# On Perfect Correctness without Derandomization

Gilad Asharov* Naomi Ephraim† Ilan Komargodski† Rafael Pass†

September 10, 2019

**Abstract**

We give a method to transform any indistinguishability obfuscator that suffers from correctness errors into an indistinguishability obfuscator that is *perfectly* correct, assuming hardness of Learning With Errors (LWE). The transformation requires sub-exponential hardness of the obfuscator and of LWE. Our technique also applies to eliminating correctness errors in general-purpose functional encryption schemes, but here it is sufficient to rely on the polynomial hardness of the given scheme and of LWE. Both of our results can be based *generically* on any perfectly correct, single-key, succinct functional encryption scheme (that is, a scheme supporting Boolean circuits where encryption time is a fixed polynomial in the security parameter and the message size), in place of LWE.

Previously, Bitansky and Vaikuntanathan (EUROCRYPT '17) showed how to achieve the same task using a derandomization-type assumption (concretely, the existence of a function with deterministic time complexity $2^{O(n)}$ and non-deterministic circuit complexity $2^{\Omega(n)}$) which is non-game-based and non-falsifiable.

*Work was concluded while at Cornell Tech, New York, NY 10044, USA, supported by a Simons Foundation junior fellow award. Currently at J.P. Morgan, AI Research. `asharov@cornell.edu`.

†Cornell Tech, New York, NY 10044, USA. Emails: `nephraim@cs.cornell.edu`, `komargodski@cornell.edu`, `rafael@cs.cornell.edu`. Supported in part by NSF Award CNS-1561209, NSF Award SATC-1704788, and AFOSR Award FA9550-18-1-0267.

# Contents

**1 Introduction** **1**
1.1 Our Results . . . 2
1.2 Technical Overview . . . 3
1.3 Related Work . . . 5

**2 Preliminaries** **5**
2.1 Circuits and Algorithms . . . 6
2.2 Indistinguishability Obfuscation . . . 6
2.3 Functional Encryption . . . 7
2.3.1 Compactness of FE . . . 9
2.4 Learning with Errors . . . 9

**3 Correcting Errors in Indistinguishability Obfuscation** **10**
3.1 Perfectly Correct Succinct FE and ABE . . . 11

**4 Correcting Errors in Functional Encryption** **13**
4.1 Weakly Sublinear Compact FE to XiO . . . 14
4.2 Weakly Sublinear Compact FE + sFE to Sublinear Compact FE . . . 16

**References** **18**

**A Perfectly Correct Succinct FE** **22**
A.1 Perfectly Correct FHE . . . 23
A.2 Perfectly Correct Garbled Circuits . . . 23
A.3 Bootstrapping Succinct FE . . . 24
A.4 Long-Output Succinct FE . . . 24

**B Perfect XiO and Succinct FE to Perfect iO** **24**
B.1 From Perfect XiO to Perfect FE . . . 24
B.2 From Perfect FE to Perfect RE . . . 25
B.3 Perfect RE to Perfect iO . . . 26

**C Additional Proofs from Section 4.2** **27**

# 1 Introduction

Randomness is a key resource in cryptography. Basic cryptographic primitives cannot be constructed without randomization, even when only requiring security against deterministic adversaries. Unfortunately, randomized algorithms are often error prone, which makes the resulting constructions suffer in correctness. As an example, consider an encryption scheme based on "noisy" assumptions (such as the ones related to lattices). In these cases, there are often a few random coins that lead to ciphertexts that do not decrypt correctly, thus preventing the scheme from being perfectly correct.

Perfectly correct schemes provide a substantive advantage over schemes that are imperfect. Particularly, there are applications in which the security relies on the perfect correctness of the underlying building blocks (these include, e.g., [10, 12, 13]). For example, Bitansky and Paneth [12] showed how to get non-interactive witness indistinguishable proofs from *perfectly* correct indistinguishability obfuscation and one-way functions (see more on this below).

The importance of perfect correctness has motivated many works in the past to focus on eliminating correctness errors in cryptographic primitives. Some works, such as [23, 34, 39, 15, 3], address correctness in encryption schemes and obfuscation-related primitives, and show how to *amplify* correctness, while not fully eliminating all errors. Other works, such as [27, 24, 26], fully eliminate correctness errors in interactive proofs and *concrete* encryption schemes. Recently, Bitansky and Vaikuntanathan [16] gave a generic technique to fully eliminate correctness errors which applies to a wide range of cryptographic primitives, including encryption schemes and indistinguishability obfuscation.

The technique of [16] is inspired by ideas from complexity theory, or more precisely, from the field of *derandomization*. They use a (non-cryptographic) pseudorandom generator (PRG) that fools bounded-resource algorithms. Given the complexity of the algorithm one wishes to fool, such PRGs are known to exist based on worst-case size lower bounds for Boolean circuits [43, 35, 47]. In the concrete application of Bitansky and Vaikuntanathan [16], the assumption is the following (plausible) worst-case size lower bound on non-deterministic Boolean circuits: there is an $n$-input Boolean function in $\mathbf{E} \triangleq \mathbf{DTime}(2^{O(n)})$ with non-deterministic circuit complexity $2^{\Omega(n)}$. Previously, this assumption was used by Barak et al. [9] in a related way yet for a different purpose: saving rounds of interaction in commitment schemes and ZAPs.

The transformation of [16] is very general, and enables correcting errors in various types of cryptographic primitives. However, while the use of derandomization is natural and elegant in the context of obtaining perfect correctness, it introduces a new dimension of assumptions for cryptographic constructions. In particular, the assumption above is *not* a game-based assumption and *not* a falsifiable one [41] (under any plausible definition).[1] This raises the question of whether derandomization is necessary to eliminate correctness errors in cryptographic primitives, or whether we can achieve this with only "standard" game-based cryptographic assumptions.

To date, there is no generic way to completely immunize indistinguishability obfuscation from errors using only falsifiable or game-based assumptions, and the only way to achieve this goal is using the derandomization approach of [16]. In this work, we show (perhaps surprisingly) how to completely immunize sub-exponentially secure indistinguishability obfuscation by relying only on perfectly correct, sub-exponentially secure, single-key Boolean functional encryption, which as we show can be instantiated with Learning With Errors (LWE). We also show a similar result for functional encryption (FE).

[1] To break the assumption one has to present a non-deterministic circuit of size $2^{o(n)}$ that computes a function in $\mathbf{E}$, but verifying the latter takes exponential time.

## 1.1 Our Results

We give a generic transformation that starts with indistinguishability obfuscation (iO) with imperfect correctness and results in perfectly correct iO. By imperfect correctness, we mean that with all but negligible probability over the choice of the randomness of the obfuscator, for any circuit $C$, the output of the obfuscation on any input $x$ is $C(x)$. Our transformation additionally relies on perfectly correct, single-key functional encryption for Boolean circuits in the public-key setting, which is succinct in the sense that the time to encrypt is independent of the circuit size that the scheme supports (up to poly-logarithmic factors). Henceforth, we will refer to this as succinct FE. Our first result is a generic transformation.

**Theorem 1.1** (Informal)**.** *Assume the existence of iO with imperfect correctness and succinct FE with perfect correctness, both with sub-exponential security. Then, there exists iO with perfect correctness.*

Moreover, we show that with some modifications, perfectly correct succinct FE can be instantiated using the construction of Goldwasser et al. [28], which relies only on the hardness of LWE. We therefore obtain the following corollary.

**Corollary 1.2** (Informal)**.** *Assume the existence of sub-exponentially secure iO with imperfect correctness and sub-exponential hardness of LWE. Then, there exists iO with perfect correctness.*

Note that our assumption on the correctness of the initial iO can be relaxed by using known generic correctness amplification transformations [15, 3]. Concretely, using the transformation of Ananth et al. [3], without additional assumptions, we can start with an iO that guarantees correctness for every $C$ only on most inputs and most random strings used by the obfuscator.

Our techniques also apply to functional encryption (FE) schemes, but in this case, our transformation relies only on *polynomial* hardness of both succinct FE and the given FE scheme. Here, imperfect correctness means that for a function $f$, with all but negligible probability over the choice of the master public key and secret key pair, the decryption of any input $x$ together with a key for $f$ results with $f(x)$. Our result applies both in the public-key and private-key settings, but we focus on the public-key setting for concreteness.

**Theorem 1.3** (Informal)**.** *Assume the existence of a public-key functional encryption scheme with imperfect correctness and succinct FE with perfect correctness. Then, there exists a public-key functional encryption scheme with perfect correctness.*

By again instantiating the perfectly correct succinct FE with LWE, we obtain the following corollary.

**Corollary 1.4** (Informal)**.** *Assume the existence of a public-key functional encryption scheme with imperfect correctness. Then, assuming (polynomial) hardness of LWE, there exists a public-key functional encryption scheme with perfect correctness.*

As in the case with iO, our assumption on the correctness of the initial FE scheme can be relaxed by using known generic transformations [15].

**Applications.** Our transformation gives a way to get applications from *imperfect* iO that were only previously known from *perfect* iO.

For example, consider *non-interactive commitments.* These are known to exist based on one-way permutations (Blum [17]), from any one-way function plus the same derandomization assumption from above (Barak et al. [9]), or from LWE (Goyal et al. [32]). Bitansky, Paneth, and Wichs [13] also gave a construction based on iO with *perfect correctness* and one-way functions.[2] Using the result of Bitansky and Vaikuntanathan [16] to fix the errors in an imperfect iO gives a strictly weaker result than that of [9], as it additionally requires assuming the existence of imperfect iO. Applying our result changes the picture: it gives a generic construction of non-interactive commitments from (sub-exponentially secure) iO, even if the latter has imperfect correctness, assuming also sub-exponentially secure succinct FE with perfect correctness.

The same reasoning applies to *non-interactive witness indistinguishable proofs* (NIWI). There are three known constructions. One, of Barak et al. [9], is based on ZAPs (NIZKs in the common random string model) [22] plus the derandomization assumption mentioned above. The second, of Groth et al. [33], is based on a specific number theoretic assumption on bilinear groups. The latest construction, of Bitansky and Paneth [12], is based on iO with *perfect* correctness, ZAPs (which can in turn be based on iO and one-way functions [12]), and non-interactive commitments (discussed above). Our result therefore implies a construction of NIWI that is based on (sub-exponentially secure) imperfect iO and perfectly correct succinct FE and no derandomization or specific number theoretic assumptions.

**Corollary 1.5.** *Assuming sub-exponentially secure iO, even with imperfect correctness, and sub-exponentially secure succinct FE with perfect correctness, there exist non-interactive commitments and non-interactive witness indistinguishable proofs.*

## 1.2 Technical Overview

Our transformations rely on many known building blocks and transformations from the literature. Most of the technical engineering effort is devoted to revisiting and adapting them to our setting in a way that achieves and preserves perfect correctness.

**Correcting iO.** Suppose we have an obfuscator $\mathsf{i}\mathcal{O}$ for which there is a tiny possibility that the obfuscated circuit does not agree with the given circuit. Namely, for every circuit $C$ and security parameter $\lambda$:

$$\Pr_{r \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}} \left[ \forall x \colon C(x) = \widetilde{C}(x), \text{ where } \widetilde{C} = \mathsf{i}\mathcal{O}(1^\lambda, C; r) \right] \geq 1 - 2^{-\lambda} \ .$$

Can this be transformed into a perfectly correct scheme? Intuitively, this is possible by cleverly choosing the randomness for the obfuscator to find a "good" $r$ that works for all circuits. Indeed, this calls for techniques from the realm of derandomization. In the context of derandomizing **BPP**, one has an algorithm $A(x, r)$ that decides (with some bounded error) whether $x$ is a member of some language $L$. The error is eliminated by running $A$ on many random tapes corresponding to the images of a Nisan-Wigderson PRG [42], and then outputting the majority. Such NW-PRGs produce $\mathsf{poly}(\lambda)$-long strings using short logarithmic-size seeds, and can be constructed under worst-case size lower bounds on circuits [43, 35, 47].

Applying this idea directly for iO would result in a deterministic obfuscator, and therefore would be insecure. Bitansky and Vaikuntanathan [16] showed how to combine true randomness

[2] Their main observation is that in (a minor variant of) Blum's construction [17] it is sufficient to have a *family* of injective one-way functions such that every key in the support of the key-generation algorithm defines an injective function. Indeed, [13] constructed such a family from iO with perfect correctness and one-way functions. We observe that their construction is insufficient for the application if the iO has imperfect correctness.

More precisely, the modification that [13] suggest to Blum's commitment scheme is that during the opening stage the committer will reveal the randomness it used for the obfuscation. In other words, the obfuscation is treated as a (statistical) commitment to the *functionality* of the circuit which prevents the committer from opening with two different functionalities. However, when the obfuscation is imperfect, there could be two functionally *different* circuits that are mapped under some randomness to the *same* obfuscated circuit. This can be used to break binding.

together with the pseudorandomness produced by the NW-PRG, to end up with a scheme that is simultaneously secure and fully correct.[3]

Instead of correcting the obfuscator by searching for "good" randomnesses, we follow a direct route for correcting $\mathsf{i}\mathcal{O}$ that obviates the need for derandomization and its associated assumption. Suppose we had an efficient procedure to check whether a circuit $C$ was obfuscated correctly, that is, whether $C(x) = \widetilde{C}(x)$ for every $x$, where $\widetilde{C} \leftarrow \mathsf{i}\mathcal{O}(C)$. Then, we could easily construct a perfectly correct iO: Given a circuit $C$ we run $\widetilde{C} \leftarrow \mathsf{i}\mathcal{O}(C)$, and check whether $C$ and $\widetilde{C}$ are functionally equivalent. If they are, we output $\widetilde{C}$. Otherwise (which happens with negligible probability), we simply output $C$. This degrades the security of the scheme by a negligible amount,[4] and guarantees perfect correctness.

Unfortunately, in general, there is no efficient procedure to check whether two circuits are functionally equivalent. Nevertheless, this becomes possible if we restrict our attention to circuits with only logarithmic-size input. Indeed, this already gives us (an unconditional!) method to convert an imperfectly correct indistinguishability obfuscator for the class of circuits with logarithmic-size input into a perfectly correct one. However, obtaining iO for such circuits is trivial (without assumptions) as one can just output the truth table as the obfuscation. So, does this observation have any bearing on iO for general circuits?

In [37], Lin et al. formalized the notion of exponentially-efficient iO (XiO), a relaxation of iO, which applies to circuits with logarithmic-size input. In XiO, the obfuscator may run in exponential time in the input length, but must output obfuscations of *sublinear* size in the truth table of the circuit. This rules out the trivial constructions of iO mentioned above. The main result of Lin et al. is that, assuming sub-exponentially secure succinct FE, sub-exponentially secure XiO is sufficient to obtain full fledged iO. We observe that correcting XiO is possible with the trick described above (and without any additional assumptions): let the obfuscator verify that the obfuscation is correct and if not output the circuit itself.

Our first step is thus to view the given imperfect iO as an imperfect XiO. Then, we can transform it into a perfectly correct one as described above. This was also done in Asharov et al. [7] in the context of (directly) correcting XiO. Once we obtain a perfect XiO, we proceed with the outline of [37]. Assuming succinct FE with perfect correctness, the steps of [37] are generic and can be easily shown to preserve correctness.

We also show that our construction can be based on LWE, rather than succinct FE with perfect correctness. To do so, we rely on the construction of succinct FE from LWE due to Goldwasser et al. [29], and modify the necessary parts of their construction to obtain perfect correctness. In particular, their construction of succinct FE is based (generically) on attribute-based encryption (ABE) [31] and FHE [21], both of which can be based on LWE. Due to the noisy nature of LWE, some of the known instantiations of these primitives introduce correctness errors. Therefore, to make the above primitives perfectly correct, we identify the points where correctness errors might occur and observe that they are all *detectable*. This allows us to give up once an error happens and "push" the correctness errors into the security loss.

**Correcting FE.** Our transformation for FE follows a similar path, but is more complicated since we wish to incur only polynomial security loss. First, we observe that FE implies XiO, by the results of [11, 5]. We then correct the XiO, as described above. Then, one option is to go all the way to iO (using [37]) and then back to FE, but this would require assuming sub-exponential security. Instead we present a direct method to go from XiO to FE assuming succinct FE with perfect correctness,

[3] A delicate point in the their proof is showing that shifting real randomness by randomness that comes from the NW-PRG is good enough. This is where they need to use the fact that the NW-PRG fools non-deterministic computation. See [16] for more detail.

[4] The security degradation can be made arbitrary small by **BPP**-style amplification (i.e., parallel repetition and majority).

which can again be based on LWE. We note that for this result, it suffices to start with an FE that is only weakly sublinear compact[5] to obtain a perfectly correct FE with sublinear compactness (that is, the type of compactness which suffices for applications such as iO).

### 1.3 Related Work

The task of eliminating correctness errors in cryptographic primitives was addressed in many works in the past. Dwork, Naor, and Reingold [23], Holenstein and Renner [34], and Lin and Tessaro [39] gave generic methods to immunize any encryption scheme. These guarantee that there are no errors in the encryption and decryption procedures, but leave the possibility of errors in the key-generation procedure. Partial elimination of errors was also recently achieved for indistinguishability obfuscation (iO) by Bitansky and Vaikuntanathan [15], who relied on the sub-exponential security of the given obfuscator and of the LWE assumption. This was later improved by Ananth, Jain, and Sahai [3] by assuming only polynomial security of the given obfuscator and one-way functions.

Regarding *completely* eliminating correctness errors, the works of Goldreich, Mansour, and Sipser [27] and Furer et al. [24] showed how to generically translate every interactive protocol that has imperfect correctness into a perfectly correct one. Concrete implementations of encryption schemes that are based on lattices (such as the ones of Ajtai and Dwork [1] and Regev [46]) are usually noisy and thus prone to correctness errors. Goldreich, Goldwasser, and Halevi [26] suggested variants that have perfect correctness. As was already mentioned, Bitansky and Vaikuntanathan [16] suggested a generic technique, based on derandomization assumptions, to achieve this task for any cryptographic primitive. The technique can be applied to any cryptographic scheme that remains secure under parallel repetition (such as encryption schemes and indistinguishability obfuscation).

In this work, we rely on exponentially-efficient iO (XiO) [37] in our transformations (see Section 1.2). Asharov et al. [7] previously showed how to correct errors in XiO, assuming (polynomial hardness of) LWE and NIZK. Their transformation starts with an XiO which is only approximately correct, that is, for every circuit $C$, the obfuscated circuit $\widetilde{C}$ is correct with noticeable probability over *both* the input to the circuit and the randomness for the obfuscator. In particular, they first transform the approximate obfuscator to an imperfect one.[6] Then, they observe that imperfect XiO can easily be made perfect. We use this observation in this work (see Section 1.2 for details).

**Organization.** The preliminaries we use in our paper are given in Section 2. In Section 3 we provide our transformation from imperfect iO into perfect iO, and in Section 4 we provide our transformation for FE. In order to obtain our results, we combine known building blocks from the literature, and in some cases, make modifications to ensure perfect correctness. The known transformation are mostly given in the appendices.

## 2 Preliminaries

For a distribution $X$ we denote by $x \leftarrow X$ the process of sampling a value $x$ from the distribution $X$. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value $x$ from the uniform distribution over $\mathcal{X}$. For a randomized function $f$ and an input $x \in \mathcal{X}$, we denote by $y \leftarrow f(x)$ the process of sampling a value $y$ from the distribution $f(x)$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. We use PPT as an abbreviation for probabilistic polynomial time.

Throughout the paper, we denote the security parameter by $\lambda$. A function $\mathsf{negl}\colon \mathbb{N} \to \mathbb{R}^+$ is *negligible* if for every constant $c > 0$ there exists an integer $N_c$ such that $\mathsf{negl}(\lambda) < \lambda^{-c}$

[5]This is a somewhat weak notion of FE that guarantees that the length of a ciphertext is sublinear in the size of the functions for which we generate keys. Our transformation thus works even if we start with stronger notions of FE.

[6]In [7], an imperfect obfuscator was called a *worst-case correct* obfuscator.

for all $\lambda > N_c$. Two sequences of random variables $X = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are $\mu$-*computationally indistinguishable* if for any probabilistic polynomial-time algorithm $\mathcal{A}$ it holds that $\left|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]\right| \leq \mu(\lambda)$ for all large enough $\lambda \in \mathbb{N}$. We denote that $\{X_\lambda\}$ and $\{Y_\lambda\}$ are computationally indistinguishable by $\{X_\lambda\} \approx \{Y_\lambda\}$.

## 2.1 Circuits and Algorithms

Boolean circuits correspond to directed acyclic graphs in which every gate is labeled by a Boolean operation. We parametrize Boolean circuits by their size $s$, the number of inputs they accept $n$, and their depth $d$. As usual, the size of a circuit is defined to be the number of wires in it.

**Definition 2.1.** *For any functions $s(\cdot)$, $n(\cdot)$, and $d(\cdot)$, we define $\mathcal{C}^{s,n,d}$ to be the class of circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ for which for any $C \in C_\lambda$, the size of $C$ is at most $s = s(\lambda)$, the input length of $C$ is at most $n = n(\lambda)$, and the depth of $C$ is at most $d = d(\lambda)$. We sometimes omit the depth $d$ and refer to $C^{s,n}$ as the class of circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ with size $s$ and input length $n$.*

**Definition 2.2.** *We define the following classes of circuits:*

- $\mathsf{P}^{\log}$*: the collection of circuit classes $\mathcal{C}^{s,n}$ for which $s$ is a polynomial and $n$ is logarithmic function.*
- $\mathsf{P}$*: the collection of circuit classes $\mathcal{C}^{s,n}$ for which $s$ and $n$ are polynomials.*
- $\mathsf{NC}^i$*: the class of circuits $C^{s,n,d}$ where $s$ and $n$ are polynomials and $d(\lambda) = O(\log^i(\lambda))$.*

**Definition 2.3.** *For a (uniform) algorithm $A$, described by a Turing machine, with input $x$, we denote by* $\mathsf{Time}[A(x)]$ *and* $\mathsf{Outlen}[A(x)]$ *upper bounds on the* running time *and* output length *of $A$ on input $x$, respectively.*

## 2.2 Indistinguishability Obfuscation

We recall the definition of indistinguishability obfuscation (iO) for circuits. Informally, iO is a compiler that gets as input a circuit and outputs a functionally equivalent circuit. Concretely, we consider an imperfect notion of correctness, which only guarantees that with overwhelming probability over the randomness of the obfuscator, the obfuscator outputs an equivalent circuit [36]. For security, it ensures that obfuscations of functionally equivalent circuits are computationally indistinguishable.

**Definition 2.4** (Functional Equivalence)**.** *We say that two circuits $C$ and $C'$ are* equivalent *and denote it by $C \equiv C'$ if they compute the same function (i.e., $\forall x : C(x) = C'(x)$).*

**Definition 2.5** (Indistinguishability Obfuscation [8, 25])**.** *An* indistinguishability obfuscator (iO) *for the circuit class $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is a pair of polynomial time algorithms* $(\mathsf{Obf}, \mathsf{Eval})$ *with the following syntax:*

- $\boldsymbol{\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C)}$**:** *The obfuscator is a randomized algorithm that receives the security parameter $1^\lambda$ and a circuit $C \in \mathcal{C}^{s,n}$ and outputs a circuit $\widetilde{C}$. The running time of this procedure is a polynomial in $\lambda, s$, and $n$.*
- $\boldsymbol{y \leftarrow \mathsf{Eval}(\widetilde{C}, x)}$**:** *The evaluator is a deterministic algorithm that receives $\widetilde{C}$ and an input $x$, and outputs a string $y$ or $\perp$.*

*We require the following properties to hold.*

- **Imperfect Correctness.** *Unless otherwise specified, we require the following correctness property. There exists a negligible function such that for all $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$ it holds that*

$$\Pr\left[\forall x : C(x) = \mathsf{Eval}(\widetilde{C}, x)\right] \geq 1 - \mathsf{negl}(\lambda)$$

*where $\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C)$ and the probability is over the random coins of* $\mathsf{Obf}$.

- **Security.** *For any probabilistic polynomial-time distinguisher $\mathcal{D}$, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for all $\lambda, \in \mathbb{N}$ and all $C_0, C_1 \in \mathcal{C}_\lambda$ with $C_0 \equiv C_1$, it holds that*

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

**Sub-exponential security.** We also consider sub-exponential security. We say that an obfuscator is sub-exponentially secure if for some $\epsilon > 0$ there exists functions $t(\lambda) = 2^{\lambda^\epsilon}$ and $\mu(\lambda) = 2^{-\lambda^\epsilon}$ such that for all adversaries $\mathcal{A}$ that run in time $t(\lambda)$, the probability of distinguishing the above two distributions is at most $\mu(\lambda)$.

We also define perfect correctness for iO.

**Definition 2.6** (Perfect Correctness)**.** *An obfuscator for a circuit class $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is* perfectly correct *if for all $\lambda \in \mathbb{N}$, all $C \in \mathcal{C}_\lambda$, and all $x \in \{0,1\}^{n(\lambda)}$, it holds that*

$$\Pr\left[C(x) = \mathsf{Eval}(\widetilde{C}, x)\right] = 1,$$

*where $\widetilde{C} \leftarrow \mathsf{Obf}(1^\lambda, C)$ and the probability is taken over the randomness of* $\mathsf{Obf}$*.*

We also recall the definition of exponentially-efficient iO, a relaxation of iO introduced by Lin et al [37].

**Definition 2.7** (Exponentially-Efficient Indistinguishability Obfuscation [37])**.** *An* exponentially-efficient indistinguishability obfuscator (XiO) *scheme for the circuit class $\mathcal{C}^{s,n} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ is a tuple of algorithm* $(\mathsf{Obf}, \mathsf{Eval})$ *that satisfy the correctness and security properties of iO and have the following efficiency requirement:*

- **Nontrivial Efficiency.** *There exists a constant $\epsilon > 0$ such that for any $\lambda \in \mathbb{N}$, any circuit $C \in \mathcal{C}_\lambda$, there exists a polynomial* $\mathsf{poly}$ *such that*

$$\mathsf{Time}\left[\mathsf{Obf}(1^\lambda, C)\right] = \mathsf{poly}(\lambda, s, 2^n),$$

*and*

$$\mathsf{Outlen}\left[\mathsf{Obf}(1^\lambda, C)\right] = 2^{n(1-\epsilon)} \cdot \mathsf{poly}(\lambda, s).$$

### 2.3 Functional Encryption

We recall the definition of functional encryption (FE). At a high level, FE is an encryption scheme which enables issuing functional keys corresponding to circuits, such that decryption of a ciphertext corresponding to a message $m$ with a key corresponding to a circuit $C$ reveals $C(m)$. As in the case of iO, we consider a notion of imperfect correctness, that guarantees that for every function, with overwhelming probability over the output of the setup algorithm, the scheme operates correctly for all messages.

**Definition 2.8** (Functional Encryption [44, 20, 37])**.** *A public key functional encryption (FE) scheme for a class of circuits $\mathcal{C}^{s,n}$ is a tuple of polynomial-time algorithms* $(\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ *that behaves as follows:*

- $(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$*: The setup algorithm is a randomized algorithm that takes as input the security parameter $\lambda$ and outputs the master secret key* $\mathsf{msk}$ *and public key* $\mathsf{pk}$*.*

- $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, C)$*: The key generation algorithm is a randomized algorithm that takes as input the master secret key* $\mathsf{msk}$ *and some circuit $C \in \mathcal{C}_\lambda$ and outputs the functional secret key* $\mathsf{sk}_C$*.*

- $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, m)$*: The encryption algorithm is a randomized algorithm that takes as input the public key* $\mathsf{pk}$ *and a message* $m$ *and outputs a ciphertext* $\mathsf{ct}$*.*

- $y \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct})$*: The decryption algorithm is a deterministic algorithm takes as input the functional secret key* $sk_C$ *and ciphertext* $\mathsf{ct}$ *and outputs* $y \in \{0,1\}^*$*.*

*We require that FE the following hold.*

- **Imperfect Correctness.** *There exists a negligible function* $\mathsf{negl}$ *such that for every* $\lambda \in \mathbb{N}$*, every* $C \in \mathcal{C}_\lambda$ *we have that:*

$$\Pr\left[\forall m, r_{\mathsf{Enc}}, r_{\mathsf{Keygen}} : \mathsf{Dec}(\mathsf{sk}_C, \mathsf{Enc}(\mathsf{pk}, m; r_{\mathsf{Enc}})) = C(m)\right] \geq 1 - \mathsf{negl}(\lambda),$$

*where* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\mathsf{sk}_C = \mathsf{Keygen}(\mathsf{msk}, C; r_{\mathsf{Keygen}})$*, and the probability is taken over the randomness of* $\mathsf{Setup}$*.*

- **Selective Indistinguishability Security.** *For every PPT* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for every* $\lambda \in \mathbb{N}$*, every circuit* $C \in \mathcal{C}_\lambda$*, and ever pair of messages* $m_0, m_1 \in \{0,1\}^{n(\lambda)}$ *such that* $C(m_0) = C(m_1)$*, it holds that*

$$\left|\Pr\left[\mathcal{A}\left(z, \mathsf{FE.Enc}(\mathsf{pk}, m_b)\right) = b\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

*where* $z = (\mathsf{pk}, C, m_0, m_1, \mathsf{sk}_C)$*,* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$*,* $b \leftarrow \{0,1\}$*, and* $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, C)$*.*

As in the case of obfuscation, we define the perfect notion of correctness for functional encryption.

**Definition 2.9** (Perfect Correctness)**.** *An FE scheme for a circuit class* $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ *is* perfectly correct *if for every* $\lambda \in \mathbb{N}$*, every* $C \in \mathcal{C}_\lambda$*, every* $m \in \{0,1\}^{n(\lambda)}$*, we have that:*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_C, \mathsf{Enc}(\mathsf{pk}, m)) = C(m)\right] = 1,$$

*where* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$*,* $\mathsf{sk}_C \leftarrow \mathsf{Keygen}(\mathsf{msk}, C)$*, and the probability is taken over the randomness of* $\mathsf{Setup}$*.*

We also give a selective simulation security definition for FE. The following definition is adapted from the definition of full simulation security [29], and suffices for our purposes.

**Definition 2.10** (Simulation Security)**.** *Let* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ *be an FE scheme for a class of circuits* $\mathcal{C}^{s,n} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$*. For every PPT* $\mathcal{A}$ *and PPT simulator* $\mathcal{S}$*, define the following two experiments:*

| $\mathsf{Exp}^{\mathsf{real}}_{\mathsf{FE},\mathcal{A}}(\lambda)$: | $\mathsf{Exp}^{\mathsf{ideal}}_{\mathsf{FE},\mathcal{A},\mathcal{S}}(\lambda)$: |
|---|---|
| *1 :* $(m, C, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda)$ | *1 :* $(m, C, \mathsf{st}) \leftarrow \mathcal{A}(1^\lambda)$ |
| *2 :* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ | *2 :* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ |
| *3 :* $\mathsf{sk}_C \leftarrow \mathsf{Keygen}(\mathsf{msk}, C)$ | *3 :* $\mathsf{sk}_C \leftarrow \mathsf{Keygen}(\mathsf{msk}, C)$ |
| *4 :* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m)$ | *4 :* $\mathsf{ct}_{\mathsf{Sim}} \leftarrow \mathcal{S}(\mathsf{mpk}, \mathsf{sk}_C, C, C(m))$ |
| *5 :* *Output* $(\mathsf{ct}, \mathsf{st})$ | *5 :* *Output* $(\mathsf{ct}_{\mathsf{Sim}}, \mathsf{st})$ |

*We say that* $\mathsf{FE}$ *is* simulation secure *if there exists a PPT simulator* $\mathcal{S}$ *such that for all PPT* $\mathcal{A}$*,*

$$\left\{\mathsf{Exp}^{\mathsf{real}}_{\mathsf{FE},\mathcal{A}}(\lambda)\right\}_{\lambda \in \mathbb{N}} \approx \left\{\mathsf{Exp}^{\mathsf{ideal}}_{\mathsf{FE},\mathcal{A},\mathcal{S}}(\lambda)\right\}_{\lambda \in \mathbb{N}}.$$

### 2.3.1 Compactness of FE

**Definition 2.11** (Notions of Compactness for Functional Encryption [14, 4, 38])**.** *We say that a functional encryption scheme* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Keygen}, \mathsf{FE.Dec})$ *for the class of circuits* $\mathcal{C}^{s,n} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$ *is*

- compact *if it holds that*

$$\mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] = \mathsf{poly}(\lambda, |m|, \log(s)),$$

- sublinearly compact *if there exists a constant* $\eta > 0$ *such that*

$$\mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] = s^{1-\eta} \cdot \mathsf{poly}(\lambda, |m|),$$

- weakly sublinearly compact *if there exists a constant* $\eta > 0$ *such that*

$$\mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] = \mathsf{poly}(\lambda, |m|, s)$$
$$\mathsf{Outlen}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] = s^{1-\eta} \cdot \mathsf{poly}(\lambda, |m|),$$

- succinct *if* $\mathsf{FE}$ *is a compact FE scheme for a class of circuits with 1-bit outputs,*

*where the above definitions hold for every* $\lambda \in \mathbb{N}$*,* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ *and* $m \in \{0,1\}^{n(\lambda)}$*, and where* $s = s(\lambda)$*.*

## 2.4 Learning with Errors

**Definition 2.12** ([46])**.** *For an integer* $q = q(n) \geq 2$ *and an error distribution* $\chi = \chi(n)$ *over* $\mathbb{Z}_q$*, the learning with errors problem* $\mathsf{LWE}_{n,m,q,\chi}$ *is to distinguish between*

$$\left\{\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}; \mathbf{s} \leftarrow \mathbb{Z}_q^n; \mathbf{e} \leftarrow \chi^m : (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})\right\}$$

*and*

$$\left\{\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}; \mathbf{u} \leftarrow \mathbb{Z}_q^m : (\mathbf{A}, \mathbf{u})\right\}.$$

**Definition 2.13** (Bounded Distributions)**.** *Let* $B = B(n)$ *such that* $B(n) \in \mathbb{N}$ *for all* $n \in \mathbb{N}$*. A family of distributions* $\chi = \{\chi_n\}_{n\in\mathbb{N}}$ *over the integers is* $B$*-bounded if for all* $n \in \mathbb{N}$*,*

$$\Pr\left[x \leftarrow \chi_n : |x| \leq B(n)\right] = 1.$$

Some of the literature (e.g., [21]) defines $B$-bounded distributions as those that are bounded with overwhelming probability. Regarding cryptographic constructions from LWE (such as those in [21]), the distribution $\chi$ is used as the error distribution, and thus the bound $B$ directly influences the correctness of the resulting scheme. Since the two definitions are statistically close, the hardness of LWE and connection to lattices [46, 45] extend to the case where $\chi$ is bounded with probability 1.

**Perfectly correct PKE from LWE [46].** The public-key encryption scheme of [46], when sampling the noise from a bounded distribution, results in a perfectly correct scheme. The key generation algorithm chooses a random secret $\mathbf{s} \in \mathbb{Z}_q^n$, which would serve as the private key. In addition, the key generation algorithm chooses $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}$, noise $\mathbf{e} \leftarrow \chi^m$, and sets $\mathbf{b} \stackrel{\mathsf{def}}{=} \mathbf{A}\mathbf{s} + \mathbf{e}$. The public key is defined as $(\mathbf{A}, \mathbf{b})$. The encryption algorithm on input message $m \in \{0,1\}$ and public key $(\mathbf{A}, \mathbf{b})$, chooses a random $\mathbf{r} \leftarrow \mathbb{Z}_q^m$ and outputs $(\mathbf{u}, v) \stackrel{\mathsf{def}}{=} (\mathbf{r}^T \cdot \mathbf{A}, \langle \mathbf{r}, \mathbf{b}\rangle + m/2)$. The decryption of $(\mathbf{u}, v)$ is 0 if $v - \langle \mathbf{u}, \mathbf{s}\rangle$ is closer to 0 than to 1/2, and 1 otherwise. The decryption algorithm is in $\mathsf{NC}^1$, the scheme is perfectly correct if the noise distribution $\chi$ is bounded, and is secure under the LWE assumption.

## 3 Correcting Errors in Indistinguishability Obfuscation

In this section we present our main result—a transformation from imperfectly correct iO to perfectly correct iO.

**Theorem 3.1.** *Assume the existence of a sub-exponentially secure iO scheme for* $\mathsf{P}$ *that is imperfectly correct. Then, assuming sub-exponentially secure succinct FE with perfect correctness for* $\mathsf{P}$*, there exists a perfectly correct iO scheme for* $\mathsf{P}$*.*

**Relaxing imperfect correctness.** The above transformation can be made to work with an iO scheme that is only approximately correct, rather than imperfectly correct. In particular, suppose $\mathsf{i}\mathcal{O}$ is correct for any circuit $C$ with probability $(1/2+1/\mathsf{poly}(\lambda))$ over the randomness of the obfuscator and the choice of input (i.e., $\mathsf{i}\mathcal{O}$ is approximately correct). Using the results of Ananth et al. [3] and Bitansky et al. [15], this can be transformed to an imperfect iO scheme (in which failure is only due to the randomness of the obfuscator and *not* the choice of the input). In particular, [3] showed how to obtain an imperfect iO scheme relying only on one-way functions (improving on [15], who gave the same result based on DDH).

*Proof of Theorem 3.1.* Let $\mathsf{i}\mathcal{O}$ be a sub-exponentially secure, imperfectly correct iO scheme for $\mathsf{P}$ (the class of all polynomial-size circuits). Let $\mathsf{P}^{\log}$ be the class of all polynomial-size circuits with logarithmic-size input (see Definition 2.2). The proof follows a sequence of two transformations:

**Imperfect iO $\Rightarrow$ Perfect XiO.** Because iO is a special case of XiO and $\mathsf{P}^{\log} \subset \mathsf{P}$, we can view $\mathsf{i}\mathcal{O}$ as an imperfectly correct XiO scheme $\mathsf{xi}\mathcal{O}$ for $\mathsf{P}^{\log}$. Then, we can transform this scheme into a perfectly XiO correct scheme, by slightly modifying the obfuscation algorithm: Upon receiving a circuit $C$, obfuscate it to obtain $\widetilde{C}$, and then verify that $\widetilde{C}$ is correct by enumerating over all inputs $x$ and checking that $C(x) = \mathsf{xi}\mathcal{O}.\mathsf{Eval}(\widetilde{C}, x)$. If the verification fails, output $C$, and otherwise, output $\widetilde{C}$. Perfect correctness of this transformation is immediate. As for security, by the imperfect correctness of $\mathsf{xi}\mathcal{O}$ it holds that with all but negligible probability over the random coins of the obfuscator, any $\mathsf{xi}\mathcal{O}$ obfuscation $\widetilde{C}$ is correct on all inputs $x$. Thus, there is only a negligible loss in security. In summary, this gives the following claim:

> **Claim 3.2** ([7])**.** *Assuming the existence of a sub-exponentially secure, imperfectly correct XiO scheme for* $\mathsf{P}^{\log}$*, there exists a sub-exponentially secure, perfectly correct XiO scheme for* $\mathsf{P}^{\log}$*.*

**XiO + sFE $\Rightarrow$ iO.** Given a perfectly correct XiO scheme, our goal is to transform it to a perfectly correct iO. Lin et al. [37] show that the existence of sub-exponentially secure XiO for $\mathsf{P}^{\log}$, together with sub-exponentially secure succinct FE for $\mathsf{P}$, suffice for achieving full-fledged iO for $\mathsf{P}$. We revisit this transformation and show that once both of the underlying primitives are perfectly correct, we end up with perfectly correct iO for $\mathsf{P}$. This transformation consists of some additional intermediate steps (first transforming XiO and sFE into weakly sublinearly compact FE, then to randomized encodings, and only then to iO). We verify that the transformation preserves the perfect correctness and elaborate on it in Appendix B. No modifications are needed for this transformation.

> **Claim 3.3** ([37])**.** *Assume the existence of a sub-exponentially secure, perfectly correct, succinct FE scheme for* $\mathsf{P}$*, and a sub-exponentially-secure, perfectly correct XiO scheme for* $\mathsf{P}^{\log}$*. Then, there exists a perfectly correct iO scheme for* $\mathsf{P}$*.*

Thus, we end up with a perfectly correct iO scheme for $\mathsf{P}$. $\square$

We also observe that perfectly correct succinct FE can be based on LWE.

**Claim 3.4.** *Assuming sub-exponential hardness of LWE, there exists a perfectly correct succinct FE for* $\mathsf{P}$*.*

Combined with Theorem 3.1, this yields the following corollary.

**Corollary 3.5.** *Assume the existence of a sub-exponentially secure iO scheme for* $\mathsf{P}$ *that is imperfectly correct. Then, assuming sub-exponential hardness of LWE, there exists a perfectly correct iO scheme for* $\mathsf{P}$*.*

To prove Claim 3.4, we show how to instantiate the $\mathsf{sFE}$ construction of [29] to ensure perfect correctness. The construction of [29] relies on two main building blocks: fully homomorphic encryption (FHE) and attribute-based encryption (ABE), such that if both satisfy perfect correctness, then $\mathsf{sFE}$ is perfectly correct. There are known constructions of FHE that satisfy perfect correctness (e.g., the one of [21]; see Section A.1), but known ABE constructions (for example, those of [31, 19]) do not satisfy perfect correctness.[7] To this end, we show how to modify the ABE construction of [31] to get a perfectly correct ABE and thereby a perfectly correct $\mathsf{sFE}$. The modification is presented in Section 3.1, and we verify that the remainder of the $\mathsf{sFE}$ construction of [29] is perfectly correct in Appendix A.

## 3.1 Perfectly Correct Succinct FE and ABE

One of the key ingredients in obtaining perfect iO is a perfectly correct, succinct FE. To give an instantiation which satisfies perfect correctness, our starting point is the construction of succinct FE of Goldwasser et al. [29]. Their construction relies on three building blocks: attribute-based encryption (ABE), fully homomorphic encryption (FHE), and garbled circuits. In Appendix A, we overview their construction, and verify that if all the building blocks are perfectly correct, then the resulting succinct FE is perfectly correct. We also verify the correctness of known constructions of FHE and garbled circuits. However, obtaining a perfectly correct ABE construction requires more care. Therefore, in this section, we show how to modify the Gorbunov et al. [31] construction of ABE, so that it can be used to construct perfectly correct, succinct FE. That is, we show the following claim.

**Claim 3.6** (Claim 3.4, restated)**.** *Assuming sub-exponential hardness of LWE, there exists a perfectly correct succinct FE scheme for* $\mathsf{P}$ *that is sub-exponentially simulation secure.*

We proceed by discussing the ABE construction of [31]. At a high level, ABE is an encryption scheme which allows encrypting a message $m$ together with an attribute $a$, and generating keys $\mathsf{sk}_P$ corresponding to predicate circuits $P$. The correctness property is that the encryption of $(m, a)$ decrypts to $m$ using $\mathsf{sk}_P$ whenever $P(a) = 1$. Gorbunov et al. [31] give a construction of ABE for polynomial-size circuits with a-priori bounded depth.

**Theorem 3.7** ([31, Corollary 6.2])**.** *For all $n$ and polynomials $d = d(n)$, there exists a selectively secure ABE scheme* $\mathsf{ABE}$ *for any class of polynomial-size circuits with $n$ inputs and depth $d$, assuming hardness of* $\mathsf{LWE}_{\ell,m,q,\chi}$ *for sufficiently large* $\ell = \mathsf{poly}(\lambda, d)$, $q = \ell^{O(d)}$, $m = \mathsf{poly}(\ell)$, *and poly*$(\ell)$*-bounded distribution* $\chi$*.*

*Moreover, assuming sub-exponential hardness of LWE, it holds that* $\mathsf{ABE}$ *is sub-exponentially secure.*

[7] Note that in [31] it is claimed that the scheme satisfies perfect correctness, however, a closer look reveals that there is negligible probability, over the setup stage and the key generation, that key generation fails (see [31, Lemma 3.2]). Our modification confirms their claim for perfect correctness. See Section 3.1 for details.

Let the ABE construction in [31] be $\mathsf{ABE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Keygen}, \mathsf{Dec})$. It was shown in [31] that $\mathsf{ABE}$ has the following correctness guarantee: For $C \in \mathcal{C}_\lambda$, it holds that

$$\Pr\left[\forall a \text{ with } C(a) = 1, \forall m, r_{\mathsf{enc}} \colon \mathsf{Dec}(\mathsf{sk}_C, \mathsf{Enc}(\mathsf{mpk}, a, m; r_{\mathsf{enc}}) = m\right] \geq \\ 1 - \mathsf{negl}(\lambda)$$

for a negligible function $\mathsf{negl}$, where $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$ and $\mathsf{sk}_C \leftarrow \mathsf{Keygen}(\mathsf{msk}, C)$ (and the probability is taken over the randomness of $\mathsf{Setup}$ and $\mathsf{Keygen}$). However, with a slight modification, the scheme can be made perfectly correct. We start with a high level overview of the scheme.

**The ABE construction.** The construction uses two-to-one recoding (TOR), which is constructed from LWE, and serves as a re-encryption mechanism. At a high level, a TOR scheme enables encoding a secret $s$ under a key $\mathsf{pk}$ using an algorithm $\mathsf{Encode}$, and then generating a recoding key $\mathsf{rk}$ using an algorithm $\mathsf{Recode}$, such that given encodings of $s$ under $\mathsf{pk}_0$ and $\mathsf{pk}_1$, one can use $\mathsf{rk}$ to compute an encoding of $s$ under a new target key $\mathsf{pk}_{\mathsf{tgt}}$. Using a TOR scheme, the ABE construction for circuits with $n$ input bits is as follows.

The setup algorithm creates a TOR key-pair $(\mathsf{pk}_{i,b}, \mathsf{sk}_{i,b})$ for each input bit $i \in [n]$ and each $b \in \{0,1\}$, as well as a special public key $\mathsf{pk}_{\mathsf{out}}$. To encrypt a message $m$ with an attribute $a$, a random value $s$ is encoded as $\psi_i \leftarrow \mathsf{Encode}(\mathsf{pk}_{i,a_i}, s)$ for each $i \in [n]$. The ciphertext contains these encodings, as well as a masked value $\tau$ of $m$ under $\mathsf{Encode}(\mathsf{pk}_{\mathsf{out}}, s)$. To generate a key for a circuit $C$, recoding keys $\mathsf{rk}_{g,b,c}$ are generated for each gate $g$ of $C$ and each pair of inputs $b, c \in \{0,1\}$. These essentially enable one to translate encodings of input bits corresponding to an input $x$ to an encoding of $C(x)$ under $\mathsf{pk}_{\mathsf{out}}$ if and only if $C(x) = 1$. Thus, at decryption, one can use the recoding keys to translate $\psi_1, \ldots, \psi_n$ into an encoding of $s$ under $\mathsf{pk}_{\mathsf{out}}$, which can be used to reveal $m$ from $\tau$.

We now analyze the correctness of $\mathsf{ABE}$, and show that with a slight modification to the scheme, it can satisfy perfect correctness.

**Proposition 3.8.** *Assuming perfectly correct public key encryption and hardness of LWE, there exists a perfectly correct ABE scheme for any class* $\mathcal{C}^{s,n,d} \in \mathsf{P}$.

**Proof.** Let $\mathsf{ABE}$ be the [31] ABE scheme. To show this proposition, we show how to modify $\mathsf{ABE}$ into a perfectly correct scheme. We note that there are two potential sources of errors that could contribute to decryption errors: errors during encryption, and errors during key generation (either of which may be caused by errors during setup).

**Errors during encryption.** The construction in [31] does not incur encryption errors. In particular, for any $\mathsf{mpk}$ in the support of $\mathsf{Setup}$, any valid attribute $a$ and message $m$, it holds that the resulting ciphertext $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, a, m)$ can be successfully decrypted using any well-formed key. Recall that the ABE ciphertexts consist of TOR encodings $\psi_i$ of $s$ and a mask $\tau$ of the message $m$. The TOR encodings are simply LWE encodings $\mathbf{As} + \mathbf{e}$ where $\mathbf{A}$ is the TOR public key $\mathsf{pk}_{i,a_i}$, and $\mathbf{e}$ is sampled from an error distribution $\chi$. In [31], $\chi$ is truncated discrete Gaussian distribution, which is a bounded distribution. In this way, they ensure that the error terms are bounded and never cause values to wrap around the LWE modulus $q$. Thus, there are no malformed ciphertexts. For more details, see the analysis of the correctness of TOR in [31].

**Errors during key generation.** The $\mathsf{ABE}$ construction can indeed result in errors during key generation. This is because the TOR construction from LWE does not achieve perfect recoding correctness. Namely, with negligible probability over the randomness of the TOR key generation, $\mathsf{Recode}$ outputs a "bad" recoding key $\mathsf{rk}$. Nevertheless, this can be *detected* by the scheme. In particular, a "bad" recoding key $\mathsf{rk}$ is one where $\|\mathsf{rk}\|_\infty$ is greater than some known threshold $t$, determined as part of the parameters of the scheme.[8] Therefore, we propose the following modification.

[8] We note that even though the error probability is over the randomness of key generation, the error is not detected until a recoding key is generated. Therefore, in the case of a "bad" pair of public and secret keys, which lead to an incorrect recoding key, we cannot verifiably generate a recoding key that would allow us to decrypt.

Let $\mathsf{PKE}$ be any public-key encryption scheme with perfect correctness. The ABE scheme is as follows. During $\mathsf{Setup}$, generate a $\mathsf{PKE}$ key-pair $(\mathsf{pk}, \mathsf{sk})$ and include $\mathsf{pk}$ in the public key of the ABE scheme, and $\mathsf{sk}$ in the master secret key. During encryption of a message $m$ with an attribute $a$, include an additional encryption $\mathsf{ct}^\star$ of $(a, m)$ under $\mathsf{pk}$. Then, in the case of the above error in key generation for a circuit $C$, simply output the master secret key $\mathsf{msk}$ and $C$. Finally, during decryption, in the case that $(\mathsf{msk}, C)$ is received in place of the key for $C$, decrypt $\mathsf{ct}^\star$ using $\mathsf{sk}$ to obtain $(a, m)$, and then output $m$ if and only if $C(a) = 1$. If $\mathsf{PKE}$ is perfectly correct, then the resulting scheme is perfectly correct. Moreover, security is not impacted by the fact that this case only occurs with negligible probability, and by the semantic security of $\mathsf{PKE}$.[9] $\square$

By combining Theorem 3.7 with Proposition 3.8, and noting that perfectly correct public-key encryption follows from LWE (as discussed in Section 2.4), we obtain the following claim.

**Claim 3.9.** *Assuming sub-exponential hardness of LWE as in Theorem 3.7, for all $n$ and polynomials $d = d(n)$, there exists a perfectly correct, sub-exponentially, selectively secure ABE scheme for any class of polynomial-size circuits with $n$ inputs and depth $d$.*

These results give Claim 3.4.

*Proof of Claim 3.4.* Let $\mathcal{C}^{s,n,d}$ be any class of circuits in $\mathsf{NC}^1$. By Theorem A.3, assuming LWE, there exists a perfectly correct $d$-leveled FHE scheme for any polynomial $d = d(n)$ for encrypting $n$ bits. Thus, there exists FHE for $\mathcal{C}^{s,n,d}$. Moreover, the circuit computing homomorphic evaluation of any circuit $C$ with size $s$ and depth $d$ has bounded, polynomial depth, and thus is in some class $\mathcal{C}^{s',n',d'} \in \mathsf{P}$. By Claim 3.9, assuming LWE, there exists a sub-exponentially secure ABE scheme for predicates in $\mathcal{C}^{s',n',d'}$. Moreover, by Theorem A.4, assuming LWE, there exists a sub-exponentially secure garbled circuit for $\mathcal{C}^{s,n,d}$. Putting this all together, by Proposition A.2, there exists a succinct FE scheme for $\mathcal{C}^{s,n,d}$, which has sub-exponential security assuming sub-exponential LWE. Finally, by Corollary A.5, this can be bootstrapped to a perfectly correct, sub-exponentially secure, succinct FE scheme for $\mathsf{P}$. $\square$

# 4 Correcting Errors in Functional Encryption

In this section, we show how to completely eliminate correctness errors in functional encryption schemes. Concretely, starting with an imperfectly correct FE, which guarantees correctness only with overwhelming probability over the randomness of the setup algorithm, we obtain perfectly correct FE for $\mathsf{P}$ and incur only polynomial security loss. We note that the result of this section also holds in the setting of secret-key FE, but for concreteness, the result is presented only in the public-key setting.

Regarding compactness, we only require the FE that we start with to be weakly sublinear compact, which guarantees that the *output length* of the encryption algorithm is sublinear in the size of the circuits that we generate keys for. However, the *time* to encrypt may be long. Our transformation results in an FE that is sublinearly compact, which guarantees that the time to generate a ciphertext is sublinear in the size of the circuits that we generate keys for. Thus, this is a significantly stronger notion.

**Theorem 4.1.** *Assume the existence of an imperfectly correct, weakly sublinear compact FE scheme for $\mathsf{P}$ and a perfectly correct succinct FE scheme for $\mathsf{P}$. Then, there exists a perfectly correct, sublinearly compact FE scheme for $\mathsf{P}$.*

[9] An alternative solution is to add a decoding mechanism to TOR, such that given $\mathsf{Encode}(\mathsf{pk}, s)$, one could decode the encoding using the secret $\mathsf{sk}$ to obtain $s$. This is possible since $\mathsf{TOR}$ is built using lattices and trapdoors, which include inversion algorithms. In this case, if there is an error during key generation for a circuit $C$, one would output the master secret key $\mathsf{msk}$ of the ABE scheme (which consists of TOR secret keys) and the circuit $C$, and then $\mathsf{msk}$ could be used to directly decrypt ciphertexts.

Using Claim 3.4, we obtain the following corollary.

**Corollary 4.2.** *Assume polynomial hardness of LWE. If there exists an imperfectly correct, weakly sublinear compact FE scheme for* $\mathsf{P}$*, there exists a perfectly correct, sublinearly compact FE scheme for* $\mathsf{P}$*.*

**Relaxing imperfect correctness.** The assumption on the initial FE scheme satisfying imperfect correctness can be relaxed to assume only approximate correctness of the scheme. In particular, given an FE scheme that is correct for all circuits with high probability over the output of the setup algorithm *and* over the choice of the message, using the results of [15], this can be transformed into an imperfect FE scheme.

**Overview and Roadmap of Theorem 4.1.** This transformation follows a similar blueprint to that of perfect iO, but includes a new construction. We start with an imperfectly correct, weakly sublinear compact FE scheme for $\mathsf{P}$.

1. **FE $\Rightarrow$ XiO.** Transform the imperfect FE scheme to an imperfect XiO scheme for $\mathsf{P}^{\log}$. This transformation is given in Section 4.1, and is due to [11, 5].
2. **Imperfect XiO $\Rightarrow$ Perfect XiO.** This follows from Claim 3.2.
3. **XiO + sFE $\Rightarrow$ Weakly sublinear compact FE.** Transform the perfectly correct XiO and succinct FE to a perfectly correct, weakly sublinear compact FE scheme $\mathsf{FE}$ for $\mathsf{P}$. This follows from Claim B.3, and is one of the intermediate steps in the proof of Claim 3.3, which we already verified in Section 3.
4. **Weakly sublinear compact FE + sFE $\Rightarrow$ Compact FE.** Combine $\mathsf{FE}$, along with a succinct FE scheme $\mathsf{sFE}$ for $\mathsf{P}$, into a perfectly correct, sublinear compact FE scheme for $\mathsf{P}$. This transformation is given in Section 4.2.

*Proof of Theorem 4.1.* Let $\mathsf{FE}$ be an imperfectly correct, weakly sublinear compact FE scheme for $\mathsf{P}$. By Claim 4.3, there exists an imperfectly correct XiO scheme $\mathsf{xi}\mathcal{O}$ for $\mathsf{P}^{\log}$. By Claims 3.2, 3.4, 3.4, and B.3, this can be transformed into a perfectly correct, weakly sublinear compact FE scheme $\mathsf{FE}$. Moreover, by Claim A.7, there exists a long-output succinct FE scheme such that we can apply Claim 3.4, to obtain a perfectly correct, succinct FE scheme $\mathsf{sFE}$ for $\mathsf{P}$. Then, by Claim 4.4, there exists a sublinear compact, selectively secure FE scheme for $\mathsf{P}$. □

## 4.1 Weakly Sublinear Compact FE to XiO

In this section, we present an FE to XiO transformation which begins with weakly sublinear compact FE and only has polynomial security loss. This transformation appears in [11, 5], but we include it here for completeness.

Let $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{P}^{\log}$ be any circuit class for which we want to obtain XiO. For a circuit $C \in \mathcal{C}_\lambda$, we denote by $C_{b_1\ldots b_t}$ the circuit $C$ where the first $t$ bits of the input are hardwired to $b_1,\ldots,b_t \in \{0,1\}$. We denote by $T$ a circuit that receives as input a circuit and outputs its truth table.

Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Keygen}, \mathsf{FE.Dec})$ be a weakly sublinearly compact, imperfectly correct FE scheme for the class of circuits $\mathcal{C}^{s',n'} = \{\mathcal{C}'_\lambda\}_{\lambda\in\mathbb{N}}$, where $s' = 2^{\frac{n}{d}} \cdot s$ and $n' = s$. Let $p$ be the polynomial of degree $d-1$ for some constant $d$ such that $\mathsf{Outlen}\left[\mathsf{FE.Keygen}(\mathsf{msk}, C)\right] \le p(\lambda, s', n')$ for any $\mathsf{msk}$ in the support of $\mathsf{FE.Setup}$ and any circuit $C \in \mathcal{C}'_\lambda$. The transformation is as follows.

**Weakly sublinear compact FE to XiO.** We define the XiO scheme $\mathsf{xi}\mathcal{O}$ as follows:

- $\widetilde{C} \leftarrow \mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, C)$ :

1. Sample $(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $\mathsf{sk}_T \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, T)$.
2. For every $x_1 \in \{0,1\}^{n-\frac{n}{d}}$, let $\mathsf{ct}_{x_1} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, C_{x_1})$.
3. Output $\widetilde{C} = \left(\mathsf{sk}_T, \{\mathsf{ct}_{x_1}\}_{x_1 \in \{0,1\}^{n-\frac{n}{d}}}\right)$.

- $\boldsymbol{y \leftarrow \mathsf{xi}\mathcal{O}.\mathsf{Eval}(\widetilde{C}, x)}$ :
  1. Parse $x = x_1x_2$ where $|x_1| = n - \frac{n}{d}$.
  2. Run $\mathsf{FE.Dec}(\mathsf{sk}_T, \mathsf{ct}_{x_1})$ to obtain the truth table of $C_{x_1}$, and output the row corresponding to input $x_2$.

**Claim 4.3.** *Let $\mathcal{C}^{s,n}$ be any circuit class in $\mathsf{P}^{\log}$. Let $\mathsf{FE}$ be an imperfectly correct, weakly sublinear compact FE scheme for the class of circuits $\mathcal{C}^{2^{n/d} \cdot s, s}$. Then, there exists an XiO scheme for the class of circuits $\mathcal{C}^{s,n}$.*

**Proof.** Let $\mathsf{xi}\mathcal{O}$ be the scheme for $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ resulting from the above transformation. Let $s = s(\lambda)$ and $n = n(\lambda)$. We show imperfect correctness, efficiency, and security of $\mathsf{xi}\mathcal{O}$.

**Imperfect correctness.** For every $\lambda \in \mathbb{N}$, for any $C \in \mathcal{C}_\lambda$, let $\widetilde{C}$ be sampled from $\mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, C)$ and consider the probability that $\mathsf{xi}\mathcal{O}.\mathsf{Eval}(\widetilde{C}, \cdot)$ agrees with $C(\cdot)$ on all inputs $x = x_1x_1$ where $|x_1| = n - \frac{n}{d}$. The obfuscation $\widetilde{C}$ consists of $\mathsf{pk}, \mathsf{sk}_T$, and $\mathsf{ct}_{x_1}$ for all $x_1$. Thus, this is the probability that for all $x_1x_2$, the $x_2$th row of $\mathsf{FE.Dec}(\mathsf{sk}_T, \mathsf{ct}_{x_1})$ is $C(x_1x_2)$, which is equivalent to the event that for all $x_1$, $\mathsf{FE.Dec}(\mathsf{sk}_T, \mathsf{ct}_{x_1}) = T(C_{x_1})$, over the probability of $\mathsf{FE.Setup}$, $\mathsf{FE.Keygen}$, and $\mathsf{FE.Enc}$. Since $\mathsf{FE}$ is correct with overwhelming probability over the randomness of $\mathsf{FE.Setup}$ (for any randomnesses of $\mathsf{FE.Keygen}$ and $\mathsf{FE.Enc}$), imperfect correctness of $\mathsf{xi}\mathcal{O}$ follows. Formally,

$$
\begin{aligned}
&\Pr\left[\widetilde{C} \leftarrow \mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, C) : \forall x, \mathsf{xi}\mathcal{O}.\mathsf{Eval}(\widetilde{C}, x) = C(x)\right] \\
&= \Pr\left[\begin{array}{l} (\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{FE.Setup}(1^\lambda) \\ \mathsf{sk}_T \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, T) \\ \forall x_1 \in \{0,1\}^{n-\frac{n}{d}} : \\ \ \mathsf{ct}_{x_1} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, C_{x_1}) \end{array} : \begin{array}{c} \forall x_1 : \\ \mathsf{FE.Dec}(\mathsf{sk}_T, \mathsf{ct}_{x_1}) = T(C_{x_1}) \end{array}\right] \\
&\geq 1 - \mathsf{negl}(\lambda),
\end{aligned}
$$

for a negligible function $\mathsf{negl}$, where $|x_1| = n - \frac{n}{d}$, by the imperfect correctness of $\mathsf{FE}$.

**Efficiency.** We have that $|T| = s \cdot 2^{\frac{n}{d}}$ and $T$ receives inputs of size $s$, and has depth linear in $s$. Therefore, for any $C \in C_\lambda$, by the weak sublinear compactness of $\mathsf{FE}$ we have that

$$
\begin{aligned}
&\mathsf{Time}\left[\mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, C)\right] \\
&= \mathsf{Time}\left[\mathsf{FE.Setup}(1^\lambda)\right] + 2^{n-\frac{n}{d}} \cdot \mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, C_x)\right] + \mathsf{Time}\left[\mathsf{FE.Keygen}(\mathsf{msk}, T)\right] \\
&\leq \mathsf{poly}(\lambda, s \cdot 2^{\frac{n}{d}}, s) + 2^{n-\frac{n}{d}} \cdot \mathsf{poly}(\lambda, s \cdot 2^{\frac{n}{d}}, s) + \mathsf{poly}(\lambda, s \cdot 2^{\frac{n}{d}}, s) = \mathsf{poly}(\lambda, s, 2^n)
\end{aligned}
$$

and

$$
\begin{aligned}
&\mathsf{Outlen}\left[\mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, C)\right] \\
&= \mathsf{Outlen}\left[\mathsf{FE.Keygen}(\mathsf{msk}, T)\right] + 2^{n-\frac{n}{d}} \cdot \mathsf{Outlen}\left[\mathsf{FE.Enc}(\mathsf{pk}, C_x)\right] \\
&= p(\lambda, s \cdot 2^{\frac{n}{d}}, s) + 2^{n-\frac{n}{d}} \cdot (s \cdot 2^{\frac{n}{d}})^{1-\epsilon} \cdot \mathsf{poly}(\lambda, s) \\
&= 2^{n \cdot \frac{d-1}{d}} \cdot \mathsf{poly}(\lambda, s) + 2^{n-\frac{n}{d}} \cdot (s \cdot 2^{\frac{n}{d}})^{1-\epsilon} \cdot \mathsf{poly}(\lambda, s) \leq 2^{n \cdot (1-\frac{\epsilon}{d})} \cdot \mathsf{poly}(\lambda, s)
\end{aligned}
$$

for some constant $\epsilon > 0$, by the weak sublinear compactness of $\mathsf{FE}$.

**Security.** It follows immediately by the construction that any distinguisher $\mathcal{D}$ that succeeds in distinguishing between the $\mathsf{xi}\mathcal{O}$ obfuscations of any two functionally equivalent circuits with probability noticeably greater than negligible in $\lambda$ can be used to break the security of the underlying functional encryption scheme. $\square$

## 4.2 Weakly Sublinear Compact FE + sFE to Sublinear Compact FE

In this section, we present a transformation from weakly sublinear compact FE and succinct FE to sublinear compact FE. This transformation is inspired by a similar transformation corresponding to randomized encoding schemes rather than FE in [38].

Throughout this section, we let $s = s(\lambda)$, $s' = s'(\lambda)$, $n = n(\lambda)$, $n' = n'(\lambda)$, and $\ell = \ell(\lambda)$. The transformation uses the following building blocks.

- $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Keygen}, \mathsf{FE.Dec})$ is a perfectly correct, weakly sublinear compact, selectively-secure FE scheme for $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$. Let $p_1$ be a fixed polynomial and let $\epsilon > 0$ be a constant such that for any $\mathsf{pk}$ in the support of $\mathsf{FE.Setup}$,

$$\mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] \le p_1(\lambda, n, s),$$
$$\mathsf{Outlen}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] \le s^{1-\epsilon} \cdot p_1(\lambda, n).$$

- $\mathsf{IFE} = (\mathsf{IFE.Setup}, \mathsf{IFE.Enc}, \mathsf{IFE.Keygen}, \mathsf{IFE.Dec})$ is a perfectly correct, long-output succinct, simulation secure FE scheme for a family of circuits $\mathcal{C}^{s',n'} = \{\mathcal{C}'_\lambda\}_{\lambda \in \mathbb{N}}$ where $s' < 2p_1(\lambda, s, n)$ and $n' = n + \lambda$,and all circuits $C \in \mathcal{C}'_\lambda$ have output length $\ell = s^{1-\epsilon} \cdot p_1(\lambda, n)$. Let $p_2$ be a fixed polynomial such that for any $(\mathsf{pk}, \mathsf{msk})$ in the support of $\mathsf{IFE.Setup}$,

$$\mathsf{Time}\left[\mathsf{IFE.Enc}(\mathsf{pk}, m)\right] \le p_2(\lambda, \log(s')) \cdot n' \cdot \ell.$$

The construction utilizes the efficiency properties of the (long-output) succinct scheme $\mathsf{IFE}$ and weakly sublinear compact scheme $\mathsf{FE}$ to get a scheme that enjoys both properties. Concretely, we know that the encryption algorithm of $\mathsf{IFE}$ is succinct in circuit size (both in time and output length) but might be long in the output length of the circuit. $\mathsf{FE}$, on the other hand, is succinct in output length but might require a long encryption time. At a high level, our scheme consists of a functional key using $\mathsf{IFE}$ that outputs a ciphertext for $\mathsf{FE}$. Since the output size of the ciphertext is short, $\mathsf{IFE}$ ciphertexts for it are compact.

**Construction of $\mathsf{FE}'$:**

- $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{FE}'.\mathsf{Setup}(1^\lambda)$ :
  1. Sample $(\mathsf{pk}_1, \mathsf{msk}_1) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{msk}_2) \leftarrow \mathsf{IFE.Setup}(1^\lambda)$.
  2. Let $G = G[\mathsf{pk}_1]$ be the circuit such that $G(m, r) = \mathsf{FE.Enc}(\mathsf{pk}_1, m; r)$.
  3. Generate $\mathsf{sk}_G \leftarrow \mathsf{IFE.Keygen}(\mathsf{msk}_2, G)$ and output $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2)$ and $\mathsf{msk} = (\mathsf{msk}_1, \mathsf{sk}_G)$.
- $\mathsf{ct} \leftarrow \mathsf{FE}'.\mathsf{Enc}(\mathsf{pk}, m)$ :
  1. Sample $r \leftarrow \{0,1\}^\lambda$ and output $\mathsf{ct} \leftarrow \mathsf{IFE.Enc}(\mathsf{pk}_2, (m, r))$.
- $\mathsf{sk} \leftarrow \mathsf{FE}'.\mathsf{Keygen}(\mathsf{msk}, C)$ :
  1. Generate $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}_1, C)$ and output $\mathsf{sk} = (\mathsf{sk}_G, \mathsf{sk}_C)$.
- $y \leftarrow \mathsf{FE}'.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ :
  1. Let $\mathsf{ct}' = \mathsf{IFE.Dec}(\mathsf{sk}_G, \mathsf{ct})$ and output $y = \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct}')$.

**Claim 4.4.** *Assuming the existence of a weakly sublinear compact, perfectly correct FE scheme* $\mathsf{FE}$ *for* $\mathsf{P}$ *and a long-output succinct, perfectly correct, simulation secure FE scheme* $\mathsf{sFE}$ *for* $\mathsf{P}$*, there exists a sublinear compact, perfectly correct FE scheme for* $\mathsf{P}$.

**Proof.** Let $\mathsf{FE}'$ be the result of the above construction. Let $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}} \in \mathsf{P}$. We will show that $\mathsf{FE}'$ is a perfectly correct, sublinear compact, selectively secure FE scheme for $\mathcal{C}^{s,n}$. Let $s = s(\lambda)$ and $n = n(\lambda)$.

**Perfect correctness.** The correctness of $\mathsf{FE}'$ follows directly from that of $\mathsf{sFE}$ and $\mathsf{FE}$. In particular, for any $m \in \{0,1\}^n$, by the perfect correctness of $\mathsf{IFE}$ it holds that $\mathsf{ct}' = \mathsf{IFE.Dec}(\mathsf{sk}_G, \mathsf{ct}) = G(m,r) = \mathsf{FE.Enc}(\mathsf{pk}_1, m; r)$ and therefore, by the perfect correctness of $\mathsf{FE}$, for any $C \in \mathcal{C}_\lambda$, it holds that $\mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct}') = C(m)$.

**Sublinear compactness.** Before analyzing compactness, we discuss the circuit classes for $\mathsf{FE}$ and $\mathsf{IFE}$. We have that $\mathsf{FE}$ is for circuits in $\mathcal{C}^{s,n}$. Regarding $\mathsf{IFE}$, we have that $\mathsf{IFE}$ must be able to generate a functional key for $G$. The circuit $G$ has input $(m,r)$ and runs the $\mathsf{FE}$ encryption algorithm on $m$ with randomness $r$. Thus, $\mathsf{IFE}$ is for the class of circuits of size $s' = 2p_1(\lambda, s, n)$ (the size of the $\mathsf{FE}$ encryption circuit) and input length $n' = n + \lambda$, because we can assume without loss of generality that the $\mathsf{FE}$ encryption algorithm takes randomness of length $\lambda$ (and may apply a PRG to stretch this polynomially). Moreover, the output length $\ell$ of $G$ is the output length of $\mathsf{FE.Enc}$, which is $s^{1-\epsilon} \cdot \mathsf{poly}(\lambda, n)$.

We now show sublinear compactness. For sufficiently large $\lambda$, any $C \in \mathcal{C}_\lambda$, and $m \in \{0,1\}^n$, by the long-output succinctness of $\mathsf{IFE}$ and the weak sublinear compactness of $\mathsf{FE}$ we have that

$$\begin{aligned}
\mathsf{Time}\left[\mathsf{FE}'.\mathsf{Enc}(\mathsf{pk}, m)\right] &\le \mathsf{Time}\left[\mathsf{IFE.Enc}(\mathsf{pk}_2, m)\right] = p_2(\lambda, \log(s')) \cdot n' \cdot \ell \\
&= p_2(\lambda, \log(2p_1(\lambda, s, n))) \cdot (n+\lambda) \cdot s^{1-\epsilon} \cdot p_1(\lambda, n) \\
&= p_3(\lambda, n, \log(s)) \cdot s^{1-\epsilon} \le p_4(\lambda, n) \cdot s^{1-\epsilon}
\end{aligned}$$

for some polynomials $p_3$ and $p_4$, where $\mathsf{pk} = \mathsf{pk}_2$ such that $(\mathsf{pk}_2, \mathsf{msk}_2) \leftarrow \mathsf{IFE.Setup}(1^\lambda)$.

**Security.** We show selective security via a sequence of hybrid games.

- $\mathbf{Hyb^0(\lambda)}$: This is an honest encryption of $m_0$. In particular, $\mathsf{ct}$ is generated by $\mathsf{IFE.Enc}(\mathsf{pk}_2, (m_0, r))$ as the output of $\mathsf{FE}'.\mathsf{Enc}(\mathsf{pk}, m_0)$. The output of this hybrid is $((\mathsf{pk}_1, \mathsf{pk}_2), C, m_0, m_1, (\mathsf{sk}_G, \mathsf{sk}_C), \mathsf{ct})$.
- $\mathbf{Hyb^1(\lambda)}$: This hybrid is formed from the previous hybrid by simulating the ciphertext $\mathsf{ct}$. In particular, if $\mathcal{S}$ is the simulator for $\mathsf{IFE}$, then $\mathsf{ct}$ is calculated as $\mathsf{ct} \leftarrow \mathcal{S}(\mathsf{mpk}_2, \mathsf{sk}_G, G, G(m_0, r))$. This is indistinguishable from the previous hybrid by the simulation security of $\mathsf{IFE}$.
- $\mathbf{Hyb^2(\lambda)}$: This hybrid is formed from the previous hybrid by changing $\mathsf{ct}$ to be computed as $\mathsf{ct} \leftarrow \mathcal{S}(\mathsf{mpk}_2, \mathsf{sk}_C, G, G(m_1, r))$, that is, $\mathcal{S}$ receives $G(m_1, r)$ instead of $G(m_0, r)$. This is indistinguishable from the previous hybrid by the security of $\mathsf{FE}$.
- $\mathbf{Hyb^3(\lambda)}$: This hybrid is formed from the previous hybrid by changing $\mathsf{ct}$ to be computed as $\mathsf{ct} \leftarrow \mathsf{IFE.Enc}(\mathsf{pk}_2, (m_1, r))$. This is indistinguishable from the previous hybrid by the simulation security of $\mathsf{IFE}$.

In Appendix C, we show that each pair of neighboring hybrids are computationally indistinguishable, which implies the security of $\mathsf{FE}'$.

□

# Acknowledgments

We thank Zvika Brakerski for answering several questions related to the perfect correctness of various LWE-based schemes.

## Disclaimer

no liability, for the completeness, accuracy or reliability of information, or the legal, compliance, financial, tax or accounting effects of matters contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

## References


[1] Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing. pp. 284–293. ACM (1997)

[2] Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: From selective to adaptive security in functional encryption. In: Advances in Cryptology - CRYPTO 2015, Proceedings, Part II. pp. 657–677 (2015)

[3] Ananth, P., Jain, A., Sahai, A.: Robust transforming combiners from indistinguishability obfuscation to functional encryption. In: Advances in Cryptology - EUROCRYPT 2017. pp. 91–121 (2017)

[4] Ananth, P., Jain, A.: Indistinguishability obfuscation from compact functional encryption. In: Advances in Cryptology - CRYPTO 2015. pp. 308–326 (2015)

[5] Ananth, P., Sahai, A.: Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In: Advances in Cryptology - EUROCRYPT 2017. vol. 10210, pp. 152–181 (2017)

[6] Applebaum, B., Ishai, Y., Kushilevitz, E.: Computationally private randomizing polynomials and their applications. Computational Complexity 15(2), 115–162 (2006), https://doi.org/10.1007/s00037-006-0211-8

[7] Asharov, G., Ephraim, N., Komargodski, I., Pass, R.: On the complexity of compressing obfuscation. In: CRYPTO (3). Lecture Notes in Computer Science, vol. 10993, pp. 753–783. Springer (2018)

[8] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. J. ACM 59(2), 6:1–6:48 (2012)

[9] Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. SIAM J. Comput. 37(2), 380–400 (2007)

[10] Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Symposium on Theory of Computing, STOC. pp. 505–514 (2014)

[11] Bitansky, N., Nishimaki, R., Passelègue, A., Wichs, D.: From cryptomania to obfustopia through secret-key functional encryption. In: Theory of Cryptography Conference. pp. 391–418 (2016)

[12] Bitansky, N., Paneth, O.: Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC. pp. 401–427 (2015)

[13] Bitansky, N., Paneth, O., Wichs, D.: Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In: TCC 2016-A. pp. 474–502 (2016)

[14] Bitansky, N., Vaikuntanathan, V.: Indistinguishability obfuscation from functional encryption. In: IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015. pp. 171–190 (2015)

[15] Bitansky, N., Vaikuntanathan, V.: Indistinguishability obfuscation: From approximate to exact. In: TCC 2016-A. pp. 67–95 (2016)

[16] Bitansky, N., Vaikuntanathan, V.: A note on perfect correctness by derandomization. In: Advances in Cryptology - EUROCRYPT 2017. Lecture Notes in Computer Science, vol. 10211, pp. 592–606 (2017)

[17] Blum, M.: Coin flipping by telephone. In: Advances in Cryptology: - CRYPTO. pp. 11–15 (1981)

[18] Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of eliminating errors in cryptographic computations. J. Cryptology 14(2), 101–119 (2001)

[19] Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT. pp. 533–556 (2014)

[20] Boneh, D., Sahai, A., Waters, B.: Functional encryption: a new vision for public-key cryptography. Commun. ACM 55(11), 56–64 (2012)

[21] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) 6(3), 13 (2014)

[22] Dwork, C., Naor, M.: Zaps and their applications. SIAM J. Comput. 36(6), 1513–1543 (2007)

[23] Dwork, C., Naor, M., Reingold, O.: Immunizing encryption schemes from decryption errors. In: Advances in Cryptology - EUROCRYPT 2004. vol. 3027, pp. 342–360 (2004)

[24] Fürer, M., Goldreich, O., Mansour, Y., Sipser, M., Zachos, S.: On completeness and soundness in interactive proof systems. Advances in Computing Research 5, 429–442 (1989)

[25] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013. pp. 40–49. IEEE Computer Society (2013)

[26] Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the ajtai-dwork cryptosystem. In: Advances in Cryptology - CRYPTO. pp. 105–111 (1997)

[27] Goldreich, O., Mansour, Y., Sipser, M.: Interactive proof systems: Provers that never fail and random selection (extended abstract). In: 28th Annual Symposium on Foundations of Computer Science, FOCS. pp. 449–461. IEEE Computer Society (1987)

[28] Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT. pp. 578–602 (2014)

[29] Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Symposium on Theory of Computing Conference, STOC'13, 2013. pp. 555–564 (2013)

[30] Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Advances in Cryptology - CRYPTO 2012. vol. 7417, pp. 162–179 (2012)

[31] Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Symposium on Theory of Computing Conference, STOC'13. pp. 545–554 (2013), `http://doi.acm.org/10.1145/2488608.2488677`

[32] Goyal, R., Hohenberger, S., Koppula, V., Waters, B.: A generic approach to constructing and proving verifiable random functions. In: Theory of Cryptography Conference. pp. 537–566. Springer (2017)

[33] Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: Advances in Cryptology - CRYPTO. pp. 97–111 (2006)

[34] Holenstein, T., Renner, R.: One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In: Advances in Cryptology - CRYPTO 2005. vol. 3621, pp. 478–493 (2005)

[35] Impagliazzo, R., Wigderson, A.: $P = BPP$ if $E$ requires exponential circuits: Derandomizing the XOR lemma. In: Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing. pp. 220–229. ACM (1997)

[36] Komargodski, I., Moran, T., Naor, M., Pass, R., Rosen, A., Yogev, E.: One-way functions and (im)perfect obfuscation. In: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014. pp. 374–383 (2014)

[37] Lin, H., Pass, R., Seth, K., Telang, S.: Indistinguishability obfuscation with non-trivial efficiency. In: Public-Key Cryptography - PKC 2016. pp. 447–462 (2016)

[38] Lin, H., Pass, R., Seth, K., Telang, S.: Output-compressing randomized encodings and applications. In: TCC 2016-A. pp. 96–124 (2016)

[39] Lin, H., Tessaro, S.: Amplification of chosen-ciphertext security. In: Advances in Cryptology - EUROCRYPT 2013. pp. 503–519 (2013)

[40] Lindell, Y., Pinkas, B.: A proof of security of yaos protocol for two-party computation. Journal of Cryptology 22(2), 161–188 (2009)

[41] Naor, M.: On cryptographic assumptions and challenges. In: Advances in Cryptology - CRYPTO. pp. 96–109 (2003)

[42] Nisan, N., Wigderson, A.: Hardness vs. randomness (extended abstract). In: 29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988. pp. 2–11 (1988)

[43] Nisan, N., Wigderson, A.: Hardness vs randomness. J. Comput. Syst. Sci. 49(2), 149–167 (1994)

[44] O'Neill, A.: Definitional issues in functional encryption. IACR Cryptology ePrint Archive 2010, 556 (2010)

[45] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: Proceedings of the forty-first annual ACM symposium on Theory of computing. pp. 333–342. ACM (2009)

[46] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM 56(6), 34:1–34:40 (2009)

[47] Shaltiel, R., Umans, C.: Simple extractors for all min-entropies and a new pseudo-random generator. In: 42nd Annual Symposium on Foundations of Computer Science, FOCS. pp. 648–657. IEEE Computer Society (2001)

## A Perfectly Correct Succinct FE

In this section, we review the succinct FE construction of [29]. In [29], they show that the correctness of the scheme relies on the correctness of the underlying building blocks, but they do not explicitly consider perfect correctness. We review this proof, and then verify that these building blocks (in some cases, with minor modifications) have perfectly correct instantiations.

**Theorem A.1** ([29, 37, 30])**.** *Assuming (sub-exponential) hardness of* $\mathsf{LWE}$*, there exists a succinct, single-key functional encryption scheme* $\mathsf{sFE}$ *for* $\mathsf{NC}^1$ *that satisfies (sub-exponential) simulation security.*

*Moreover, for a class of circuits* $\mathcal{C}^{s,n} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$*, the time to encrypt is*

$$\mathsf{Time}\left[\mathsf{Enc}(\mathsf{pk}, m)\right] = n(\lambda) \cdot \mathsf{poly}(\lambda, \log(s(\lambda))).$$

*where* $\mathsf{pk}$ *is in the support of the public keys of* $\mathsf{sFE}$ *and* $m \in \{0,1\}^{n(\lambda)}$.

We proceed with a high level overview of the construction. To construct $\mathsf{sFE}$ for a class of circuits $\mathcal{C}^{s,n,d} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}} \in \mathsf{NC}^1$, the transformation uses the following building blocks:

- $\mathsf{FHE} = (\mathsf{FHE.Keygen}, \mathsf{FHE.Enc}, \mathsf{FHE.Eval}, \mathsf{FHE.Dec})$ is a leveled FHE scheme for $\mathcal{C}^{s,n,d}$. Let $\ell = \ell(\lambda)$ be the length of the FHE ciphertexts.
- $\mathsf{ABE} = (\mathsf{ABE.Setup}, \mathsf{ABE.Keygen}, \mathsf{ABE.Enc}, \mathsf{ABE.Dec})$ is a single-key, two-outcome ABE scheme for the class of predicates $\mathcal{P} = \mathcal{P}_{\mathcal{C}_\lambda,\mathsf{FHE}}$ where

  $$\mathcal{P}_{C_\lambda,\mathsf{FHE}} = \left\{\mathsf{FHE.Eval}^i_C, 1 - \mathsf{FHE.Eval}^i_C : C \in \mathcal{C}_\lambda, i \in [\ell]\right\},$$

  where $\mathsf{FHE.Eval}^i_C$ is a circuit that on input $(\mathsf{hpk}, \psi_1, \ldots, \psi_n)$ computes the $i$th bit of the homomorphic evaluation of $C$ on $\psi_1, \ldots, \psi_n$.
- $\mathsf{Gb} = (\mathsf{Gb.Garble}, \mathsf{Gb.Enc}, \mathsf{Gb.Eval})$ is a garbling scheme that satisfies input- and circuit- privacy, which can be constructed from one-way functions.

**Two-Outcome ABE.** Before reviewing the construction, we give an overview of two-outcome ABE using in this scheme. Two-outcome ABE differs from standard ABE in that the encryption algorithm, in addition to receiving the public key $\mathsf{pk}$ and an attribute $a$, also receives two messages $m_0$ and $m_1$. The correctness requirement is that when decrypting a ciphertext $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, a, m_0, m_1)$ under a key $\mathsf{sk}_P = \mathsf{Keygen}(\mathsf{msk}, P)$, if $P(a) = 0$, then the result of decryption is $m_0$, and otherwise if $P(a) = 1$, it is $m_1$. In [28], they give a construction of two-outcome ABE from standard ABE. Moreover, the correctness of the new scheme follows directly from the correctness of the underlying ABE scheme. See [28] for details.

**The construction.** We now present the high level construction of $\mathsf{sFE}$. The master public and secret keys consist of $\ell$ master public keys $\{\mathsf{pk}_i\}_{i\in[\ell]}$ and secret keys $\{\mathsf{msk}_i\}_{i\in[\ell]}$, respectively, for the ABE scheme. A functional key for a circuit $C$ consists of ABE keys $\mathsf{sk}_i$ generated using $\mathsf{msk}_i$ for the predicate $\mathsf{FHE.Eval}^i_C$ for each $i \in [\ell]$. To encrypt a message $x$, they generate an FHE key-pair $(\mathsf{hpk}, \mathsf{hsk})$ and encrypt $x$ one bit at a time using $\mathsf{FHE}$ to obtain a vector $\psi$ of $\mathsf{FHE}$ ciphertexts. Then, they garble the FHE decryption algorithm $\mathsf{FHE.Dec}(\mathsf{hsk}, \cdot)$ to obtain a garbled circuit $\Gamma$ and labels $L^0_i, L^1_i$ for each $i \in [\ell]$. To enable evaluating the garbled circuit with the correct labels for $x$, they create ABE ciphertexts $\mathsf{ct}_i$ generated using $\mathsf{pk}_i$ of $L^0_i, L^1_i$ under the attribute $(\mathsf{hpk}, \psi)$ for each $i \in [\ell]$, and output these ciphertexts as well as the garbled circuit $\Gamma$. To decrypt, they simply decrypt each ABE ciphertext $\mathsf{ct}_i$ under $\mathsf{sk}_i$ to get the corresponding labels for $\mathsf{FHE.Eval}(\mathsf{hpk}, \psi)$, which thus enable them to evaluate the garbled circuit and obtain $C(x)$.

**Proposition A.2** ([29]). *If* $\mathsf{FHE}$*,* $\mathsf{ABE}$*, and* $\mathsf{Gb}$ *all satisfy perfect correctness, then* $\mathsf{sFE}$ *is a perfectly correct succinct FE scheme.*

**Proof.** This follows directly from the correctness of the underlying primitives. By the correctness of ABE, it holds that if $\mathsf{ct}_i \leftarrow \mathsf{ABE.Enc}(\mathsf{pk}_i, (\mathsf{hpk}, \psi), L_i^0, L_i^1)$ and $\mathsf{sk}_i \leftarrow \mathsf{ABE.Keygen}(\mathsf{msk}_i, \mathsf{FHE.Eval}_C^i)$, then $\mathsf{ABE.Dec}(\mathsf{sk}_i, \mathsf{ct}_i) = L_i^{d_i}$ where $d_i = \mathsf{FHE.Eval}_C^i(\mathsf{hpk}, \psi)$ for all $i \in [\ell]$. Then, by the correctness of $\mathsf{Gb}$, evaluating the garbled circuit $\Gamma$ using labels $L_i^{d_i}$ gives $\mathsf{FHE.Dec}(\mathsf{hsk}, \mathsf{FHE.Eval}(\mathsf{hpk}, C, \psi))$, which evaluates to $C(x)$ by the correctness of $\mathsf{FHE}$. □

## A.1 Perfectly Correct FHE

Brakerski et al [21] show a perfectly correct, leveled FHE scheme. In particular, they observe that as long as the initial noise from LWE is bounded, and the noise does not wrap around the modulus, correctness is preserved. They show how to set the parameters to ensure that the latter holds, and as discussed in the preliminaries, we can assume that the LWE noise comes from a bounded distribution, thus ensuring the former.

**Theorem A.3** ([21], restatment in [29]). *Assuming hardness of LWE, for every* $n$ *and polynomial* $d = d(n)$*, there is a perfectly correct* $d$*-leveled FHE scheme for encrypting* $n$ *bits, such that given any circuit* $C$ *of size* $s'$ *and depth* $d'$*, the circuit for homomorphic evaluation of* $C$ *has size* $s' \cdot \mathsf{poly}(\lambda, n, d)$ *and depth* $d' \cdot \mathsf{poly}(\log(n), \log(d))$*.*

*Moreover, assuming sub-exponential hardness of LWE, the FHE scheme is sub-exponentially secure.*

## A.2 Perfectly Correct Garbled Circuits

In this section, we overview the correctness of Yao's garbled circuits, as presented in [40].

**Theorem A.4** ([40], restatement in [29]). *Assuming (sub-exponentially secure) one-way functions, there exists a (sub-exponentially secure) perfectly correct garbled circuit scheme for* $\mathsf{P}$*.*

The $\mathsf{sFE}$ construction requires a garbled circuit that satisfies input-privacy and circuit-privacy. Thus, they use Yao's garbled circuit, because, as shown in [40], there is an instantiation that satisfies these specific security requirements needed to construct $\mathsf{sFE}$.

Regarding perfect correctness, the Lindell et al [40] construction $\mathsf{Gb}$ is correct with all but negligible probability. Recall that in Yao's garbled circuit, to garble a function represented by a circuit $C$, each wire $w$ of $C$ is assigned two keys $k_0^w$ and $k_1^w$ for a symmetric encryption scheme, such that key $k_b^w$ corresponds to a value of $b$ on wire $w$. Then, for a gate $g$ with input wires $u, v$ and output wire $w$, four ciphertexts are created, such that for every pair $a, b \in \{0, 1\}$, there is an encryption $\mathsf{ct}_{a,b}^g$ of $k_w^{g(a,b)}$ under keys $k_u^a$ and $k_v^b$. Given labels corresponding to the inputs of the circuit, one can simply decrypt the corresponding ciphertexts at each gate to reveal the keys for the next gate, and following this pattern, evaluate the circuit.

There are two sources of potential error in $\mathsf{Gb}$. As mentioned in [40], these can both be avoided, giving a perfectly correct garbled circuit scheme. The first source of error is a decryption error among one of the ciphertexts, which would be caused if the underlying symmetric encryption scheme did not satisfy perfect correctness. This can be solved by using an encryption scheme that does not have decryption errors, such as the standard symmetric-key encryption construction from PRFs which is used in [40]. The second source of error is that for some gate $g$, there exist two different ciphertexts $\mathsf{ct}_{a,b}^g$ and $\mathsf{ct}_{a',b'}^g$ that successfully decrypt under the same key. As shown in [40], the probability of this event is negligible, and there are several possible solutions to mitigate this problem, such as assigning randomly permuted indices to each wire, such that decryption at each gate reveals exactly which ciphertext should be decrypted at the next gate. See [40] for details.

### A.3 Bootstrapping Succinct FE

The [29] scheme gives a succinct FE scheme for $\mathsf{NC}^1$. This is because the $\mathsf{sFE}$ ciphertext length depends on the length of $\mathsf{ABE}$ ciphertexts, which depend polynomially on the depth of the circuits representing the predicates it supports. Nevertheless, $\mathsf{sFE}$ can be bootstrapped to $\mathsf{P}$ through the transformation in [2]. While we note that this transformation is in the case of secret-key FE, it can be applied in the public-key case as well.

At a high level, this transformation uses randomized encodings for circuits [6] to bootstrap $\mathsf{sFE}$ for $\mathsf{NC}^1$ to $\mathsf{sFE}$ for $\mathsf{P}$. In particular, to generate a key for a circuit $C$, they generate a key using $\mathsf{sFE}$ for the circuit which, on input $x, K$, computes the randomized encoding of $C$ on $x$ using randomness derived from the PRF key $K$. To encrypt a message $x$, they encrypt $x$ along with a PRF key. As shown in [2], the correctness of the new scheme follows directly from the correctness of $\mathsf{sFE}$ and the randomized encoding. Moreover, the randomized encoding used is constructed using Yao's garbled circuits with a symmetric-key encryption scheme with decryption in $\mathsf{NC}^1$. Therefore, using the encryption scheme from LWE discussed in Section 2.4 and by Theorem A.4, the randomized encoding scheme has perfect correctness, and thus the resulting succinct FE scheme for $\mathsf{P}$ has perfect correctness.

Moreover, while simulation security was not considered in [2], their construction satisfies it, assuming $\mathsf{sFE}$ is simulation secure. Thus, we obtain the following corollary.

**Corollary A.5.** *Assuming sub-exponential hardness of LWE, if there exists a perfectly correct succinct, sub-exponentially simulation secure FE scheme for* $\mathsf{NC}^1$*, then there exists a perfectly correct, succinct, sub-exponentially simulation secure FE scheme for* $\mathsf{P}$*.*

### A.4 Long-Output Succinct FE

In the above section, we discussed succinct FE, which is, by definition, for functions that output one bit. In this section, we observe that parallel repetition of such a scheme yields a scheme with desirable succinctness properties, which we call long-output succinct FE. This will be useful in our correctness amplification for FE.

**Claim A.6.** *Let $\ell = \ell(\lambda)$ be any polynomial. Let* $\mathsf{sFE}$ *be a succinct, perfectly correct, simulation secure FE scheme for* $\mathsf{P}$*. Then, there exists a perfectly correct, simulation secure FE scheme* $\mathsf{lFE}$ *for any class of circuits in* $\mathsf{P}$ *with $\ell$ output bits, such that the time to encrypt and the time to generate a functional key using* $\mathsf{lFE}$ *are each $\ell$ times larger than those of* $\mathsf{sFE}$*.*

This theorem follows simply by $\ell(\lambda)$ parallel repetitions of $\mathsf{sFE}$ (also described in [29]). The following claim follows from the succinctness bounds given in Theorem A.1, as well as Claim 3.4.

**Claim A.7.** *Assuming hardness of LWE, for every $\ell = \ell(\lambda)$, there is a perfectly correct, simulation secure, FE scheme* $\mathsf{lFE}$ *for any polynomial-size class of circuits $\mathcal{C}^{s,n} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ with $\ell$-bit outputs, such that for any $C \in C_\lambda$ and $m \in \{0,1\}^n$, it holds that the time to encrypt is*

$$\mathsf{Time}\left[\mathsf{FE.Enc}(\mathsf{pk}, m)\right] = \mathsf{poly}(\lambda, \log(s)) \cdot n \cdot \ell$$

*for $(\mathsf{msk}, \mathsf{pk}) \leftarrow \mathsf{lFE.Setup}(1^\lambda)$, $n = n(\lambda)$, $s = s(\lambda)$ and a fixed polynomial* $\mathsf{poly}$*.*

## B Perfect XiO and Succinct FE to Perfect iO

### B.1 From Perfect XiO to Perfect FE

In this section we review the transformation from XiO to weakly sublinear compact FE, due to [37]. We verify that it satisfies perfect correctness if the underlying building blocks satisfy perfect correctness.

**Theorem B.1** ([37]). *Assume the existence of a sub-exponentially secure succinct FE scheme* $\mathsf{sFE}$ *for* $\mathsf{P}$*, and a sub-exponentially-secure XiO scheme* $\mathsf{xi}\mathcal{O}$ *for* $\mathsf{P}^{\log}$*. Then, there exists a sub-exponentially secure, weakly sublinear compact FE scheme* $\mathsf{FE}$ *for* $\mathsf{P}$*.*

At a high level, the construction of $\mathsf{FE}$ is as follows. The key-pair $(\mathsf{pk}, \mathsf{msk})$ is generated as the output of $\mathsf{sFE.Setup}(1^\lambda)$. To encrypt a message $m$ under $\mathsf{pk}$, the $\mathsf{FE}$ encryption algorithm uses $\mathsf{xi}\mathcal{O}$ to obfuscate a circuit $G[\mathsf{pk}, K, m]$, which, on input $i$, outputs an $\mathsf{sFE}$ encryption of $(m, i)$ using randomness derived from the hardcoded PRF key $K$. The functional secret key $\mathsf{sk}_C$ for a circuit $C$ is generated as an $\mathsf{sFE}$ functional secret key for a circuit $C'$ such that on input $(m, i)$, outputs the $i$th bit of $C(m)$. Finally, to decrypt an encryption $\mathsf{ct}$ under $\mathsf{sk}_C$, the decryption algorithm obtains the $i$th bit of the output by evaluating the obfuscated circuit on $i$ to obtain a ciphertext $\mathsf{ct}_i$ and then decrypting $\mathsf{ct}_i$ using $\mathsf{sFE}$ with $\mathsf{sk}_C$. Concatenating these bits for all $i$ gives the output of the decryption.

**Proposition B.2.** *If* $\mathsf{xi}\mathcal{O}$ *and* $\mathsf{sFE}$ *are perfectly correct, then* $\mathsf{FE}$ *is perfectly correct.*

**Proof.** Consider the $\mathsf{FE}$ decryption algorithm. For any circuit $C$ and input $m$, and any random strings $r_{\mathsf{Enc}}, r_{\mathsf{Keygen}}$ we have that

$$\Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)\\ \mathsf{sk}_C = \mathsf{FE.Keygen}(\mathsf{msk}, C; r_{\mathsf{Keygen}})\\ \mathsf{ct} = \mathsf{FE.Enc}(\mathsf{pk}, m; r_{\mathsf{Enc}})\end{array} : \mathsf{FE.Dec}(\mathsf{sk}_C, \mathsf{ct}) = C(m)\right]$$
$$= \Pr\left[\begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{sFE.Setup}(1^\lambda)\\ \mathsf{sk}_C = \mathsf{sFE.Keygen}(\mathsf{msk}, C'; r_{\mathsf{Keygen}})\\ \widetilde{G} = \mathsf{xi}\mathcal{O}.\mathsf{Obf}(1^\lambda, G; r_{\mathsf{Enc}})\\ \forall i, \mathsf{ct}_i = \mathsf{xi}\mathcal{O}.\mathsf{Eval}(\widetilde{G}, i)\end{array} : \begin{array}{c}\forall i\ \mathsf{ct}_i = \mathsf{sFE.Enc}(\mathsf{pk}, (m, i))\\ \wedge\ \mathsf{sFE.Dec}(\mathsf{sk}_C, \mathsf{ct}_i) = C(m)_i\end{array}\right].$$

By using a union bound over all $i$ and invoking the perfect correctness of $\mathsf{sFE}$ and $\mathsf{xi}\mathcal{O}$, it holds that $\mathsf{FE}$ satisfies perfect correctness. □

By combining Theorem B.1 and Proposition B.2, we obtain the following.

**Claim B.3** ([37]). *Assume the existence of a sub-exponentially secure, perfectly correct, succinct FE scheme* $\mathsf{sFE}$ *for* $\mathsf{P}$*, and a sub-exponentially-secure, perfectly correct XiO scheme* $\mathsf{xi}\mathcal{O}$ *for* $\mathsf{P}^{\log}$*. Then, there exists a sub-exponentially secure, perfectly correct, weakly sublinear compact FE scheme* $\mathsf{FE}$ *for* $\mathsf{P}$*.*

## B.2 From Perfect FE to Perfect RE

In this section, we discuss transformations from FE to randomized encodings (RE) for Turing machines in the CRS model [38]. A randomized encoding enables encoding a machine $\Pi$ and an input $x$ together to obtain $\hat{\Pi}_x$, such that $\hat{\Pi}_x$ can be evaluated to obtain $\Pi(x)$ (or more specifically, the first $\ell$ bits of $\Pi(x)$ when executed for $T$ steps, where $\ell$ and $T$ are known). The security of RE is that $\hat{\Pi}_x$ does not reveal anything other than $\Pi(x)$, and is formalized using a simulation-based definition. For more detail on the definitions of RE used in this section, see [38].

We verify that the FE to RE transformation of [38] preserves perfect correctness. Looking ahead, we will apply this to both $\mathsf{sFE}$ and $\mathsf{FE}$, and then combine the resulting RE schemes to obtain an RE scheme that is sublinearly compact. We begin by overviewing the FE to RE transformation.

**Theorem B.4** ([38]). *If there exists a sub-exponentially secure PRG and a succinct (resp. weakly sublinear compact) sub-exponentially secure FE scheme for* $\mathsf{P}$*, then there exists a succinct (resp. weakly sublinear compact) RE scheme for Turing machines in the CRS model with sub-exponential simulation security.*

At a high level, the transformation is as follows. The setup algorithm generates $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and generates a string $c$ (which is irrelevant to our analysis of correctness). Then, letting $U$ be the universal circuit that on input $(\Pi, x)$ runs $\Pi$ for $T$ steps and outputs the first $\ell$ bits, it defines the circuit $C_{U,c}$ such that $C_{U,c}(\Pi, x, s', b)$ outputs $U(\Pi, x)$ when $b = 0$ and $c \oplus \mathsf{PRG}(s')$ when $b = 1$. Finally, it outputs $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, C_{U,c})$ as the CRS and $\mathsf{pk}$ as the public key. The encoding algorithm, on input $(\mathsf{pk}, \Pi, x)$ simply outputs the encryption $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, (\Pi, x, 0^\lambda, 0))$. The evaluation algorithm, on input $(\mathsf{ct}, \mathsf{sk}_C)$, outputs the corresponding decryption $\mathsf{FE.Dec}(\mathsf{ct}, \mathsf{sk}_C)$.

**Proposition B.5.** *If* $\mathsf{FE}$ *is perfectly correct, then* $\mathsf{RE}$ *is perfectly correct.*

**Proof.** For any TM $\Pi$, and input $x$, the $\mathsf{RE}$ evaluation algorithm is simply outputs the decryption $y = \mathsf{FE.Dec}(\mathsf{ct}, \mathsf{sk}_C)$ where $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{pk}, (\Pi, x, 0^\lambda, 0))$ and $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}, C_{U,c})$. Therefore, by the perfect correctness of $\mathsf{FE}$, it holds that $y = C_{U,c}(\Pi, x, 0^\lambda, 0) = U(\Pi, x) = \Pi(x)$. Therefore, $\mathsf{RE}$ is perfectly correct. □

Given a succinct RE and weakly sublinear compact RE scheme, they are then used to construct a sublinearly compact RE scheme.

**Theorem B.6** ([38])**.** *Assume the existence of PRGs. If there is a succinct RE scheme and a weakly sublinear compact RE scheme for Turing machines, both in the CRS model, then there is a sublinearly compact randomized encoding scheme for Turing machines in the CRS model.*

Let $\mathsf{RE}_1$ be the succinct RE scheme and let $\mathsf{RE}_2$ be the weakly sublinear compact RE scheme. To encode a Turing machine $\Pi$ and input $x$, the construction generates an $\mathsf{RE}_1$ encoding of the $\mathsf{RE}_2$ encoding circuit with $\Pi$ hardcoded inside. Then, to evaluate, it simply evaluates the outer encoding to obtain the inner encoding, and evaluates the inner encoding to obtain the output. Perfect correctness follows from the underlying correctness of $\mathsf{RE}_1$ and $\mathsf{RE}_2$.

**Proposition B.7.** *If* $\mathsf{RE}_1$ *and* $\mathsf{RE}_2$ *are perfectly correct, then* $\mathsf{RE}$ *is a perfectly correct RE scheme.*

By combining Theorems B.4 and B.6 with Propositions B.5 and B.7, we obtain the following.

**Claim B.8** ([38])**.** *Assuming the existence of sub-exponentially secure, succinct, perfectly correct FE and sub-exponentially secure, weakly sublinear compact, perfectly correct FE, there exists a perfectly correct, sublinearly compact RE scheme for Turing machines in the CRS model with sub-exponential simulation security.*

## B.3 Perfect RE to Perfect iO

In this section, we verify that the RE to iO transformation of [38] preserves perfect correctness. Given a sublinearly compact RE scheme, the last step is to apply the [38] transformation to obtain iO.

**Theorem B.9** ([38])**.** *If there exists a sublinearly compact RE scheme in the CRS model with sub-exponential simulation security, then there exists an iO scheme for* $\mathsf{P}$.

The construction is as follows. At a high level, to obfuscate a circuit $C$ with $n$ inputs, the obfuscation algorithm creates an encoding of a machine $\Pi_\perp$ corresponding to the empty string, such that evaluating it produces an encodings of machines $\Pi_0$ and $\Pi_1$. More generally, evaluating the encoding of $\Pi_s$ for a string $s$ with $|s| < n$ produces an encoding of $\Pi_{s0}$ and an encoding of $\Pi_{s1}$. When the string $s$ has length $n$, evaluating the encoding of $\Pi_s$ produces $C(s)$. More formally, the obfuscation consists of the "top-level" encoding of $\Pi_\perp$ hardwired with $C$, RE public keys for all levels $i > 1$, and a random string to use to derive randomness of all later encodings. The obfuscation also consists of the CRS string for all levels. To evaluate the obfuscation of $C$ on an input $x$, one evaluates the encoding of $\Pi_\perp$ to obtain an encoding of $\Pi_{x_0}$, and then evaluates that the obtain an encoding of $\Pi_{x_0 x_1}$, and so forth, until one obtains an encoding of $\Pi_x$ that can be evaluated to reveal $C(x)$. If $\mathsf{RE}$ has perfect correctness, then the encodings at each level are correct, and thus the iO scheme has perfect correctness.

**Proposition B.10.** *If* $\mathsf{RE}$ *has perfect correctness, then the iO scheme from the above construction has perfect correctness.*

By combining Theorem B.9 and Proposition B.10, we obtain the following.

**Claim B.11** ([38])**.** *If there exists a sublinearly compact, perfectly correct RE scheme in the CRS model with sub-exponential simulation security, then there exists a perfectly correct iO scheme for* $\mathsf{P}$.

Thus, by combining Claims B.3, B.8, and B.11, we obtain Claim 3.3.

# C Additional Proofs from Section 4.2

In this section, we complete the proof of security of Claim 4.4 by showing that each pair of neighboring hybrids are computationally indistinguishable.

**Claim C.1.** *For any PPT* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}$ *such that* $|\Pr\left[\mathcal{A}(\mathsf{Hyb}^0(\lambda)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{Hyb}^1(\lambda) = 1)\right]| \leq \mathsf{negl}(\lambda)$.

**Proof.** Suppose for the sake of contradiction that there exists an adversary $\mathcal{A}$ and a polynomial $p$ such that $\mathcal{A}$ distinguishes between $\mathsf{Hyb}^0(\lambda)$ and $\mathsf{Hyb}^1(\lambda)$ with probability $\frac{1}{p(\lambda)}$. Then, we can construct an adversary $\mathcal{B}$ that breaks the simulation security of $\mathsf{IFE}$ as follows.

Given $m_0, m_1, C$, the adversary $\mathcal{B}$ samples $(\mathsf{pk}_1, \mathsf{msk}_1) \leftarrow \mathsf{FE.Setup}(1^\lambda)$, samples $r \leftarrow \{0,1\}^\lambda$, and generates $G = G[\mathsf{pk}_1]$. Then, $\mathcal{B}$ interacts with a challenger for either $\mathsf{Exp}^{\mathsf{real}}_{\mathsf{FE},\mathcal{B}}(\lambda)$ or $\mathsf{Exp}^{\mathsf{ideal}}_{\mathsf{FE},\mathcal{B},\mathcal{S}}(\lambda)$, where $\mathcal{B}$ sets the challenge message as $(m_0, r)$ and the key generation query as $G$, and in turn receives the $\mathsf{IFE}$ public key $\mathsf{pk}_2$, the functional key $\mathsf{sk}_G$, and the challenge ciphertext $\mathsf{ct}^\star$.

$\mathcal{B}$ then generates $\mathsf{sk}_C \leftarrow \mathsf{FE.Keygen}(\mathsf{msk}_1, C)$ and sends $((\mathsf{pk}_1, \mathsf{pk}_2), C, m_0, m_1, (\mathsf{sk}_G, \mathsf{sk}_C), \mathsf{ct}^\star)$ to $\mathcal{A}$. Observe that if $\mathsf{ct}^\star$ is generated as in $\mathsf{Exp}^{\mathsf{real}}_{\mathsf{FE},\mathcal{B}}(\lambda)$, then the input to $\mathcal{A}$ is distributed according to $\mathsf{Hyb}^0(\lambda)$ and otherwise is distributed according to $\mathsf{Hyb}^1(\lambda)$. Therefore, the distinguishing advantage of $\mathcal{A}$ translates into the advantage of $\mathcal{B}$, which contradicts the security of $\mathsf{IFE}$. $\square$

**Claim C.2.** *For any PPT* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}$ *such that* $|\Pr\left[\mathcal{A}(\mathsf{Hyb}^1(\lambda)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{Hyb}^2(\lambda) = 1)\right]| \leq \mathsf{negl}(\lambda)$.

**Proof.** Suppose for the sake of contradiction that there exists an adversary $\mathcal{A}$ and a polynomial $p$ such that $\mathcal{A}$ distinguishes between $\mathsf{Hyb}^1(\lambda)$ and $\mathsf{Hyb}^2(\lambda)$ with probability $\frac{1}{p(\lambda)}$. Then, we can construct an adversary $\mathcal{B}$ that breaks the security of $\mathsf{FE}$ as follows.

Given $m_0, m_1, C$, the adversary $\mathcal{B}$ samples $(\mathsf{pk}_2, \mathsf{msk}_2) \leftarrow \mathsf{IFE.Setup}(1^\lambda)$. Then, $\mathcal{B}$ receives as input $(\mathsf{pk}_1, C, m_0, m_1, \mathsf{sk}_C, \mathsf{ct}^\star)$ as in the $\mathsf{FE}$ security definition, where $\mathsf{pk}_1$ is an $\mathsf{FE}$ public key, $\mathsf{sk}_C$ is an $\mathsf{FE}$ functional key for $C$, and $\mathsf{ct}^\star$ is the challenge ciphertext. Observe that by definition, $C(m_0) = C(m_1)$, so this is a valid challenge for the $\mathsf{FE}$ security game.

$\mathcal{B}$ then sets $G = G[\mathsf{pk}_1]$ and generates $\mathsf{sk}_G \leftarrow \mathsf{IFE.Keygen}(\mathsf{msk}_2, G)$ and $\mathsf{ct} \leftarrow \mathcal{S}(\mathsf{mpk}_2, \mathsf{sk}_G, G, \mathsf{ct}^\star)$. $\mathcal{B}$ then sends $((\mathsf{pk}_1, \mathsf{pk}_2), C, m_0, m_1, (\mathsf{sk}_G, \mathsf{sk}_C), \mathsf{ct})$ to $\mathcal{A}$. Observe that if $\mathsf{ct}^\star$ is an encryption of $m_0$ under uniformly chosen randomness $r$ then $G(m_0, r) = \mathsf{FE.Enc}(\mathsf{pk}_1, m_0; r) = \mathsf{ct}^\star$ and thus the input that $\mathcal{A}$ receives is distributed exactly as the output of $\mathsf{Hyb}^1(\lambda)$, and otherwise, if $\mathsf{ct}^\star$ is an encryption of $m_1$ under randomness $r$, then $G(m_1, r) = \mathsf{FE.Enc}(\mathsf{pk}_1, m_1; r) = \mathsf{ct}^\star$ and thus the input to $\mathcal{A}$ is distributed exactly as the output of $\mathsf{Hyb}^2(\lambda)$. Therefore, the distinguishing advantage of $\mathcal{A}$ translates into the advantage of $\mathcal{B}$, which contradicts the security of $\mathsf{FE}$. $\square$

**Claim C.3.** *For any PPT $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}$ *such that* $\left|\Pr\left[\mathcal{A}(\mathsf{Hyb}^2(\lambda)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{Hyb}^3(\lambda) = 1)\right]\right| \leq \mathsf{negl}(\lambda)$.

**Proof.** This proof is analogous to the proof that $\mathsf{Hyb}^0(\lambda) \approx \mathsf{Hyb}^1(\lambda)$, with $m_0$ replaced by $m_1$. $\square$