based on the algebraic framework of *multilinear graded encodings* [22]. Roughly speaking, this framework extends the traditional concept of *encoding in the exponent* in groups. It allows *encoding* values in a field (or ring), evaluating polynomials of a certain bounded *degree* d over the encoded values, and testing whether the result is zero.

Based on graded encodings of polynomial degree Garg, Gentry, Halevi, and Zhandry [26] constructed *unbounded-collusion* FE, which in turn is known to lead to weakly succinct FE [2, 11]. Starting from the work of Lin [32], several works [36, 3, 33] have shown that assuming also pseudorandom generators with constant locality, weakly-succinct FE can be constructed based on *constant-degree* graded encodings under simple assumptions like asymmetric DDH. However, these constructions require constant degree $d \geq 5$.

[3] Here *weak* succinctness is in contrast to *full* succintness, where the ciphertext size does not depend at all on the function size.

[4] Formally, [1, 11] require that not only the ciphertext is succinct, but also the encryption circuit itself. This difference can be bridged assuming LWE [35], and for simplicity is ignored in this introduction. Our results will anyhow rely on LWE.

Despite extensive efforts, our understanding of graded encodings of any degree larger than two is quite limited. Known instantiations are all based on little-understood lattice problems, and have exhibited different vulnerabilities [22, 20, 21, 15, 39]. In contrast, bilinear group encodings [14, 30], akin to degree-2 graded encodings, have essentially different instantiations based on elliptic curve groups, which are by now quite well understood and considered standard. Bridging the gap between degree 2 and degree $d > 2$ is a great challenge.

Our Main Result in a Nutshell: Size Matters. We show that the exact level of succinctness in FE schemes has a major impact on the latter challenge. Roughly speaking, we prove that *black-box constructions* [41] of weakly-succinct FE from degree-d graded encodings, with compression factor $\gamma < \frac{1}{d}$, can be transformed to rely *only on bilinear groups*. Specifically, assuming LWE [5] and for any constant ε, starting from $\frac{1}{d+\varepsilon}$-succinct FE in the *ideal degree-d graded encoding model*, we construct weakly-succinct FE in the *ideal bilinear model*.

The ideal graded encoding model generalizes the classical generic-group model [44]. In this model, the construction as well as the adversary perform all graded encoding operations through an *ideal oracle*, without access to an explicit representation of encoded elements. Having this ideal model as a starting point allows capturing a large class of constructions and assumptions, as it models *perfectly secure* graded encodings. Indeed, the FE schemes in [32, 36, 3, 33] can be constructed and proven secure in this model.

The resulting construction from ideal bilinear encodings can further be instantiated in the plain model using existing bilinear groups, and proven secure under an *über assumption* on bilinear groups [13, 16]. In particular, assuming also subexponential-security, it implies IO in the plain model.

How Close are We to IO from Bilinear Maps? Existing weakly-succinct FE schemes in the ideal constant-degree model [32, 36, 3, 33] have a compression factor $\gamma = C/d$, for some absolute constant $C > 1$. Thus, our result draws a fine line that separates known FE constructions based on constant-degree graded encodings and constructions that would already take us to the promised land of IO based on much better-understood mathematical objects. Crossing this line may very well require a new set of techniques. Indeed, one may also interpret our result as a negative one, which puts a barrier on black-box constructions of FE from graded encodings.

Discussion: Black-Box vs Non-Black-Box Constructions. For IO schemes (rather than FE), a combination of recent works [40, 12] demonstrates that

[5] More precisely, we need to assume the hardness of LWE with subexponential modulus-to-noise ratio. For simplicity, we ignore the parameters of LWE in this introduction; see Section 4 for more details.

black-box constructions from constant-degree graded encodings are already very powerful. They show that any IO construction relative to a constant-degree oracle can be converted to a construction in the plain model (under standard assumptions, like DDH). Since weakly-succinct FE schemes imply IO, we may be lead to think that weakly-succinct black-box constructions of FE from constant-degree graded encodings would already imply IO in the plain model from standard assumptions. Interestingly, this is not the case.

The crucial point is that the known transformations from FE to IO [1, 11] are *non-black-box*, they use the code of the underlying FE scheme, and thus do not *relativize* with respect to graded encoding oracle. That is, we do not know how to move from an FE scheme based on graded encodings to an IO scheme that uses graded encodings in a black-box way. Indeed, if there existed such a black-box transformation between FE and IO, then combining [40, 37, 12, 32, 36], IO in the plain model could be constructed from standard assumptions.

Instead, we show how to directly remove constant-degree oracles from FE. Our transformation relies on new techniques that are rather different than those used in the above works for removing such oracles from IO.

1.1 Our Results in More Detail

We now describe our results in further detail. We start by describing the ideal graded encoding model and the ideal bilinear encoding model more precisely.

The Ideal Graded Encoding Model. A graded encoding [23] is an encoding scheme for elements of some field.[6] The encoding supports a restricted set of homomorphic operations that allow one to evaluate certain polynomials over the encoded field elements and test whether these polynomials evaluate to zero or not. Every field element is encoded with respect to a *label* (sometimes called the *level* of the encoding). For a given sequence of encodings, their labels control which polynomials are valid and can be evaluated over the encodings. The *degree* of the graded encoding is the maximal degree of a polynomial that is valid with respect to any sequence of labels.

In the ideal graded encoding model, explicit encodings are replaced by access to an oracle that records the encoded field elements and provides an interface to perform operations over the elements. Different formalizations of such ideal graded encoding oracles exist in the literature (*e.g.* [17, 5, 4, 40]) and differ in details. In this work, we follow the model of Pass and shelat in [40].

[6] For ease of exposition, we consider graded encodings over fields. Our results can also be obtained with any commutative ring in which it is computationally hard to find non-unit elements.

The ideal graded encoding oracle $\mathcal{M}$ is specified by a field $\mathbb{F}$ and a validity predicate V operating on a polynomial and labels taken from a set $\mathbb{L}$. The oracle $\mathcal{M} = (\mathbb{F}, V)$ provides two functions — encoding and zero-testing.

Encoding: Given a field element $\xi \in \mathbb{F}$ and a label $\ell \in \mathbb{L}$ the oracle $\mathcal{M}$ samples a sufficiently long random string r to create a *handle* $h = (r, \ell)$. It records the pair (h, ξ) associating the handle with the encoded field element.

Zero-testing: a query to $\mathcal{M}$ consists of a polynomial p and a sequence of handles $h_1, \cdots, h_m$ where h_i encodes the field elements ξ_i relative to label ℓ_i. $\mathcal{M}$ tests if the polynomial and the labels satisfy the validity predicate and whether the polynomial vanishes on the corresponding field elements. That is, $\mathcal{M}$ returns `true` if and only if $V(p, \ell_1, \cdots \ell_m) = \texttt{true}$ and $p(\xi_1, \cdots \xi_m) = 0$.

Like in [40], we restrict attention to well-formed validity predicates. For such predicates, a polynomial p is valid with respect to labels $\ell_1, \cdots, \ell_m$, if and only if every monomial Φ in p is valid with respect to the labels of the handles that Φ acts on. Indeed, existing graded encodings all consider validity predicates that are *well-formed*.[7]

The Ideal Bilinear Encoding Model. The ideal bilinear encoding model corresponds to the ideal graded encoding model where valid polynomials are of degree at most two. We note that in the ideal graded encoding model described above, encoding is a randomized operation. In particular, encoding the same element and label (ξ, ℓ) twice gives back two different handles. In contrast, traditional instantiations of the ideal bilinear encoding model are based on bilinear pairing groups (such as elliptic curve groups) where the encoding is a deterministic function. We can naturally capture such instantiations, by augmenting the ideal bilinear encoding model to use a *unique handle* for every pair of field element and label (as done for instance in [4, 45, 36]).

The Main Result. Our main result concerns FE schemes in the ideal graded encoding model. In such FE schemes, all algorithms (setup, key derivation, encryption, and decryption), as well as all adversaries against the scheme, have access to a graded encoding oracle $\mathcal{M}$. We show:

Theorem 1 (Informal). *Assume the hardness of LWE. For any constants $d \in \mathbb{N}$ and $\gamma \leq \frac{1}{d}$, any γ-succinct secret-key FE scheme for **P**/**poly**, in the ideal degree-d graded encoding model, can be transformed into a weakly-succinct public-key FE scheme for **P**/**poly** in the ideal bilinear encoding model.*

[7] In the body, we make another structural requirement on validity predicates called *decomposability*. This requirement is somewhat more technical, but is also satisfied by all known formulations of graded encodings. For the simplified technical exposition in this introduction it can be ignored. See further details in Definition 4.

IO in the Plain Model under an Über Assumption. Our main transformation results in a weakly-succinct public-key FE scheme in the ideal bilinear encoding model. By instantiating the ideal bilinear encoding oracle with concrete bilinear pairing groups, we get a corresponding FE scheme in the plain model. For security to hold, we make an über assumption [13] on the bilinear groups. An über assumption essentially says that two encoded sequences of elements in the plain model can be distinguished only if they are also distinguishable in the ideal model. There are no known attacks on the über security of existing instantiations of bilinear pairing groups.

Since weakly-succinct public-key FE with subexponential security in the plain model implies IO we deduce the following corollary

Corollary 1 (Informal). *Assume subexponential hardness of LWE and bilinear groups with subexponential über security. For any constants $d \in \mathbb{N}, \gamma < \frac{1}{d}$, any subexponentially-secure, γ-succinct, secret-key FE for $\boldsymbol{P/poly}$ in the ideal degree-d graded encoding model, can be transformed into an IO scheme for $\boldsymbol{P/poly}$ in the plain model.*

FE in Weaker Ideal Models. We also consider FE schemes in ideal models that are weaker than the ideal bilinear encoding model. Specifically, we consider the generic-group model (that corresponds to the ideal degree-1 graded encoding model) and the random-oracle model. We give transformations from FE in these models directly to FE in the plain model *without relying on bilinear encodings*.

In the transformation given by Theorem 1, from the ideal constant-degree graded encoding model to the ideal bilinear encoding model, we considered the notion of single-key weakly succinct FE. In contrast, our transformations from the generic-group model and the random-oracle model to the plain model require that we start with a stronger notion of *collusion-resistant FE*. Collusion-resistance requires security in the presence of an *unbounded* number of functional keys. Crucially, ciphertexts are required not to grow with the number of keys (but are allowed to grow polynomially in the size of the evaluated functions).

Collusion-resistant FE is known to imply weakly-succinct FE through a black-box transformation [2, 11]. In the converse direction, only a non-black-box transformation is known [27, 31], and therefore we cannot apply it to ideal model constructions of FE.

Theorem 2 (Informal). *Assume the hardness of LWE. Any collusion-resistant secret-key FE scheme in the generic-group model, or in the random-oracle model, can be transformed into a collusion-resistant public-key FE scheme in the plain model.*

1.2 Our Techniques

We next give an overview of the main ideas behind our degree-reduction transformation given by Theorem 1.

Can We Adopt Techniques from IO? As already mentioned, we do not know how to transform FE schemes into IO schemes in a black-box way. Thus, we cannot rely directly on existing results that remove ideal oracles from IO [40]. Furthermore, trying to import ideas from these results in the IO setting to the setting of FE encounters some inherent difficulties, which we now explain.

Roughly speaking, removing ideal oracles from IO is done as follows. Starting with a scheme in an ideal oracle model, we let the obfuscator emulate the oracle by itself and publish, together with the obfuscated circuit, some *partial view* of the self-emulated oracle. This partial view is on one hand, sufficient to preserve the functionality of the obfuscated circuit on most inputs, and on the other hand, does not compromise security. The partial view is obtained by evaluating the obfuscation on many random inputs (consistently with the self-emulated oracle), observing how evaluation interacts with the oracle, and performing a certain *learning process*. Arguing that the published partial view does not compromise security crucially relies on the fact that *evaluating the obfuscated program is a public procedure that does not share any secret state with the obfuscator*.

The setting of FE, however, is somewhat more complicated. Here rather than an evaluator we have a decryptor that given a function key SK_f and ciphertext CT encrypting x, should be able to compute $f(x)$. In contrast to the evaluator in obfuscation, the state of the decryptor is *not publicly samplable*. Indeed, generating function keys SK_f for different functions requires knowing a master secret key. Accordingly, it is not clear how to follow the same approach as before.

XIO instead of IO. Nevertheless, we observe that there is a way to reduce the problem to a setting much more similar to IO. Specifically, *there exists [9] a black-box transformation from FE to a weaker version of IO called XIO*. XIO [34], which stands for *exponentially-efficient IO*, allows the obfuscation and evaluation algorithms to run in exponential time $2^{O(n)}$ in the input size n, and only requires that the *size* of an obfuscation $\widetilde{C}$ of a circuit C is slightly subexponenetial in n:

$$|\widetilde{C}| \leq 2^{\gamma n} \cdot \mathrm{poly}(|C|) \quad \text{for some constant } \textit{compression factor } \gamma < 1 \ .$$

Despite this inherent inefficiency, [34] show that XIO for *logarithmic-size* inputs implies IO assuming subexponential hardness of LWE. A natural direction is thus to try and apply the techniques used to remove oracles from IO to remove

the same oracles also from XIO; indeed, if this can be done, such oracles can also be removed from FE, due to the black-box transformation between the two.

This, again, does not work as is. The issue is that the transformations removing degree-d graded encoding oracle from IO may blow up the size of the original obfuscation from $|\widetilde{C}|$ in the oracle model to roughly $|\widetilde{C}|^{2d}$ in the plain model. However, the known black-box construction of XIO from FE [9] is not sufficiently compressing to account for this blowup. Even starting from FE with great compression, say $\gamma_{\text{FE}} < d^{-10}$, the resulting XIO has a much worst compression factor $\gamma_{\text{XIO}} > 1/2$. In particular, composing the two would result in a useless plain model obfuscation of exponential size $2^{n \cdot d}$.

Motivating our Solution. To understand our solution, let us first describe an over-simplified candidate transformation for reducing XIO with constant-degree graded encoding oracles to XIO with degree-1 oracles (akin to the generic-group model). This transformation will suffer from the same size blowup of the transformations mentioned above.

For simplicity of exposition, we first restrict attention to XIO schemes with the following simple structure:

- Any obfuscated circuit $\widetilde{C}$ consists of a set of handles $h_1, \ldots, h_m$ corresponding to field elements $\xi_1, \ldots, \xi_m$ encoded during obfuscation, under certain labels $\ell_1, \ldots, \ell_m$.
- Evaluation on any given input x consists of performing valid zero-tests over the above handles, which are given by degree-d polynomials $p_1, \ldots, p_k$.

A simple idea to reduce the degree-d oracle to a linear oracle is to change the obfuscation algorithm so that it computes ahead of time the field elements ξ_Φ corresponding to all valid degree-d monomials $\Phi(\xi_1, \ldots, \xi_m) = \prod_{i \in [d]} \xi_{j_i}$. Then, rather than using the degree-d oracle, it uses the linear oracle to encode the field elements ξ_Φ, and publishes the corresponding handles $\{h_\Phi\}_\Phi$. Evaluation is done in a straight forward manner by writing any zero-test polynomial p of degree d as a linear function in the corresponding monomials

$$p(\xi_1, \cdots, \xi_m) = \sum_{\Phi} \alpha_\Phi \Phi(\xi_1, \ldots, \xi_m) \ ,$$

and making the corresponding zero-test query $L_p(\{h_\Phi\}) := \sum_\Phi \alpha_\Phi h_\Phi$ to the linear oracle.

Indeed, the transformation blows up the size of the obfuscated circuit from roughly m, the number of encodings in the original obfuscation, to m^d, the number of all possible monomials. While such a polynomial blowup is acceptable in the context of IO, for XIO with compression $d^{-1} \leq \gamma < 1$, it is devastating.

Key Idea: XIO in Decomposable Form. To overcome the above difficulty, we observe that the known black-box construction of XIO from FE [9] has certain structural properties that we can exploit. At a very high level, it can be decomposed into smaller pieces, so that instead of computing *all* monomials over *all* the encodings created during obfuscation, we only need to consider a much smaller subset of monomials. In this subset, each monomial only depends on a few small pieces, and thus only on few encodings.

To be more concrete, we next give a high-level account of this construction. To convey the idea in a simple setting of parameters, let us assume that we have at our disposal an FE scheme that support an unbounded number of keys, rather than a single key scheme, with the guarantee that the size of ciphertexts does not grow with the number of keys. In this case, the XIO scheme in [9] works as follows:

- To obfuscate a circuit C with n input bits, the scheme publishes a collection of function keys $\{\mathsf{SK}_{D_\tau}\}_\tau$ for circuits D_τ, indexed by prefixes $\tau \in \{0,1\}^{n/2}$ (will be specified shortly), and a collection of ciphertexts $\left\{\mathsf{CT}_{\rho\|C}\right\}_\rho$, each encrypting the circuit C and a suffix $\rho \in \{0,1\}^{n/2}$.
- Decrypting a ciphertext $\mathsf{CT}_{\rho\|C}$ with key SK_{D_τ} reveals $D_\tau(\rho\|C) := C(\tau,\rho)$.

The obfuscated circuit indeed has slightly subexponential size . It contains:

- $2^{n/2}$ function keys SK_{D_τ}, each of size $\mathrm{poly}(|C|)$,
- $2^{n/2}$ ciphertexts $\mathsf{CT}_{\rho\|C}$, each of size $\mathrm{poly}(|C|)$.

Going back to the ideal graded-encoding model, the FE key generation and encryption algorithms use the ideal oracle to encode elements. Therefore, generating the obfuscation involves generating a set of k encodings $\boldsymbol{h}_\tau = \{h_{\tau,i}\}_{i\in[k]}$ for each secret key SK_{D_τ} and a set of k encodings $\boldsymbol{h}_\rho = \{h_{\rho,i}\}_{i\in[k]}$ for each ciphertext $\mathsf{CT}_{\rho\|C}$, for some $k = \mathrm{poly}(|C|)$. The crucial point is that now, evaluating the obfuscation on a given input (τ,ρ) only involves the two small sets of encodings $\boldsymbol{h}_\tau, \boldsymbol{h}_\rho$. In particular, any zero-test made by the decryption algorithm is a polynomial defined only over the underlying field elements $\boldsymbol{\xi}_\tau = \{\xi_{\tau,i}\}_{i\in[k]}$ and $\boldsymbol{\xi}_\rho = \{\xi_{\rho,i}\}_{i\in[k]}$.

This gives rise to the following degree reduction strategy. In the obfuscation, rather then precomputing all monomials in all encodings as before, we precompute only the monomials corresponding to the different pieces $\left\{\Phi(\boldsymbol{\xi}_\rho)\right\}_{\rho,\Phi}, \{\Phi(\boldsymbol{\xi}_\tau)\}_{\tau,\Phi}$. Now, rather than representing zero-tests made by the decryption algorithm as linear polynomials in these monomials, they can be represented as quadratic polynomials

$$p(\boldsymbol{\xi}_\tau, \boldsymbol{\xi}_\rho) = Q_p\left(\{\Phi(\boldsymbol{\xi}_\tau)\}_\Phi, \left\{\Phi(\boldsymbol{\xi}_\rho)\right\}_\Phi\right) \quad .$$

To support such quadratic zero tests, we resort to bilinear groups. We use the bilinear encoding oracle to encode the values $\{\Phi(\boldsymbol{\xi}_\rho)\}_{\rho,\Phi}, \{\Phi(\boldsymbol{\xi}_\tau)\}_{\tau,\Phi}$, and publish the corresponding handles $\{h_{\tau,\Phi}\}_{\tau,\Phi}, \{h_{\rho,\Phi}\}_{\rho,\Phi}$. Evaluation is done in a straight forward manner by testing the quadratic polynomial Q_p.

The key gain of this construction is that now the blowup is tolerable. Now, each set of k encodings, blows up to k^d, which is acceptable since $k = \mathrm{poly}(|C|)$ is small (and not proportional to the size of the entire obfuscation as before, which is exponential in n). In the body, we formulate a general *product form* property for XIO schemes, which can be used as the starting point of the above-described transformation; we further show that single-key FE schemes with $\frac{1}{d+\varepsilon}$-succinctness implies such XIO schemes.

A Closer Look. The above exposition is oversimplified. To actually fulfill our strategy, we need to overcome two main challenges.

Challenge 1: Explicit Handles. The core idea described above assumes that the obfuscation is simply given as an explicit list of handles, which may not be the case starting from an arbitrary FE scheme. In particular, the obfuscator may use the oracle $\mathcal{M}$ to produce a set of encodings, but not output them explicitly; indeed, it can output an arbitrary string. In this case, we can no longer apply the degree reduction technique, since we do not know which encodings are actually contained in the obfuscation. Naïvely publishing all monomials in all field elements ever encoded by the obfuscator may be insecure — some of these encodings, which are never explicitly included in the obfuscation, may leak information.

To handle XIO schemes constructed from general FE schemes, we need a way to make any "implicit" handles explicit, without compromising security. Our idea is to *learn* the *significant handles* that would later suffice for evaluation on most inputs, and publish them explicitly. This idea is inspired by [19, 40, 37] and their observation (already mentioned above) that in obfuscation, the evaluator's view, including all the handles it sees, is publicly and efficiently samplable.

Roughly speaking, the learning process involves evaluating the obfuscated circuit on many random inputs and making explicit all handles involved in these evaluations. When doing this naïvely, the number of such test evaluations required to guarantee reasonable correctness is proportional to the number of elements encoded by the obfuscator. This would result in a quadratic overhead in the size of the obfuscation, which would again completely foil XIO compression. Avoiding the blowup requires a somewhat more sophisticated learning process that once again exploits the local structure of the construction in [9].

The scheme resulting from the above learning process is only approximately correct — the obfuscation with explicit handles errs on say 10% of the inputs. We show that even such *approximate XIO* is sufficient for obtaining FE and IO in the plain model (this step is described later in this overview).

Challenge 2: Invalid Monomials. Another main challenge is that it may be insecure to publish encodings of all the monomials $\{\Phi(\boldsymbol{\xi}_\rho)\}_{\rho,\Phi}$, $\{\Phi(\boldsymbol{\xi}_\tau)\}_{\tau,\Phi}$. The problem is that some products $\Phi(\boldsymbol{\xi}_\rho) \cdot \Phi'(\boldsymbol{\xi}_\tau)$ may result in monomials that would have been invalid in the degree-d ideal model. For example, $\Phi(\boldsymbol{\xi}_\rho)$ could correspond to a degree-$(d-2)$ monomial Φ. In the degree-d ideal model, it would only be possible to multiply such a monomial by degree-2 monomials $\Phi'(\boldsymbol{\xi}_\tau)$, and zero test. In the the described new scheme, however, it can multiply monomials $\Phi'(\boldsymbol{\xi}_\tau)$ of degree 3, or even d, which might compromise security.

Our solution proceeds in two steps. First, we show how to properly preserve validity by going to a more structured model of bilinear encodings that generalizes *asymmetric* bilinear groups. In this model, every encoding contains one of many labels and only pairs of encodings with valid labels can be multiplied. We then encode the monomials $\{\Phi(\boldsymbol{\xi}_\rho)\}_{\rho,\Phi}$, $\{\Phi(\boldsymbol{\xi}_\tau)\}_{\tau,\Phi}$ with appropriate labels that preserve the information regarding the original set of labels. This guarantees that the set of monomials that can be zero-tested in this model corresponds exactly to the set of valid monomials in the constant-degree graded encoding model we started from.

Second, we show how to transform any construction in this (more structured) ideal model into one in the standard ideal bilinear encoding model (corresponding to *symmetric* bilinear maps). At a very high-level, we develop a "secret-key transformation" from asymmetric bilinear groups to symmetric bilinear groups. The transformation allows anyone in the possession of a secret key to translate encodings in the asymmetric setting to new encodings in the symmetric setting in a manner that enforces the asymmetric structure.

From Approximately-Correct XIO back to FE. After applying all the above steps, we obtain an approximately-correct XIO scheme in the ideal bilinear encoding model. The only remaining step is going from such an XIO scheme back to FE. The work of [35] showed how to construct FE from XIO with *perfect* correctness, assuming in addition LWE. We modify their transformation to construct FE starting directly from approximately-correct XIO. This is done using appropriate Error Correcting Codes to accommodate for the correctness errors from XIO.[8] The transformation uses XIO as a black-box, and can thus be performed in the ideal bilinear model.

[8] We note that existing transformations for removing errors from IO [12] do not work for XIO. See Section 4 for details.

Putting It All Together. Putting all pieces together, we finally obtain our transformation from $\frac{1}{d+\varepsilon}$-succinct FE in the constant-degree graded encoding model to weakly-succinct FE in the bilinear encoding model. To recap the structure of the transformation:

1. Start with a $\frac{1}{d+\varepsilon}$-succinct (single-key) FE in the ideal constant-degree graded encoding model.
2. Transform it into an XIO scheme in the ideal constant-degree graded encoding model satisfying an appropriate decomposition property (which we call product form).
3. Transform it into an approximate XIO scheme in the ideal bilinear encoding model.
4. Use the resulting approximate XIO scheme and LWE to get a weakly-succinct FE (still, in the ideal bilinear encoding model).

Instantiating the oracle in bilinear groups with über security gives a corresponding construction in the plain model.

Organization In Section 2, we define (oracle-aided) XIO, and introduce the constant-degree oracles considered in this work. In Section 3, we show how to transform XIO, in a certain product form, relative to constant-degree oracles to approximate XIO relative to symmetric bilinear oracles. In Section 4, we explain how to move from approximate XIO and LWE, to IO. Due to the space limit, some of the details and proofs are omitted. These can be found in the full version of this paper [8], where we additionally describe how to remove generic-group oracles and random oracles from unbounded collusion FE schemes.

2 Preliminaries

2.1 XIO

We next formally define the notion of exponentially-efficient indistinguishability obfuscation (XIO) for any collection of circuit classes $\mathcal{C} \subseteq \mathbf{P}^{\log}/\mathbf{poly}$, where $\mathbf{P}^{\log}/\mathbf{poly}$ is the collection of all classes of polynomial-size circuits with logarithmic size input. The definition extends the one in [34] by considering also approximate correctness.

Definition 1 ($\mathbf{P}^{\log}/\mathbf{poly}$). *The collection $\boldsymbol{P}^{\log}/\boldsymbol{poly}$ includes all classes $\mathcal{C} = \{\mathcal{C}_\lambda\}$ for which there exists a constant $c = c(\mathcal{C})$, such that the input of any circuit $C \in \mathcal{C}_\lambda$ is bounded by $c \log \lambda$ and the size of C is bounded by λ^c.*

Definition 2 (XIO [34]). *A pair of algorithms* $\mathsf{xiO} = (\mathsf{xiO.Obf}, \mathsf{xiO.Eval})$ *is an* exponentially-efficient indistinguishability obfuscator *(XIO) for a collection of circuit classes* $\boldsymbol{\mathcal{C}} = \{\mathcal{C} = \{\mathcal{C}_\lambda\}\} \subseteq \boldsymbol{P}^{\log}/\boldsymbol{poly}$ *if it satisfies:*

- ***Functionality:*** *for any* $\mathcal{C} \in \boldsymbol{\mathcal{C}}$*, security parameter* $\lambda \in \mathbb{N}$*, and* $C \in \mathcal{C}_\lambda$ *with input size* n*,*

$$\Pr_{\substack{\mathsf{xiO} \\ x \leftarrow \{0,1\}^n}} [\mathsf{xiO.Eval}(\widetilde{C}, x) = C(x) : \widetilde{C} \leftarrow \mathsf{xiO.Obf}(C, 1^\lambda)] \geq 1 - \alpha(\lambda) \ .$$

  We say that $\mathsf{xiO.Obf}$ *is correct if* $\alpha(\lambda) \leq \mathrm{negl}(\lambda)$ *and approximately-correct if* $\alpha(\lambda) \leq 1/100$ *.*
- ***Non-trivial Efficiency:*** *there exists a constant* $\gamma < 1$ *and a fixed polynomial* $\mathrm{poly}(\cdot)$*, depending on the collection* $\boldsymbol{\mathcal{C}}$ *(but not on any specific class* $\mathcal{C} \in \boldsymbol{\mathcal{C}}$*), such that for any class* $\mathcal{C} \in \boldsymbol{\mathcal{C}}$ *security parameter* $\lambda \in \mathbb{N}$*, circuit* $C \in \mathcal{C}_\lambda$ *with input length* n*, and input* $x \in \{0,1\}^n$ *the running time of both* $\mathsf{xiO.Obf}(C, 1^\lambda)$ *and* $\mathsf{xiO.Eval}(\widetilde{C}, x)$ *is at most* $\mathrm{poly}(2^n, \lambda, |C|)$ *and the size of the obfuscated circuit* $\widetilde{C}$ *is at most* $2^{n\gamma} \cdot \mathrm{poly}(|C|, \lambda)$*. We call* γ *the compression factor, and say that the scheme is* γ*-compressing.*
- ***Indistinguishability:*** *for any* $\mathcal{C} = \{\mathcal{C}_\lambda\} \in \boldsymbol{\mathcal{C}}$ *and polynomial-size distinguisher* $\mathcal{D}$*, there exists a negligible function* $\mu(\cdot)$ *such that the following holds: for all security parameters* $\lambda \in \mathbb{N}$*, for any pair of circuits* $C_0, C_1 \in \mathcal{C}_\lambda$ *of the same size and such that* $C_0(x) = C_1(x)$ *for all inputs* x*,*

$$\left|\Pr\left[\mathcal{D}(\mathsf{xiO.Obf}(C_0, 1^\lambda)) = 1\right] - \Pr\left[\mathcal{D}(\mathsf{xiO.Obf}(C_1, 1^\lambda)) = 1\right]\right| \leq \mu(\lambda) \ .$$

  We further say that $\mathsf{xiO.Obf}$ *is* δ*-secure, for some concrete negligible function* $\delta(\cdot)$*, if for all polynomial-size distinguishers the above indistinguishability gap* $\mu(\lambda)$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Remark 1 (Logarithmic Input). Indeed, for XIO to be useful, we must restrict attention to circuit collections $\boldsymbol{\mathcal{C}} \subseteq \mathbf{P}^{\log}/\mathbf{poly}$. This ensures that obfuscation and evaluation are computable in time $2^{O(n)} = \mathrm{poly}(\lambda)$.

Remark 2 (Probabilistic $\mathsf{xiO.Eval}$*).* Above, we allow the evaluation algorithm $\mathsf{xiO.Eval}$ to be probabilistic. Throughout most of the paper, we restrict attention to *deterministic* evaluation algorithms. This typically will simplify exposition and is without loss of generality.

XIO with an Oracle. We say that an XIO scheme $\mathsf{xiO} = (\mathsf{xiO.Obf}, \mathsf{xiO.Eval})$ is constructed relative to an oracle $\mathcal{O}$ if the corresponding algorithms, as well as the adversary, may access the oracle $\mathcal{O}$. Namely, the obfuscation algorithm $\mathsf{xiO.Obf}^{\mathcal{O}}(C, 1^\lambda)$ and the evaluation algorithm $\mathsf{xiO.Eval}^{\mathcal{O}}(\widetilde{C}, x)$ are given oracle access to $\mathcal{O}$. In the security definition, the adversarial distinguisher $\mathcal{D}^{\mathcal{O}}$ also gets access to the oracle.

2.2 The Ideal Graded Encoding Model

The ideal graded-encoding model we consider is inspired by previous generic group and ideal graded-encoding models [44, 38, 17, 5] and is closest to the model of Pass and Shelat [40]. As in [40], we consider well-formed predicates that are determined by the validity of monomials.

Definition 3 (Well-Formed Validity Predicate). *V is well-formed if for any $d \in \mathbb{N}$ and degree-d polynomial $p(v_1, \ldots, v_m) = \sum_{i \le d, j_1, \ldots, j_i \in [m]} \rho_{j_1,\ldots,j_i} v_{j_1} \cdots v_{j_i}$, it holds that $V(p, \ell_1, \ldots, \ell_m) = \bigwedge_{i \le d, j_1, \ldots, j_i \in [m],\ \rho_{j_1,\ldots,j_i} \neq 0} V(\{\ell_{j_1}, \ldots, \ell_{j_i}\})$; namely, p is valid relative to the labels $\ell_1, \ldots, \ell_m$ if every monomial of p is valid relative to the corresponding multi-set of labels $\{\ell_{j_1}, \ldots, \ell_{j_i}\}$.*

We additionally consider the following decomposability requirement.

Definition 4 (Decomposable Validity Predicate). *V is* decomposable *if it is well-formed and there exist a projection function Π and a two-input predicate V_Π satisfying: For every two multisets $A = \{\ell_{1,1}, \ldots, \ell_{1,k_1}\}$ and $B = \{\ell_{2,1}, \ldots, \ell_{2,k_2}\}$ of labels, the validity of their union is given by*

$$V(A \uplus B) = V_\Pi(\Pi(A), \Pi(B)) \ .^{9}$$

The arity *of a decomposable predicate V is*

$$\mathsf{Arity}(V) := \max_A |\{\Pi(B) \ : \ V_\Pi(\Pi(A), \Pi(B)) = 1\}| \ ;$$

namely, it is the maximum number of projections $\Pi(B)$ that satisfy the validity predicate together with any given projection $\Pi(A)$, where A and B are multi-sets of labels.

Intuitively, a decomposable validity predicate has the property that any two different pairs of multi-sets $(A, B) \neq (A', B')$ share the same validity decision if they have the same projection $(\Pi(A), \Pi(B)) = (\Pi(A'), \Pi(B'))$. In other words, any information about the multi-sets beyond their projection does not matter. In the literature, all known ideal graded encoding models consider decomposable validity predicates with arity bounded by the degree (or even less). For instance, in set-based graded encodings, the labels correspond to subsets of some fixed universe $\mathbb{U}$, and a set of labels $\{S_1, \ldots, S_k \mid S_i \subseteq \mathbb{U}\}$ is valid if the sets are disjoint and $\biguplus S_i = \mathbb{U}$. Therefore, we can define the projection of any $A = \{S_1, \ldots, S_i\}$ to be $\Pi(A) = \biguplus S_i$ (or $\perp$ if the sets are not disjoint), in which case the arity is exactly one (indeed, for any $\Pi(A)$ only $\mathbb{U} \setminus \Pi(A)$ may satisfy the induced validity predicate).

We now formally define the ideal graded encoding model.

[9] For two multisets $A = \{a_1, \ldots, a_n\}, B = \{b_1, \ldots, b_m\}$, their union $A \uplus B = \{a_1, \ldots, a_n, b_1, \ldots, b_m\}$ counts multiplicity; e.g., $\{1, 1\} \uplus \{1, 2\} = \{1, 1, 1, 2\}$.

Definition 5 (Ideal Graded Encoding Oracle). *The oracle $\mathcal{M}_{\mathbb{F},V}$ is a stateful oracle, parameterized by a field $\mathbb{F}$ and a validity predicate V. The oracle answers queries of two forms:*

1. **Encoding Queries:** *Given a field element $\xi \in \mathbb{F}$ and label ℓ, the oracle samples a uniformly random string $r \leftarrow \{0,1\}^{\log|\mathbb{F}|}$, returns the handle $h = (r, \ell)$, and stores (h, ξ).*
2. **Zero-Test Queries:** *Given a polynomial $p \in \mathbb{F}[v_1, \ldots, v_m]$, and handles $h_1, \ldots, h_m$, the oracle does the following:*
   - *For each $i \in [m]$, obtains a tuple (h_i, ξ_i) from the stored list. If no such tuple exists, stops and returns* `false` .
   - *From each $h_i = (r_i, \ell_i)$, obtains ℓ_i, and checks that $V(p, \ell_1, \ldots, \ell_m) =$* `true` *to verify the query is valid and if not, returns* `false` .
   - *Performs a zero test, returning* `true` *if $p(\xi_1, \ldots, \xi_m) = 0$ and* `false` *otherwise.*

An ideal graded encoding oracle $\mathcal{M} = \{\mathcal{M}_{\mathbb{F}_\lambda, V_\lambda}\}$ is a collection of oracles $\mathcal{M}_{\mathbb{F}_\lambda, V_\lambda}$, one for each $\lambda \in \mathbb{N}$, where $|\mathbb{F}| = 2^{\Theta(\lambda)}$.

The oracle $\mathcal{M}$ is said to be degree-d, if for every polynomial p of degree $\deg(p) > d$, and any label vector $\boldsymbol{\ell}$, $V(p, \boldsymbol{\ell}) =$ `false` *. We say that an oracle $\mathcal{M}$ is decomposable if it has a decomposable validity predicate with bounded polynomial arity* $\mathrm{poly}(\lambda)$.

Remark 3. In some previous models (e.g., [40]), the ability to make encoding queries is further restricted. The above definition does not enforce any such restrictions. The results in this paper are presented in a public encoding model, which allows anyone to encode at any time. Our results on removing generic group oracle and random oracle from FE schemes can be extended to the model of private encodings, and the same holds for our results on reducing the degree of graded encoding oracles (Section Section 3), under certain mild assumptions. See the full version [8] for more details.

3 Reducing Constant-Degree Oracles to Bilinear Oracles

We show that any XIO scheme with a constant-degree decomposable ideal oracle can be transformed into an approximately-correct one with an ideal symmetric bilinear oracle (analogous to symmetric bilinear groups), provided that the XIO scheme is in a certain *product form*. We start by defining formally the notion of XIO in product form and of a symmetric bilinear oracle.

Definition 6 (Product Collection). $\mathcal{X} = \{\mathcal{X}_n\}_{n\in\mathbb{N}}, \mathcal{Y} = \{\mathcal{Y}_n\}_{n\in\mathbb{N}}$ *are said to be a product collection if:*

1. **Equal-Size Partition:** *For any* $X, X' \in \mathcal{X}_n$ *and* $Y, Y' \in \mathcal{Y}_n$*:*

$$|X| = |X'|, X \cap X' = \emptyset \qquad |Y| = |Y'|, Y \cap Y' = \emptyset \ ,$$

2. **Product Form:** *let* $\boldsymbol{X}_n = \biguplus_{X \in \mathcal{X}_n} X, \boldsymbol{Y}_n = \biguplus_{Y \in \mathcal{Y}_n} Y$ *then the input space* $\{0,1\}^n$ *factors:*

$$\{0,1\}^n \cong \boldsymbol{X}_n \times \boldsymbol{Y}_n \ .$$

Definition 7 (XIO in Product Form). *We say that an XIO scheme* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{\mathcal{O}}, \mathsf{xiO.Eval}^{\mathcal{O}})$*, relative to oracle* $\mathcal{O}$*, for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$*, is in* $(\mathcal{X}, \mathcal{Y})$*-product form for a product collection* $(\mathcal{X}, \mathcal{Y})$ *if:*

- *The obfuscation algorithm* $\mathsf{xiO.Obf}^{\mathcal{O}}$ *factors into two algorithms* $(\mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{X}}, \mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{Y}})$*, such that for any circuit* $C \in \boldsymbol{\mathcal{C}}$*,* $\mathsf{xiO.Obf}^{\mathcal{O}}(C, 1^\lambda; r)$*, outputs*

$$\left(\left\{ \widetilde{C}_X \leftarrow \mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{X}}(C, X, 1^\lambda; r) \right\}_{X \in \mathcal{X}_n}, \left\{ \widetilde{C}_Y \leftarrow \mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{Y}}(C, Y, 1^\lambda; r) \right\}_{Y \in \mathcal{Y}_n} \right) \ ,$$

  and all executions may use joint randomness r*.*
- *There is an evaluation algorithm* $\mathsf{xiO.Eval}^{\mathcal{O}}_{\mathcal{X},\mathcal{Y}}$ *such that for any* $(X, Y) \in \mathcal{X}_n \times \mathcal{Y}_n$*,*

$$\mathsf{xiO.Eval}^{\mathcal{O}}_{\mathcal{X},\mathcal{Y}}(\widetilde{C}_X, \widetilde{C}_Y) = \left(\mathsf{xiO.Eval}^{\mathcal{O}}(\widetilde{C}, (x, y)) \right)_{(x,y) \in X \times Y} \ .$$

Corresponding notation:

- *We denote by* $q_o^{\mathcal{X}} = q_o^{\mathcal{X}}(C, \lambda)$ *the maximal total size* $\sum_{Q \in \mathbf{Q}_o^X} |Q|$ *of all oracle queries* $\mathbf{Q}_o^X = \{Q\}$ *made by* $\mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{X}}(C, X, 1^\lambda)$ *when obfuscating an* n*-bit input circuit* $C \in \boldsymbol{\mathcal{C}}$ *for any* $X \in \mathcal{X}_n$*. Symmetrically, we denote by* $q_o^{\mathcal{Y}} = q_o^{\mathcal{Y}}(C, \lambda)$ *the bound on the total size* $\sum_{Q \in \mathbf{Q}_o^Y} |Q|$ *of oracle queries* $\mathbf{Q}_o^Y = \{Q\}$ *made by* $\mathsf{xiO.Obf}^{\mathcal{O}}_{\mathcal{Y}}(C, Y, 1^\lambda)$ *for any* $Y \in \mathcal{Y}_n$*.*

Definition 8 (Symmetric Bilinear Oracle). *The symmetric Bilinear Oracle* $\mathcal{B}^2 = \left\{ \mathcal{B}^2_{\mathbb{F}_\lambda, V} \right\}$ *is a special case of the ideal graded encoding oracle, where the validity predicate* V *is of degree two and is defined over a single label* $\ell_{\mathcal{B}}$*. That is,* $V(L) = \mathtt{true}$ *for a multiset of labels* L*, if and only if* $L \subseteq \{\ell_{\mathcal{B}}, \ell_{\mathcal{B}}\}$*.*

We now state the main theorem of this section.

Theorem 3. *Let* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{\mathcal{M}}, \mathsf{xiO.Eval}^{\mathcal{M}})$ *be an* $\mathsf{xiO.Obf}$ *scheme, relative to a degree-*d *decomposable ideal graded encoding oracle* $\mathcal{M}$*, for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$ *that is in* $(\mathcal{X}, \mathcal{Y})$*-product form, for some product collection* $(\mathcal{X}, \mathcal{Y})$*. Further assume that for some constant* $\gamma < 1$*,*

$$|\mathcal{X}_n| \cdot \left(q_o^{\mathcal{X}} \cdot \min\left(q_o^{\mathcal{X}}, |\mathcal{Y}_n| \cdot \log q_o^{\mathcal{X}} \right) \right)^d + |\mathcal{Y}_n| \cdot \left(q_o^{\mathcal{Y}} \cdot \min\left(q_o^{\mathcal{Y}}, |\mathcal{X}_n| \cdot \log q_o^{\mathcal{Y}} \right) \right)^d \leq 2^{\gamma n} \cdot \mathrm{poly}(|C|, \lambda) \ .$$

Then xiO *can be converted into an approximately-correct scheme* $\mathsf{xiO}^\star$ *relative to the symmetric bilinear oracle* $\mathcal{B}^2$.

Remark 4. A slightly easier to parse version of the above condition, with some loss in parameters, is that $|\mathcal{X}_n| \cdot \left(q_o^{\mathcal{X}}\right)^{2d} + |\mathcal{Y}_n| \cdot \left(q_o^{\mathcal{Y}}\right)^{2d} \leq 2^{\gamma n} \cdot \mathrm{poly}(|C|, \lambda)$.

Remark 5. Our ideal symmetric bilinear oracle captures symmetric bilinear pairing groups, but with two small gaps: Our oracle generates randomized encodings (following the Pass-shelat model) whereas bilinear pairing groups have unique encodings (of the form g^a), and our oracle does not support homomorphic opeartions whereas bilinear paring groups do. These differences are not consequential. In the full version of this paper [8], we show how to instantiate the transformed XIO schemes produced by the above theorem using concrete bilinear pairing groups.

Without Loss of Generality. Throughout this section, we make the following assumptions without loss of generality.

- **Obfuscator only encodes:** The XIO obfuscation algorithm only performs encoding queries and does not perform any zero tests. This is without loss of generality, as the obfuscator knows the field elements and labels underlying any generated handle (it encoded them itself), so zero-tests can be internally simulated.
- **Evaluator and adversary only zero-test:** The XIO evaluation algorithm as well as the adversary only perform zero tests and do not encode any elements themselves. Indeed, encoding of any (ξ, ℓ) can be internally simulated by sampling a corresponding handle $\widetilde{h}$. Then, whenever a zero-test $(p, h_1, \ldots, h_m, \widetilde{h}_1, \ldots, \widetilde{h}_{\widetilde{m}})$ includes such self-simulated handles $\widetilde{h}_i$, it is translated to a new zero test that does not include such handles, by hard-wiring the required field elements into the polynomial p.

3.1 Step 1: Explicit Handles

In this section, we show how to transform any XIO in product form relative to an ideal degree-d oracle (not necessarily decomposable) into one where all handles required for evaluation are given explicitly (also in product form). We start by defining the notion of explicit handles in product form and then state and describe the transformation.

Definition 9 (Explicit Handles in Product Form). *An XIO scheme* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{\mathcal{M}}, \mathsf{xiO.Eval}^{\mathcal{M}})$*, relative to an ideal graded encoding oracle, for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$*, is said to have explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product*

form, for a product collection $(\mathcal{X}, \mathcal{Y})$, if the obfuscation and evaluation algorithms satisfy the following structural requirement:

- *The algorithm* $\mathsf{xiO.Obf}^{\mathcal{M}}(C, 1^\lambda)$ *outputs* $\widetilde{C} = \left(\widetilde{Z}, \{\widetilde{H}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}_Y\}_{Y \in \mathcal{Y}_n}\right)$, *where each* $\widetilde{H}_X$ *and* $\widetilde{H}_{\mathcal{Y}}$ *are sets of handles generated by the oracle* $\mathcal{M}$ *during obfuscation, and* $\widetilde{Z}$ *is arbitrary auxiliary information.*
- *All* true *zero-test queries* $(p, h_1, \ldots, h_m)$ *— that is, zero-test queries that evaluate to* `true` *— made by the evaluation algorithm* $\mathsf{xiO.Eval}^{\mathcal{M}}\left(\widetilde{C}, (x, y)\right)$ *are such that for all* $j \in [m]$, $h_j \in \widetilde{H}_X \cup \widetilde{H}_Y$, *where* $(X, Y) \in \mathcal{X}_n \times \mathcal{Y}_n$ *are the (unique) sets such that* $(x, y) \in X \times Y$.

Corresponding notation:

- *We denote by* $q_h^{\mathcal{X}} = q_h^{\mathcal{X}}(C, \lambda)$ *the bound* $\max_{X \in \mathcal{X}_n} |\widetilde{H}_X|$ *on the maximum size of the set of explicit handles corresponding to any* $X \in \mathcal{X}_n$. *We denote by* $q_h^{\mathcal{Y}} = q_h^{\mathcal{Y}}(C, \lambda)$ *the bound on* $\max_{Y \in \mathcal{Y}_n} |\widetilde{H}_Y|$.

We show that any $\mathsf{xiO.Obf}$ scheme relative to an ideal graded encoding oracle that is in product form can be turned into one that has explicit handles in product form, but is approximately correct.

Lemma 1. *Let* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{\mathcal{M}}, \mathsf{xiO.Eval}^{\mathcal{M}})$ *be an* $\mathsf{xiO.Obf}$ *scheme, relative to an ideal graded encoding oracle* $\mathcal{M}$, *for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$, *that is in* $(\mathcal{X}, \mathcal{Y})$*-product form, for some product collection* $(\mathcal{X}, \mathcal{Y})$. *Then* xiO *can be converted into a new approximately-correct scheme* $\mathsf{xiO}^\star$ *with explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product form (relative to the same oracle* $\mathcal{M}$*).*

Furthermore, the size of the explicit handle sets are bounded as follows

$$q_h^{\mathcal{X}} \leq O\left(q_o^{\mathcal{X}}\right) \cdot \min\left(q_o^{\mathcal{X}}, |\mathcal{Y}_n| \cdot \log q_o^{\mathcal{X}}\right),\ q_h^{\mathcal{Y}} \leq O\left(q_o^{\mathcal{Y}}\right) \cdot \min\left(q_o^{\mathcal{Y}}, |\mathcal{X}_n| \cdot \log q_o^{\mathcal{Y}}\right)\ .$$

Our New XiO Scheme with Explicit Handles We now describe the new obfuscator xiO. We assume w.l.o.g that $q_o^{\mathcal{X}} \geq q_o^{\mathcal{Y}}$ (otherwise, the obfuscator reverses the roles of $\mathcal{X}, \mathcal{Y}$).

The Obfuscator $\mathsf{xiO}^\star.\mathsf{Obf}$: Given a circuit $C \in \boldsymbol{\mathcal{C}}$ with input size n, and security parameter 1^λ, $\mathsf{xiO}^\star.\mathsf{Obf}^{\mathcal{M}}(C, 1^\lambda)$ does the following:

- **Obfuscate:** Emulate the obfuscator $\mathsf{xiO.Obf}^{\mathcal{M}}(C, 1^\lambda)$ to obtain

  $$\left(\{\widetilde{C}_X \leftarrow \mathsf{xiO.Obf}^{\mathcal{M}}_{\mathcal{X}}(C, X, 1^\lambda)\}_{X \in \mathcal{X}_n}, \{\widetilde{C}_Y \leftarrow \mathsf{xiO.Obf}^{\mathcal{M}}_{\mathcal{Y}}(C, Y, 1^\lambda)\}_{Y \in \mathcal{Y}_n}\right)\ .$$

  For each $X \in \mathcal{X}_n$ store a list L_X of all tuples (h, ξ) such that $\mathsf{xiO.Obf}^{\mathcal{M}}_{\mathcal{X}}(C, X, 1^\lambda)$ requested the oracle $\mathcal{M}$ to encode (ξ, ℓ) and obtained back a handle $h = (r, \ell)$. Store a similar list L_Y for each execution $\mathsf{xiO.Obf}^{\mathcal{M}}_{\mathcal{Y}}(C, Y, 1^\lambda)$.

– **Learn Heavy Handles for $\mathcal{X}_n$:** for each $X \in \mathcal{X}_n$, let $\widetilde{H}_X = \emptyset$.
For $i \in \left\{1, \ldots, K_{\mathcal{X}} = \min\left(400q_o^{\mathcal{X}}, |\mathcal{Y}_n| \cdot \log\left(400q_o^{\mathcal{X}}\right)\right)\right\}$ do:
  - Sample a random $Y_i \leftarrow \mathcal{Y}_n$.
  - Emulate $\mathsf{xiO.Eval}^{(\cdot)}_{\mathcal{X},\mathcal{Y}}(\widetilde{C}_X, \widetilde{C}_{Y_i})$. To answer zero-test queries, emulate $\mathcal{M}$ using the lists (L_X, L_{Y_i}) constructed during the obfuscation phase.
  - In the process, for every zero-test query $(p, h_1, \ldots, h_m)$, if $\mathcal{M}(p, h_1, \ldots, h_m) = \texttt{true}$, namely it is a valid zero test and the answer is indeed zero, add $h_1, \ldots, h_m$ to $\widetilde{H}_X$.

  Store the resulting $\widetilde{H}_X$.
– **Learn Remaining Handles for $\mathcal{Y}_n$:** for each $Y \in \mathcal{Y}_n$, let $\widetilde{H}_Y = \emptyset$.
For $i \in \left\{1, \ldots, K_{\mathcal{Y}} = \min\left(200q_o^{\mathcal{Y}}, |\mathcal{X}_n| \cdot \log\left(200q_o^{\mathcal{Y}}\right)\right)\right\}$ do the following:
  - Sample a random $X_i \leftarrow \mathcal{X}_n$, and let $\widetilde{H}_{X_i,Y} = \emptyset$.
  - Emulate $\mathsf{xiO.Eval}^{(\cdot)}_{\mathcal{X},\mathcal{Y}}(\widetilde{C}_{X_i}, \widetilde{C}_Y)$. To answer zero-test queries, emulate $\mathcal{M}$ using the lists (L_{X_i}, L_Y) constructed during the obfuscation phase.
  - In the process, for every zero-test query $(p, h_1, \ldots, h_m)$, if $\mathcal{M}(p, h_1, \ldots, h_m) = \texttt{true}$, namely it is a valid zero test and the answer is indeed zero, add $h_1, \ldots, h_m$ to $\widetilde{H}_{X_i,Y}$.
  - Remove from $\widetilde{H}_{X_i,Y}$ all handles in $\widetilde{H}_{X_i}$.
  - If $|\widetilde{H}_{X_i,Y}| \leq q_o^{\mathcal{Y}}(C, \lambda)$, add $\widetilde{H}_{X_i,Y}$ to $\widetilde{H}_Y$. Otherwise discard $\widetilde{H}_{X_i,Y}$.

  Store the resulting $\widetilde{H}_Y$.
– **Output:**

$$\widetilde{C}^\star = (\widetilde{Z}, \{\widetilde{H}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}_Y\}_{Y \in \mathcal{Y}_n}), \text{ where } \widetilde{Z} = (\{\widetilde{C}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{C}_Y\}_{Y \in \mathcal{Y}_n}) \ .$$

The Evaluator $\mathsf{xiO}^\star.\mathsf{Eval}$**:** Given an obfuscation $\widetilde{C}^\star = (\widetilde{C}, \{\widetilde{H}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}_Y\}_{Y \in \mathcal{Y}_n})$, $(x, y) \in \boldsymbol{X}_n \times \boldsymbol{Y}_n$, $\mathsf{xiO}^\star.\mathsf{Eval}^{\mathcal{M}}(\widetilde{C}^\star, (x, y))$ does the following:

– Let $(X, Y) \in \mathcal{X}_n \times \mathcal{Y}_n$ be the (unique) sets such that $(x, y) \in X \times Y$.
– Emulate $\mathsf{xiO.Eval}^{(\cdot)}_{\mathcal{X},\mathcal{Y}}(\widetilde{C}_X, \widetilde{C}_Y)$.
– Whenever $\mathsf{xiO.Eval}$ makes a zero-test query $(p, h_1, \ldots, h_m)$:
  - If for some i, $h_i \notin \widetilde{H}_X \cup \widetilde{H}_Y$, answer $\texttt{false}$.
  - Forward any other zero-test to the oracle $\mathcal{M}$ and return its answer.

In the full version [8], we show that the new obfuscator is approximately correct, secure, and efficient as stated in Lemma 1.

3.2 Step 2: From Constant-Degree to Degree Two

We show that any XIO scheme with explicit handles in product form, relative to a degree-d decomposable ideal oracle (for arbitrary $d = O(1)$), can be transformed into one relative to a degree-2 decomposable ideal oracle. The resulting

degree-2 oracle is defined with respect to a validity predicate V^2 related to the validity predicate V^d of the degree-d oracle we start with.

Intuitively, this model can be seen as an extension of the standard asymmetric bilinear maps, where instead of two base groups we may have more. That is, instead of two asymmetric base-groups G_1, G_2 where $(g_1^a, g_2^b) \in G_1 \times G_2$ can be mapped to $e(g_1, g_2)^{ab}$ in the target group G_T, we possibly have a larger number of groups $G_1, \ldots, G_n$ and a collection of *valid mappings* $\{e_k : G_{i_k} \times G_{j_k} \to G_T\}$, which may be a strict subset of all possible bilinear maps.

Lemma 2. *let* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{(\cdot)}, \mathsf{xiO.Eval}^{(\cdot)})$ *be an XIO scheme, for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$*, defined relative to a degree-*d *decomposable ideal oracle* $\mathcal{M}^d = \{\mathcal{M}^d_{\mathbb{F}_\lambda, V_\lambda}\}$*, with explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product form, for some product collection* $(\mathcal{X}, \mathcal{Y})$*. Assume further that for some constant* $\gamma < 1$*,*

$$|\mathcal{X}_n| \cdot \left(q_h^{\mathcal{X}}\right)^d + |\mathcal{Y}_n| \cdot \left(q_h^{\mathcal{Y}}\right)^d \leq 2^{\gamma n} \cdot \mathrm{poly}(|C|, \lambda) \ .$$

Then xiO *can be converted to a new scheme* $\mathsf{xiO}^\star$*, also with explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product form, relative to a degree-*2 *decomposable oracle* $\mathcal{M}^2$*.*

We now present our new XiO scheme relative to a degree-2 decomposable oracle; see the full version for its analysis.

The New XiO Scheme Relative to a Degree-2 Oracle $\mathcal{M}^2$ In what follows, let $\mathsf{xiO} = (\mathsf{xiO.Obf}^{(\cdot)}, \mathsf{xiO.Eval}^{(\cdot)})$ be an XIO scheme with explicit handles in product form, defined relative to a degree-d decomposable ideal oracle $\mathcal{M}^d = \{\mathcal{M}^d_{\mathbb{F}_\lambda, V_\lambda}\}$. We describe a new scheme $\mathsf{xiO}^\star = (\mathsf{xiO}^\star.\mathsf{Obf}^{(\cdot)}, \mathsf{xiO}^\star.\mathsf{Eval}^{(\cdot)})$ (also, with explicit handles in product form) defined relative to a degree-2 decomposable ideal oracle $\mathcal{M}^2 = \{\mathcal{M}^2_{\mathbb{F}_\lambda, V^\star_\lambda}\}$.

The Obfuscator $\mathsf{xiO}^\star.\mathsf{Obf}$**:** Given a circuit $C \in \boldsymbol{\mathcal{C}}$ with input size n, and security parameter 1^λ, and oracle access to $\mathcal{M}^2$, $\mathsf{xiO}^\star.\mathsf{Obf}^{\mathcal{M}^2}(C, 1^\lambda)$ does as follows:

- **Emulate Obfuscation:**
  - Emulate $\mathsf{xiO.Obf}^{\mathcal{M}^d}(C, 1^\lambda)$.
  - Throughout the emulation, emulate the oracle $\mathcal{M}^d$, storing a list $L = \{(h, \xi)\}$ of encoded element-label pairs (ξ, ℓ) and corresponding handles $h = (r, \ell)$.
  - Obtain the obfuscation $(\widetilde{Z}, \{\widetilde{H}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}_X\}_{Y \in \mathcal{Y}_n})$.
- **Encode Monomials:**
  - For each $X \in \mathcal{X}_n$:
    1. Retrieve $\widetilde{H}_X = (h_1, \ldots, h_m)$ and the corresponding field elements and labels $(\xi_1, \ell_1), \ldots, (\xi_m, \ell_m)$ from the stored list L.

2. For every formal monomial $\Phi(v_1, \ldots, v_m) = v_{i_1} \ldots v_{i_j}$, where $j \leq d$ and $i_1, \ldots, i_j \in [m]$, compute

$$\Phi(\boldsymbol{\xi}) := \xi_{i_1} \cdots \xi_{i_j},\ \Phi(\boldsymbol{\ell}) := \left\{\ell_{i_1}, \ldots, \ell_{i_j}\right\},\ \Phi(\boldsymbol{h}) := \left\{h_{i_1}, \ldots, h_{i_j}\right\}\ .$$

(For simplicity of notation, we overload Φ to describe different functions when acting on field elements, labels, and handles.) Then, request $\mathcal{M}^2$ to encode the field element and label $(\xi^\star_{X,\Phi}, \ell^\star_{X,\Phi}) := (\Phi(\boldsymbol{\xi}), \Phi(\boldsymbol{\ell}))$, and obtain a handle $h^\star_{X,\Phi}$.

3. Store $\widetilde{H}^\star_X = \left\{(h^\star_{X,\Phi}, \Phi(\boldsymbol{h}))\right\}_\Phi$

- For each $Y \in \mathcal{Y}_n$:
  1. Symmetrically perform the above two steps with respect to $\widetilde{H}_Y$ (instead of $\widetilde{H}_X$).
  2. Store $\widetilde{H}^\star_Y = \left\{(h^\star_{Y,\Phi}, \Phi(\boldsymbol{h}))\right\}_\Phi$.

– **Output:**

$$\widetilde{C}^\star = (\widetilde{C}, \{\widetilde{H}^\star_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}^\star_Y\}_{Y \in \mathcal{Y}_n}),\ \text{where}\ \widetilde{C} := (\widetilde{Z}, \{\widetilde{H}_X\}_X, \{\widetilde{H}_Y\}_Y)\ .$$

The Evaluator $\mathsf{xiO}^\star.\mathsf{Eval}$**:** Given an obfuscation $\widetilde{C}^\star = (\widetilde{C}, \{\widetilde{H}^\star_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}^\star_Y\}_{Y \in \mathcal{Y}_n})$, input $(x, y) \in \boldsymbol{X}_n \times \boldsymbol{Y}_n$, and oracle $\mathcal{M}^2, \mathsf{xiO}^\star.\mathsf{Eval}^{\mathcal{M}^2}(\widetilde{C}^\star, (x, y))$ does the following:

– Emulate $\mathsf{xiO}.\mathsf{Eval}^{\mathcal{M}^d}(\widetilde{C}, (x, y))$.
– Emulate any zero-test query $(p, h_1, \ldots, h_m)$ it makes to $\mathcal{M}^d$ as follows:
  1. Parse $\widetilde{C} = (\widetilde{Z}, \{\widetilde{H}_X\}_{X \in \mathcal{X}_n}, \{\widetilde{H}_Y\}_{Y \in \mathcal{Y}_n})$.
  2. Let $(X, Y) \in \mathcal{X}_n \times \mathcal{Y}_n$ be the (unique) sets such that $(x, y) \in X \times Y$. Retrieve $\widetilde{H}_X, \widetilde{H}_Y$.
  3. Split $\boldsymbol{h} = (h_1, \ldots, h_m)$ into two verctors of handles $\boldsymbol{h}_X \subseteq \widetilde{H}_X$ and $\boldsymbol{h}_Y \subseteq \widetilde{H}_Y$. (Such a partition always exists, by the guarantee of explicit handles in product form.)
  4. Viewing $p(\boldsymbol{h})$ as a formal polynomial in variables $\boldsymbol{h}$, factor it as

$$p(\boldsymbol{h}) = \sum_i \gamma_i \Phi_i(\boldsymbol{h}) = \sum_i \gamma_i \Phi_{X,i}(\boldsymbol{h}_X) \Phi_{Y,i}(\boldsymbol{h}_Y)\ ,$$

where $\gamma_i \in \mathbb{F} \setminus \{0\}$ are the coefficients, and each monomial $\Phi_i(\boldsymbol{h})$ is factored into $\Phi_{X,i}(\boldsymbol{h}_X) \cdot \Phi_{Y,i}(\boldsymbol{h}_Y)$.

  5. Translate $\{\Phi_{X,i}(\boldsymbol{h}_X), \Phi_{Y,i}(\boldsymbol{h}_Y)\}_i$ into handles $\{h^\star_{X,i}, h^\star_{Y,i}\}_i$ by locating $(h^\star_{X,i}, \Phi_{X,i}(\boldsymbol{h}_X)) \in \widetilde{H}^\star_X$ and $(h^\star_{Y,i}, \Phi_{Y,i}(\boldsymbol{h}_Y)) \in \widetilde{H}^\star_Y$.

6. Consider the degree-2 formal polynomial:

$$p^\star(\boldsymbol{h}^\star) = \sum_i \gamma_i h^\star_{X,i} h^\star_{Y,i} \ .$$

7. Make the zero-test $(p^\star, \boldsymbol{h}^\star)$ to the oracle $\mathcal{M}^2$ and return the result.

Labels and Validity Predicate V^2 of Oracle $\mathcal{M}^2$. Note that labels with respect to $\mathcal{M}^2$ are subsets of the label set of $\mathcal{M}$. Let V^d be the decomposable validity predicate associated with $\mathcal{M}^d$. We define a new validity predicate of degree 2, which is also decomposable. For this purpose, we need to define V^2 for labels corresponding to bilinear monomials given by a multi-set $\{\ell^\star_1, \ell^\star_2\}$. For all other multi-sets L (with cardinality larger than 2), $V^2(L) = \texttt{false}$, capturing that this is a degree 2-predicate.

The validity predicate $V^2(\{\ell^\star_1, \ell^\star_2\})$ is computed as follows:

- Parse $\ell^\star_1$ and $\ell^\star_2$ as as two multi-sets $\{\ell_{1,1}, \ldots, \ell_{1,k_1}\}, \{\ell_{2,1}, \ldots, \ell_{2,k_2}\}$.
- Apply the original predicate to the disjoint union multi-set:

$$V^2(\{\ell^\star_1, \ell^\star_2\}) := V^d(\ell^\star_1 \uplus \ell^\star_2) = V^d(\{\ell_{1,1}, \ldots, \ell_{1,k_1}\} \uplus \{\ell_{2,1}, \ldots, \ell_{2,k_2}\}) \ .$$

Recall that the fact that V^d is decomposable means that there exist a projection function Π^d and predicate V^d_Π, such that, for every two multi-sets $\ell^\star_1, \ell^\star_2$, $V^d(\ell^\star_1 \uplus \ell^\star_2) = V^d_\Pi(\Pi^d(\ell^\star_1), \Pi^d(\ell^\star_2))$. We show that V^2 is also decomposable, by defining its corresponding projection function Π^2 and predicate V^2_Π, and showing that on input two multisets $A = \{\ell^\star_i\}_i$ and $B = \{\ell^\star_j\}_j$, $V^2(A \uplus B) = V^2_\Pi(\Pi^2(A), \Pi^2(B))$. The projection function Π^2 on input a multiset A computes: $\Pi^2(A) = \big(|A|, \Pi^d(\uplus_{\ell^\star \in A} \ell^\star)\big)$. The predicate V^2_Π on input two multisets A, B outputs $\texttt{false}$ if $|A| + |B| > 2$. Otherwise, if A, B contain exactly two labels $\ell^\star_1, \ell^\star_2$, the predicate computes:

$$\begin{aligned} V^2_\Pi(\Pi^2(A), \Pi^2(B)) &= V^d(\Pi^d(\uplus_{\ell^\star \in A} \ell^\star), \Pi^d(\uplus_{\ell^\star \in B} \ell^\star)) \\ &= V^d((\uplus_{\ell^\star \in A} \ell^\star) \uplus (\uplus_{\ell^\star \in B} \ell^\star)) = V^d(\ell^\star_1 \uplus \ell^\star_2) = V^2(A \uplus B) \end{aligned}$$

Therefore V^2 is decomposable. Moreover, it is easy to see that the arity of V^2 is exactly that of V^d, which is bounded by a fixed polynomial.

3.3 Step 3: Asymmetric Oracles to Symmetric Oracles

We show that any XIO scheme with explicit handles relative to the oracle $\mathcal{M}^2$ can be converted to a scheme relative to a symmetric bilinear oracle $\mathcal{B}^2$ (also with explicit handles). This model is analogous to the symmetric bilinear pairing groups where there is a single base group G with a bilinear map $e : G \times G \to G_T$ (Definition 8). The transformation will incur a certain blowup depending on the arity of the oracle $\mathcal{M}^2$, which is a bounded polynomial.

Lemma 3. *let* $\mathsf{xiO} = (\mathsf{xiO.Obf}^{(\cdot)}, \mathsf{xiO.Eval}^{(\cdot)})$ *be an XIO scheme, for a collection of circuit classes* $\boldsymbol{\mathcal{C}}$*, defined relative to the (asymmetric) decomposable oracle* $\mathcal{M}^2$*, with explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product form, for some product collection* $(\mathcal{X}, \mathcal{Y})$*. Then* xiO *can be converted to a new scheme* $\mathsf{xiO}^\star$ *relative to the (symmetric) oracle* $\mathcal{B}^2$*, also with explicit handles in* $(\mathcal{X}, \mathcal{Y})$*-product form.*

Towards the lemma, we show a transformation that reduces the oracle $\mathcal{M}^2$ to a symmetric bilinear oracle $\mathcal{B}^2$. In the full version [8], we use this transformation to convert any XiO scheme relative to $\mathcal{M}^2$ to one relative to $\mathcal{B}^2$.

Reducing Oracle $\mathcal{M}^2$ to Oracle $\mathcal{B}^2$ The transformation consists of a recoding process $\mathcal{E}$ that takes a secret key K, and an arbitrary encoding query of the form (ξ, ℓ) to $\mathcal{M}^2$, and transforms it into a set of new encoding queries $(\xi_1^\star, \ell_{\mathcal{B}}), \ldots, (\xi_k^\star, \ell_{\mathcal{B}})$ which it gives $\mathcal{B}^2$ (all with respect to the unique label $\ell_{\mathcal{B}}$). $\mathcal{E}$ then outputs a handle $\boldsymbol{h}$ representing (ξ, ℓ) consisting of a list of handles $\boldsymbol{h} = (h_1^\star, \ldots, h_k^\star)$ generated by $\mathcal{B}^2$ for $\xi_1^\star, \ldots, \xi_k^\star$.

The encoder $\mathcal{E}$ is associated with a (public) decoder $\mathcal{D}$. The decoder $\mathcal{D}$ is given as input a zero-test query $(p, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_m)$ for $\mathcal{M}^2$ to be evaluated over underlying field elements $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m)$, and now represented by $\boldsymbol{\xi}^\star = (\xi_{1,1}^\star, \ldots, \xi_{1,k}^\star, \ldots, \xi_{m,1}^\star, \ldots, \xi_{m,k}^\star)$ encoded in $\mathcal{B}^2$ with handles $\boldsymbol{h}^\star = (h_{1,1}^\star, \ldots, h_{1,k}^\star, \ldots, h_{m,1}^\star, \ldots, h_{m,k}^\star)$. The decoder then translates it into a new zero-test query $(p^\star, \boldsymbol{h}^\star)$ and submits it to $\mathcal{B}^2$, with the guarantee that if the zero test is valid with respect to the validity predicate V associated with $\mathcal{M}^d$, then $p(\boldsymbol{\xi}) = p^\star(\boldsymbol{\xi}^\star)$, and otherwise, $p^\star(\boldsymbol{\xi}^\star)$ evaluates to non-zero with overwhelming probability.

We next turn to a more formal description of the transformation. In what follows, let V be an arbitrary degree-2 decomposable validity predicate, defined over pairs of labels $(\ell, \ell') \in \mathbb{L} \times \mathbb{L}$ from a label set $\mathbb{L}$, and associated with projection function $\mathit{\Pi}$ and predicate $V_{\mathit{\Pi}}$ with bounded arity $\mathsf{Arity}(V_{\mathit{\Pi}}) \leq \mathrm{poly}(\lambda)$.

Secret Encoding Key. The secret key K consists of random invertible field elements $\eta_\ell, \varphi_\ell \leftarrow \mathbb{F} \setminus \{0\}$ for each label $\ell \in \mathbb{L}$, and random invertible field elements $\alpha_\pi, \beta_\pi, \gamma_\pi, \delta_\pi \leftarrow \mathbb{F} \setminus \{0\}$ for every π in the corresponding set of projections $\mathit{\Gamma} = \{\mathit{\Pi}(\{\ell\}) \,:\, \ell \in \mathbb{L}\}$.

Remark 6 (Lazy Secret-Key Sampling). Note that the total number of labels and their projection could be superpolynomial, making the secret key superpolynomial in length. To deal with such cases, the recoder uses lazy sampling to sample the above random invertible elements only when needed and keeps a record of all sampled elements. As we argue below, the total number of random invertible elements to be sampled is polynomial in the number of tuples (ξ, ℓ) to be

recoded. For simplicity of exposition, we describe the procedure with respect to a key consisting of all possible random invertible elements.

Recoding. Given the secret key K and $(\xi, \ell) \in \mathbb{F} \times \mathbb{L}$, the encoder $\mathcal{E}^{\mathcal{B}^2}((\xi, \ell), K)$ does the following:

- Samples two secret shares ξ_L, ξ_R at random from $\mathbb{F}$ subject to $\xi_L + \xi_R = \xi$.
- Let $\pi = \Pi(\{\ell\})$ be the projection of $\{\ell\}$. Generates the field elements:

$$\boldsymbol{\xi}^\star_\circ := \left(\xi^\star_{\circ,\alpha,L} = \alpha_\pi \cdot \xi_L,\ \xi^\star_{\circ,\beta,R} = \beta_\pi \cdot \xi_R,\ \xi^\star_{\circ,\gamma,L} = \gamma_\pi \cdot \xi_L,\ \xi^\star_{\circ,\delta,R} = \delta_\pi \cdot \xi_R\right) .$$

- Let $\mathsf{match}(\pi) = \{\pi' : V_\Pi(\pi, \pi') = \mathtt{true}\}$ be the set of projections that evaluates to $\mathtt{true}$ with π. (For every $\pi' \in \mathsf{match}(\pi)$, and every ℓ', such that, $\pi' = \Pi(\{\ell'\})$, it holds that $V(\{\ell, \ell'\}) = \mathtt{true}$.)
  For each $\pi' \in \mathsf{match}(\pi)$, generates the field elements:

$$\boldsymbol{\xi}^\star_{\pi'} := \Big(\xi^\star_{\pi',\frac{1}{\alpha},L} = \frac{1}{\alpha_\pi} \cdot \xi_L, \quad \xi^\star_{\pi',\frac{1}{\beta},L} = \frac{1}{\beta_\pi} \cdot \xi_L, \quad \xi^\star_{\pi',\frac{1}{\gamma},R} = \frac{1}{\gamma_\pi} \cdot \xi_R, \quad \xi^\star_{\pi',\frac{1}{\delta},R} = \frac{1}{\delta_\pi} \cdot \xi_R\Big) .$$

- If $V(\{\ell\}) = \mathtt{true}$, generates field elements

$$\boldsymbol{\xi}^\star_\triangle := \left(\xi^\star_{\triangle,\eta,L} = \eta_\ell \cdot \xi_L, \quad \xi^\star_{\triangle,\frac{1}{\eta}} = \frac{1}{\eta_\ell}, \quad \xi^\star_{\triangle,\varphi,R} = \varphi_\ell \cdot \xi_R, \quad \xi^\star_{\triangle,\frac{1}{\varphi}} = \frac{1}{\varphi_\ell}\right) ,$$

- Asks $\mathcal{B}^2$ to encode (with respect to the unique label $\ell_\mathcal{B}$) the field elements $\boldsymbol{\xi}^\star_\circ, (\boldsymbol{\xi}^\star_{\pi'})_{\pi' \in \mathsf{match}(\pi)}, \boldsymbol{\xi}^\star_\triangle$ generated above, obtaining corressponding handles

$$\boldsymbol{h}^\star = \left(\boldsymbol{h}^\star_\circ, (\boldsymbol{h}^\star_{\pi'})_{\pi' \in \mathsf{match}(\pi)}, \boldsymbol{h}^\star_\triangle\right) .$$

- Outputs handles $\boldsymbol{h}^\star$.

We argue that when V has bounded $\mathrm{poly}(\lambda)$ arity, the size of the new encoding $\boldsymbol{h}^\star$ is bounded by $\mathrm{poly}(\lambda)$. This is because, $\boldsymbol{h}^\star_\circ$ and $\boldsymbol{h}^\star_\triangle$ each consists of 4 encodings, while $(\boldsymbol{h}^\star_{\pi'})_{\pi' \in \mathsf{match}(\pi)}$ consists of $O(|\mathsf{match}(\pi)|) = \mathsf{Arity}(V_\Pi) \leq \mathrm{poly}(\lambda)$.

Decoding. Given a degree-2 polynomial p and handles $(\boldsymbol{h}^\star_1, \ldots, \boldsymbol{h}^\star_m)$, where $\boldsymbol{h}^\star_i = \boldsymbol{h}^\star_{i,\circ}, \left(\boldsymbol{h}^\star_{i,\pi'}\right)_{\pi' \in \mathsf{match}(\pi)}, \boldsymbol{h}^\star_{i,\triangle}$ the decoder $\mathcal{D}^{\mathcal{B}^2}(p, \boldsymbol{h}^\star_1, \ldots, \boldsymbol{h}^\star_m)$:

- Writes p as a formal polynomial

$$p(\boldsymbol{h}^\star_1, \ldots, \boldsymbol{h}^\star_m) = \sigma + \sum_k \rho_k \boldsymbol{h}^\star_k + \sum_{i \leq j} \rho_{i,j} \boldsymbol{h}^\star_i \boldsymbol{h}^\star_j .$$

– If for any monomial $\boldsymbol{h}_k^\star$ in p, $V(\{\ell_k\}) = \texttt{false}$, or for any monomial $\boldsymbol{h}_i^\star \boldsymbol{h}_j^\star$, $V(\{\ell_i, \ell_j\}) = \texttt{false}$, return $\texttt{false}$. Otherwise, continue.
– Generates a new degree-2 formal polynomial

$$p^\star(\boldsymbol{h}^\star) = \sigma + \sum_k \rho_k \cdot \left(h^\star_{k,\Delta,\eta,L} h^\star_{k,\Delta,\frac{1}{\eta}} + h^\star_{k,\Delta,\varphi,R} h^\star_{k,\Delta,\frac{1}{\varphi}} \right) +$$

$$\sum_{i \le j} \rho_{i,j} \cdot \left(h^\star_{i,\circ,\alpha,L} h^\star_{j,\pi_i,\frac{1}{\alpha},L} + h^\star_{i,\circ,\gamma,L} h^\star_{j,\pi_i,\frac{1}{\gamma},R} + h^\star_{i,\circ,\beta,R} h^\star_{j,\pi_i,\frac{1}{\beta},L} + h^\star_{i,\circ,\delta,R} h^\star_{j,\pi_i,\frac{1}{\delta},R} \right) \ .$$

– It submits to $\mathcal{B}^2$ the zero test $(p^\star, \boldsymbol{h}^\star)$ and returns the result.

3.4 Putting it All Together

We conclude the proof of Theorem 3.

Proof (of Theorem 3). To obtain $\mathsf{xiO}^\star$, we apply to xiO Lemmas 1, 2, 3.

– Lemma 1 turns xiO into an approximately-correct XIO scheme xiO_1 with explicit handles, relative to the same degree-d decomposable oracle $\mathcal{M}^d$ that xiO uses.
– Lemma 2 turns xiO_1 into an approximately-correct XiO scheme xiO_2 with explicit handles, relative to an asymmetric bilinear oracle $\mathcal{M}^2$ that is also decomposable.
– Lemma 3 turns xiO_2 into an approximately-correct XiO scheme xiO_3 with explicit handles, relative to a symmetric bilinear oracle $\mathcal{B}^2$.

The final XiO scheme xiO_3 is exactly the new XiO scheme $\mathsf{xiO}^\star$. By composing the three lemmas, we have that $\mathsf{xiO}^\star$ is approximately correct and secure. The only thing to argue that $\mathsf{xiO}^\star$ is also weakly succinct. Note that the obfuscated circuits of $\mathsf{xiO}^\star$ have the form

$$\widetilde{C} = \left(\widetilde{Z}, \{\widetilde{H}_X\}, \{\widetilde{H}_Y\}, \ \{\widetilde{H}^\star_X\}, \{\widetilde{H}^\star_Y\}, \ \{\widetilde{H}'_X\}, \{\widetilde{H}'_Y\} \right)$$

where $\widetilde{Z}$ is an obufscated circuit of the original scheme xiO, $\widetilde{H}_X$ and $\widetilde{H}_Y$ are the sets of explicit handles of $\mathcal{M}^d$ added by Lemma 1, $\widetilde{H}^\star_X$ and $\widetilde{H}^\star_Y$ are the encodings of monomials of $\mathcal{M}^2$ added by Lemma 2, $\widetilde{H}'_X$ and $\widetilde{H}'_Y$ are the re-encodings of $\mathcal{B}^2$ added by Lemma 3. By the three lemmas and the fact that the original scheme xiO is $\gamma^\star$-compressing and satisfies the efficiency requirement stated in Theorem 3, we have,

$$\begin{aligned}
|\widetilde{C}| &\le |\widetilde{Z}| + O\left(\left| \{\widetilde{H}'_X\}, \{\widetilde{H}'_Y\} \right| \right) \\
&\le 2^{\gamma^\star n} \mathrm{poly}(\lambda, |C|) + \Big(|\mathcal{X}_n| \cdot \left(q_o^{\mathcal{X}} \cdot \min\left(q_o^{\mathcal{X}}, |\mathcal{Y}_n| \cdot \log q_o^{\mathcal{X}} \right) \right)^d \\
&\qquad\qquad + |\mathcal{Y}_n| \cdot \left(q_o^{\mathcal{Y}} \cdot \min\left(q_o^{\mathcal{Y}}, |\mathcal{X}_n| \cdot \log q_o^{\mathcal{Y}} \right) \right)^d \Big) \cdot \mathrm{poly}(\lambda) \\
&\le (2^{\gamma^\star n} + 2^{\gamma n}) \cdot \mathrm{poly}(\lambda, |C|) \le 2^{\gamma' n} \cdot \mathrm{poly}(\lambda, |C|) \ ,
\end{aligned}$$

for some $\gamma' < 1$. Thus,the new XIO scheme is weakly succinct.

4 From (Approximate) XIO and LWE to FE

We describe at a high-level how to use approximate XIO to construct 1-key weakly succinct FE for $\mathbf{P}/\mathbf{poly}$, assuming LWE. The formal transformation can be found in the full version of this paper [8].

Theorem 4. *Assuming LWE with subexponential modulus-to-noise ratio and the existence of an approximate XIO scheme for $\boldsymbol{P}^{\log}/\boldsymbol{poly}$, there exists a single-key weakly-succinct FE scheme FE for $\boldsymbol{P}/\boldsymbol{poly}$.*

A Failed Attempt. Lin, Pass, Seth and Telang [34] showed a transformation from correct XIO for $\mathbf{P}^{\log}/\mathbf{poly}$ to IO for $\mathbf{P}/\mathbf{poly}$, assuming LWE.[10] Previously, Bitansky and Vaikuntanathan [12] showed how to make any approximately correct IO correct (assuming, say, LWE). Thus, to prove the above theorem, a natural idea is to amplify the correctness of approximate XIO to obtain correct XIO by [12], and then invoke the transformation of [34]. This approach turns out to completely fail. Indeed, the [12] transformation only works for classes of circuits that are expressive enough; in particular, it relies on the ability of circuits in the class to process encrypted inputs, which must inherently be of super-logarithmic length in the security parameter. However, XIO for such circuit classes, which lie outside of $\mathbf{P}^{\log}/\mathbf{poly}$, is inefficient (see Remark 1).

Instead, we show how to modify the transformation of [34], based on error-correcting codes, so that, it works directly with approximate XIO. Below, we briefly review the [34] transformation and describe our key ideas.

Review of the [34] Transformation. Goldwassar et al. [28] constructed, from LWE with subexponential modulus-to-noise ratio, a fully succinct, public-key, single-key, FE scheme for *Boolean* $\mathbf{NC}^1$ circuits; namely, the encryption circuit of their scheme has size $\mathrm{poly}(n, \lambda)$, where n is the message length.

Starting from such an FE scheme bFE for Boolean circuits, the first observation in [34] is as follows: To construct an FE scheme, FE for any (possibly non-Boolean) circuit C, one can use bFE to issue a key for the corresponding Boolean circuit B that produces *one output bit at a time*, that is, $B(m, i) = (C(m))_i$. Then to enable evaluating the circuit C, it suffices to publish a list of bFE ciphertexts encrypting all pairs (m, i). This, however, leads to a scheme with encryption time linear in the length of the output (as it needs to produce a ciphertext for every output bit), and is not weakly succinct. The key

[10] The LWE assumption was later weakened to the existence of public key encryption by [9], but only for sufficiently-compressing XIO.

idea in [34] is using XIO to generate the list of encrypted pairs (m, i). Namely, obfuscate a circuit that given as input i, outputs the encryption of (m, i), where randomness is derived with a pseudorandom function. Since XIO achieves "sub-linear compression", the resulting FE scheme is now weakly succinct for all of $\mathbf{NC}^1$, including circuits with non-Boolean output.

Our Approach. The basic idea behind replacing XIO with approximate XIO is to use good error-correcting codes to allow recovering the output of a given function even if some of the encryptions (m, i) are faulty. Specifically, we make the following modification to the transformation of [34]. Instead of deriving a key for the Boolean function $B(m, i) = (C(m))_i$, which computes the i-th bit of the circuit's output, we consider the function $B^\star(m, i) = (\mathsf{ECC}(C(m)))_i$ that outputs the i-th bit of an error-corrected version of this output. As before, we use XIO to to generate the list of encryptions (m, i), only that now, with approximate XIO, some of these encryptions may be faulty. Nevertheless, we can still recover $(\mathsf{ECC}(C(m)))_i$ for a large enough fraction of indices i, and can thus correct, and obtain $C(m)$. By using codes with constant rate, and a linear-size constant-depth encoding circuit, we can show that this transformation achieves the required compression.

Acknowledgements: We thank V. Vaikuntanathan for enlightening discussions.

References

1. Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 308–326, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.
2. Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption. *IACR Cryptology ePrint Archive*, 2015:730, 2015.
3. Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. *IACR Cryptology ePrint Archive*, 2016:1097, 2016.
4. Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 528–556, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.
5. Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg, Germany.
6. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012.

7. Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In Madhu Sudan, editor, *ITCS 2016: 7th Innovations in Theoretical Computer Science*, pages 345–356, Cambridge, MA, USA, January 14–16, 2016. Association for Computing Machinery.
8. Nir Bitansky, Huijia Lin, and Omer Paneth. On removing graded encodings from functional encryption. *IACR Cryptology ePrint Archive*, 2016:962, 2016.
9. Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From cryptomania to obfustopia through secret-key functional encryption. In *Theory of Cryptography - 14th International Conference, TCC 2016-B*, 2016.
10. Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In Venkatesan Guruswami, editor, *56th Annual Symposium on Foundations of Computer Science*, pages 1480–1498, Berkeley, CA, USA, October 17–20, 2015. IEEE Computer Society Press.
11. Nir Bitansky and Vinod 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*, pages 171–190, 2015.
12. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation: From approximate to exact. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 67–95, 2016.
13. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, pages 440–456, 2005.
14. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, pages 213–229, 2001.
15. Dan Boneh, David J. Wu, and Joe Zimmerman. Immunizing multilinear maps against zeroizing attacks. *IACR Cryptology ePrint Archive*, 2014:930, 2014.
16. Xavier Boyen. The uber-assumption family. In *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings*, pages 39–56, 2008.
17. Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 1–25, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.
18. Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 176–206, 2016.
19. Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On obfuscation with random oracles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 456–467, Warsaw, Poland, March 23–25, 2015. Springer, Heidelberg, Germany.
20. Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 3–12, 2015.
21. Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Rosario Gennaro and Matthew J. B.

Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I*, volume 9215 of *Lecture Notes in Computer Science*, pages 247–266, Santa Barbara, CA, USA, August 16–20, 2015. Springer, Heidelberg, Germany.

22. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17, Athens, Greece, May 26–30, 2013. Springer, Heidelberg, Germany.
23. Sanjam Garg, Craig Gentry, and Shai 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*, pages 1–17, 2013.
24. Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure MPC from indistinguishability obfuscation. In Yehuda Lindell, editor, *TCC 2014: 11th Theory of Cryptography Conference*, volume 8349 of *Lecture Notes in Computer Science*, pages 74–94, San Diego, CA, USA, February 24–26, 2014. Springer, Heidelberg, Germany.
25. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual Symposium on Foundations of Computer Science*, pages 40–49, Berkeley, CA, USA, October 26–29, 2013. IEEE Computer Society Press.
26. Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II*, volume 9563 of *Lecture Notes in Computer Science*, pages 480–511, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.
27. Sanjam Garg and Akshayaram Srinivasan. Unifying security notions of functional encryption. *IACR Cryptology ePrint Archive*, 2016:524, 2016.
28. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing*, pages 555–564, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.
29. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 162–179, 2012.
30. Antoine Joux. The weil and tate pairings as building blocks for public key cryptosystems. In *Algorithmic Number Theory, 5th International Symposium, ANTS-V, Sydney, Australia, July 7-12, 2002, Proceedings*, pages 20–32, 2002.
31. Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. *IACR Cryptology ePrint Archive*, 2016:561, 2016.
32. Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 28–57. Springer, 2016.
33. Huijia Lin. Indistinguishability obfuscation from DDH on 5-linear maps and locality-5 prgs. *IACR Cryptology ePrint Archive*, 2016:1096, 2016.
34. Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 9615 of *Lecture Notes in Computer Science*, pages 447–462, Taipei, Taiwan, March 6–9, 2016. Springer, Heidelberg, Germany.

35. Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 96–124, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.
36. Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016*, 2016.
37. Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. On the impossibility of virtual black-box obfuscation in idealized models. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 18–48, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.
38. Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 1–12, Cirencester, UK, December 19–21, 2005. Springer, Heidelberg, Germany.
39. Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 629–658, 2016.
40. Rafael Pass and Abhi Shelat. Impossibility of VBB obfuscation with ideal constant-degree graded encodings. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 3–17, Tel Aviv, Israel, January 10–13, 2016. Springer, Heidelberg, Germany.
41. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 1–20, 2004.
42. Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 10: 17th Conference on Computer and Communications Security*, pages 463–472, Chicago, Illinois, USA, October 4–8, 2010. ACM Press.
43. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, USA, May 31 – June 3, 2014. ACM Press.
44. Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 256–266, Konstanz, Germany, May 11–15, 1997. Springer, Heidelberg, Germany.
45. Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II*, volume 9057 of *Lecture Notes in Computer Science*, pages 439–467, Sofia, Bulgaria, April 26–30, 2015. Springer, Heidelberg, Germany.