2015 IEEE 56th Annual Symposium on Foundations of Computer Science

Indistinguishability Obfuscation from Functional Encryption

*Nir Bitansky and †Vinod Vaikuntanathan
MIT CSAIL
{*nirbitan,vinodv*}*@csail.mit.edu*

Abstract

Indistinguishability obfuscation (IO) is a tremendous notion, powerful enough to give rise to almost any known cryptographic object. So far, candidate IO constructions were based on specific assumptions on algebraic objects called multi-linear graded encodings.

We present a generic construction of indistinguishability obfuscation from public-key functional encryption with succinct ciphertexts and sub-exponential security. This shows the equivalence of indistinguishability obfuscation and public-key functional encryption, a primitive that has so far seemed to be much weaker, lacking the power and the staggering range of applications of indistinguishability obfuscation.

As an application, we obtain a new candidate IO construction based on the functional encryption scheme of Garg, Gentry, Halevi, and Zhandry [Eprint 14] under their assumptions on multi-linear graded encodings. We also show that, under the Learning with Errors assumptions, our techniques imply that any indistinguishability obfuscator can be converted to one where obfuscated circuits are of linear size in the size of the original circuit plus a polynomial overhead in its depth.

Our reduction highlights the importance of ciphertext succinctness in functional encryption schemes, which we hope will serve as a pathway to new IO constructions based on solid cryptographic foundations.

I. Introduction

Program obfuscation, aiming to turn programs into "unintelligible" ones while preserving functionality, has been a holy grail in cryptography for over a decade. Rather unfortunately, the most natural and intuitively appealing notion of obfuscation, namely *virtual-black-box* (VBB) obfuscation [1], was shown to have strong limitations [1, 2, 3]. Furthermore, except for very restricted function classes, no candidate construction with any form of meaningful security was known.

This changed dramatically with recent breakthrough results. First, Garg, Gentry, Halevi, Raykova, Sahai and Waters [4] demonstrated a candidate obfuscation algorithm for all circuits, and conjectured that it satisfies an apparently weak notion of *indistinguishability obfuscation* (IO) [1, 5], requiring only that the obfuscations of any two circuits of the same size and functionality are computationally indistinguishable. Since then, a sequence of works, pioneered by Sahai and Waters [6], have demonstrated that IO is not such a weak notion after all, leading to a plethora of applications and even resolving long-standing open problems. The number of cryptographic primitives that we do not know how to construct from IO is small and dwindling fast.[1]

The tremendous power of IO also begets its reliance on strong and untested computational assumptions. Despite significant progress [8, 9], all known IO constructions [4, 8, 10, 11, 9, 12, 13] are still based on the hardness of little-studied problems on multi-linear maps [14]. Thus, an outstanding foundational question in cryptography is:

Can we base indistinguishability obfuscation on strong cryptographic foundations?

*Supported in part by NSF Grants CNS-1350619 and CNS-1414119.
†Supported in part by DARPA Grant number FA8750-11-2-0225, an Alfred P. Sloan Research Fellowship, Qatar Computing Research Institute, Microsoft Faculty Fellowship, and a Steven and Renee Finn Career Development Chair from MIT.

[1]Strictly speaking, we need the assumption that IO exists, plus a very mild (and minimal) complexity-theoretic assumption that NP $\neq$ ioBPP [7].

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.20

A. This Work

In this work, we make progress in the above direction, showing how to construct indistinguishability obfuscation from an apparently weaker primitive: *public-key functional encryption*. In a functional encryption scheme [15, 16, 17, 18], the owner of a master secret key MSK can produce functional keys FSK_f for functions f (represented as circuits throughout this paper). Given an encryption of an input x computed using the master public key PK and the functional key FSK_f, anyone can compute $f(x)$, but nothing more about x itself.

In the past few years, functional encryption (FE) schemes with different efficiency and security features were constructed from various computational assumptions. A central measure of interest (in general and in the specific context of this work) is the size of ciphertexts, or more generally the encryption time. Here the ideal requirement is that the time to encrypt depends only on the underlying plaintext x, but this requirement may be relaxed in several meaningful ways, such as allowing dependence on the size of outputs, the number of generated functional keys, or the size of the circuit computing the function.

Functional encryption, on the face of it, seems much less powerful than IO and sure enough, it has not had nearly as many applications. In our eyes, IO derives its power from the fact that it allows *anyone* to compute meaningfully with a hidden object (say, a circuit) with no additional help. In contrast, FE does allow us to encrypt circuits[2] but to evaluate the circuit on an input, one needs a secret key associated to the input! Not surprisingly, the power of FE seems to be limited to achieving a notion of "obfuscation on a leash" or "token-based obfuscation" [19].

Rather surprisingly, we show:

Theorem I.1 (informal)**.** *Assuming the existence of a sub-exponentially secure public-key functional encryption scheme for all circuits, where encryption time is polynomial in the input-size and sub-linear in the circuit-size, there exists indistinguishability obfuscation for all circuits.*

Furthermore, in the above theorem, it suffices to start from a scheme that supports only a single-key and satisfies a mild selective-security indistinguishability-based guarantee. Assuming (sub-exponential) puncturable pseudo-random functions in NC^1, we can further relax the above to allow the encryption to also depend exponentially on circuit-depth.

We also show that the requirement for sublinear dependence on circuit size can be traded, when moving to multi-key functional encryption schemes, with sublinear dependence on the number of derived keys. We do this by showing a generic transformation from the latter to the former (which we find to be of independent interest). As a corollary of this transformation, relying on the recent functional encryption scheme of Garg et al. [20], we obtain a new IO candidate constructions whose security is based on the same assumptions on multi-linear graded encodings (in their subepxonential version). The scheme can be instantiated based on the recent graded encodings of [21].

Corollary I.2. *Under a subexponential variant of the assumptions in [20] on multi-linear graded encodings, there exists an IO construction.*

Another corollary the follows as a simple case of our technique and of previous results on FE with succinct keys [22] is that obfuscation size can always be reduced to linear in the size circuit plus some overhead in the circuit's depth.

Corollary I.3. *Assuming subexponential LWE and IO, there exists IO such that an obfuscation of any circuit C is of size $2|C| + \mathrm{poly}(n, \mathsf{dep}(C), \lambda)$.*

[2]Note that given an FE for a sufficiently expressive class, we can switch the roles of circuits and inputs, going through a universal circuit.

Interpretation: Functional encryption schemes satisfying the ciphertext compactness required in Theorem I.1 are also known based on indistinguishability obfuscation [4, 23] or the stronger notion of differing-inputs obfuscation [24]. Thus, our result establishes the equivalence of functional encryption and IO, up to some sub-exponential security loss. The question of basing IO on more standard assumptions still stands, but is now reduced to improving the state of the art in functional encryption.

It is rather tempting to be pessimistic and to interpret our result as a lower-bound showing that improving functional encryption based on standard assumptions may be very hard, or perhaps straight out impossible. Our take on the result is quite optimistic. First, it may lead to constructions from more standard assumptions on multi-linear graded encodings. Furthermore one may hope that the construction would eventually lead to a construction from more standard assumptions. Indeed, in the past few years, we have seen a remarkable progress in constructions of functional encryption based on standard assumptions [25, 26, 27]. The state of the art scheme based on a standard assumption is that of Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [19] relying on the sub-exponential learning with errors assumption. The construction achieves ciphertext size that only grows polynomially with the circuit output size and depth; thus, for circuits with say a single output bit, ciphertexts may indeed be sub-linear in circuit size, but this will not be the case for circuits with long outputs. Interestingly, the latter construction achieves a strong simulation-based security guarantee, under which sub-linear growth in the output size (let alone circuit-size) is actually impossible [28, 19]. Reducing the dependence on the output (under an indistinguishability-based notion) has been a tantalizing problem. Now this question becomes of central importance in the quest to achieve indistinguishability obfuscation.

In a recent result, Gorbunov, Vaikuntanathan and Wee [29] showed how to construct predicate encryption schemes for all circuits (with a-priori bounded depth) from the sub-exponential learning with errors (LWE) assumption. In their scheme, the ciphertext size is polynomial in the input length and the depth of the circuit, and otherwise independent of the circuit size and output size. A predicate encryption scheme can be interpreted as a functional encryption scheme with a "weak attribute hiding" property (see [30, 31, 29] for more details). Strengthening this to "full attribute hiding" will give us a functional encryption scheme that satisfies the requirements of Theorem I.1, and is yet another frontier in achieving indistinguishability obfuscation from LWE.

B. Main Ideas

A salient feature present in obfuscation and absent in functional encryption is *function-hiding*. Indeed, the standard notion of functional encryption does not guarantee that functional keys do not leak information regarding the underlying function. Moreover, it seems that any meaningful notion of public-key functional encryption that is *also* function-hiding would already imply some sort of obfuscation.

As observed in [19], and generalized in [32], in *private-key* functional encryption schemes, it is always possible to harness the existing message-hiding to also guarantee function-hiding. This can be interpreted as a relaxed form of interactive obfuscation termed in [19] as *token based obfuscation*. Here the function-hiding functional-key FSK_f is seen as an obfuscation of f. In order to evaluate the obfuscation on an input x, the evaluator first needs to request a corresponding token, which is just an encryption of x. The major drawback of course is that encryption is a private-key operation, meaning that tokens cannot be generated publicly and require interaction with the secret-key owner.

While the above solution may still be far in spirit from the desired notion of obfuscation, it does seem to have a certain gain. Intuitively, and thinking for a moment in terms of ideal obfuscation, it seems that rather than obfuscating an entire circuit f, we can first derive a function-hiding key FSK_f (namely, a token-based obfuscation), and then *only obfuscate the encryption algorithm* $\mathsf{Enc}(\cdot)$ (namely, the token generator). Indeed, we may expect $\mathsf{Enc}(\cdot)$ to be less complex, or at least smaller, than the circuit f we started with; in fact, ideally it should depend only on the size of the input x and nothing else.

Our approach attempts to exploit exactly this gain, and can be divided into two high-level steps:

1) *IO from much less IO.* We first show that the above intuition can be fulfilled, not only with ideal obfuscation, but also in the context of indistinguishability obfuscation. Concretely, starting from functional encryption,

we obfuscate, under the IO notion, any function f assuming IO only for a restricted class of smaller circuits that simply generate encrypted inputs.

2) *IO from no IO.* In the second step, with the goal of obfuscating the latter input encryption circuit, we show how the first step can be repeatedly invoked to recursively reduce IO for circuits that encrypt n-bit inputs to functional encryption and IO for circuits that encrypt a smaller number of $n-1$ bits. At the base of this recursion, we only need to obfuscate circuits with a single input bit, which can be done trivially by writing only their respective outputs. Unravelling the recursion we obtain IO for n-bit inputs from FE alone.

Materializing this high-level strategy encounters several difficulties, which eventually lead to our requirement on the efficiency of encryption, to the sub-exponential security requirement, as well as the fact the need for public-key functional encryption (rather than private-key). We next go in more detail into the above two steps and overview these challenges and the way they are dealt with.

Step 1: IO from much less IO: A natural first attempt to achieve our goal is to mimic the ideal solution. Namely, starting from a (private-key) function-hiding functional encryption scheme, to obfuscate any f, generate the functional key FSK_f and add an obfuscation $i\mathcal{O}(\mathsf{Enc}(\mathsf{PK},\cdot))$ of the corresponding encryption circuit. Here encryption is derandomized in the standard way by applying a pseudo-random function (PRF) to the inputs. While this solution would have worked with an ideal notion of obfuscation (e.g. auxiliary-input VBB), it is not clear how to prove its security based solely on IO. In fact, using similar ideas to those in the impossibility result of Barak et al. [1], one can show that we cannot hope to rely *any* private-key (function-hiding) scheme, since there exists such schemes where access to an encryption circuit may lead to a devastating attack.

Our solution will, in fact, rely on public-key functional encryption. Here function-hiding is not be guaranteed; rather, we shall enforce it explicitly in our construction using similar techniques to those used in the private-key setting [32] going back to the classic two-key paradigm [33, 34].

Concretely, to obfuscate f, our obfuscation will once again consist of a functional key FSK_{f^*}, this time to an augmented function f^*, and an obfuscation $i\mathcal{O}(\mathsf{Enc}^*)$ to an augmented encryption algorithm Enc^*. The circuit f^* will consist of two symmetric-key encryptions $\mathsf{CT}_0, \mathsf{CT}_1$, under two independently chosen symmetric keys $\mathsf{SK}_0, \mathsf{SK}_1$, where in the real world both ciphertexts encrypt f. The function f^* expects as input, not only an input x for f, but also a symmetric SK_b, and decrypts the corresponding ciphertext CT_b, and applies the decrypted function to the input x. Accordingly, the encryption algorithm Enc^*, given input x, will generate a (public-key) encryption of x as well as SK_b, where in the real world b will always be set to say 0.

Proving that the above construction is secure can be decoupled into two main ideas that go back to previous works. The first comes from the work of Brakerski and Segev [32]. There the adversary, whose goal is to distinguish between a functional key corresponding to f_0 to one corresponding to a functionally-equivalent f_1, does not ever obtain a circuit that computes the above encryptions. Rather it only views the outputs of this circuit. Let us, in fact, think about a simple case where the distinguisher only obtains a single encryption $\mathsf{Enc}^*(x) := \mathsf{Enc}(\mathsf{PK},(x,\mathsf{SK}_0))$ of some pre-selected input x. In this setting, we can employ a straight forward hybrid argument to show that the functional keys $(\mathsf{FSK}_{f_0^*}, \mathsf{FSK}_{f_1^*})$ corresponding to f_0 and f_1 are indistinguishable. Indeed, relying on the symmetric-key guarantee we can change CT_1 to encrypt f_1, and then relying on the FE guarantee we change $\mathsf{Enc}^*(x)$ to encrypt SK_1 instead of SK_0, indeed we know that $f_0(x) = f_1(x)$. Then, we can symmetrically switch the other cipher to encrypt f_1 and switch the keys again.

The above argument would even hold had the functional encryption scheme been a symmetric-key one. However, going back to reality, we have to deal with a setting where the adversary does not get a single (or a polynomial) number of encryptions, but rather has the actual circuit for generating any encryption. Can we still employ the previous argument? It turns out that, at least if we use public-key functional encryption, the answer is yes.

Concretely, it would suffice to show that we can change the circuit $\mathsf{Enc}^*(x)$ to freely switch between encrypting SK_0 to encrypting SK_1 *for all inputs simultaneously*. Here comes into play another idea that has been used in several recent works and formalized by Canetti, Lin, Tessaro, and Vaikuntanathan [35] as *probabilistic IO*. They show that given two public samplers $C_0(x;r), C_1(x;r)$ such that for any input x $C_0(x)$ and $C_1(x)$ are

computationally indistinguishable, the circuits can be derandomized using a *puncturable PRF* and obfuscated so that their IO obfuscations are indistinguishable. In our setting, we simply apply this argument to the circuits $C_b(x) := \mathsf{Enc}^*(x, \mathsf{SK}_b)$, and make sure to derandomize it with a puncturable PRF. One restriction inherited from this argument is that it only works assuming that the underlying IO and puncturable PRF are both sub-exponentially secure. Also, for the argument to hold indistinguishability is required even given the public circuits, which is the reason for our reliance on public-key functional encryption.

Step 2: IO from no IO: We have reduced the complexity of the circuit to be obfuscated from that of f to that of Enc^*, but how do we obfuscate Enc^*? Here using a similar approach to that above, we show how to reduce the obfuscation of Enc^* that deals with n bit inputs, to an obfuscation of Enc^*_{n-1} that only deals with $n-1$ input bits. We note that, in general, a naive attempt to recursively reduce obfuscation of circuits with n inputs to obfuscation of circuits with $n-1$ inputs, e.g. by obfuscating $C(0,\cdot)$ and $C(1,\cdot)$, would double the size in each step. To avoid blowup, the circuit to be obfuscated in each recursive step should not outgrow the previous circuit. Fortunately, this is exactly the property achieved by the first step, here each time we recurse we only need to obfuscate a circuit that's proportional to the (gradually reducing) input size.

In more detail, we now think of a function f^*_n that expects an input $x \in \{0,1\}^{n-1}$ and outputs two encryptions $\mathsf{Enc}^*(x0), \mathsf{Enc}^*(x1)$. Accordingly, we publish $\mathsf{FSK}_{f^*_n}$ and an obfuscation of Enc^*_{n-1}. This process is than performed recursively, at each level sampling a new instance of the functional encryption scheme as well of the symmetric key encryption, until the last step where Enc^*_1 simply consists of two hardwired encryptions.

Proving that the obfuscation of Enc^*_i is IO assuming that the obfuscation of Enc^*_{i-1} is IO is done using a similar argument to the one used in the first step. The exponential loss due to the use of probabilistic IO accumulates recursively: roughly, the indistinguishability gap δ_i for level i is at most $2^i \cdot \delta_{i-1}$, requiring that all underlying cryptographic primitives are roughly $2^{-\Omega(n^2)}$-secure.

A Recap: Unravelling the recursion, an obfuscation of f eventually consists of n functional keys $\mathsf{FSK}_1, \ldots, \mathsf{FSK}_n$ as well as a single initial pair of encryptions of 0 and 1. The evaluator gradually constructs an encryption of its input x, where at step i it chooses the encryption of $x_1 \ldots x_{i-1}x_i$ between the two encryptions of $x_1 \ldots x_{i-1}0$ and $x_1 \ldots x_{i-1}1$ produced by the previous function decryption step. Then, the next key FSK_i is used to obtain the next two encryptions. Eventually, having constructed the encryption of x, the evaluator decrypts using FSK_n and obtains the actual function value $f(x)$.

Crucially, for this recursion to be efficient and not result in an obfuscation of exponential size, we must require that encrypting is simple enough. Indeed, as long as it only depends on the underlying plaintext, throughout we will have the invariant that the functions Enc^*_i that we recursively obfuscate are always bounded by a fixed polynomial in the total input size n and the security parameter, and accordingly so do the functions f^*_i for which keys are derived (except for the last one which depends on the size of the function f we started from). In the body, we show that we may in fact allow the complexity of encryption to depend also on the circuit-size, as long as this dependence is only sub-linear (and even exponentially on the circuit depth if we also assume puncturable PRFs in NC^1).

Is Private-Key FE Enough?: We do not know whether our construction (or any construction) of IO can be based on *general private-key* functional encryption. The key difficulty arises already in our first step of building "IO from much less IO", where we need to IO-obfuscate the (randomized) FE encryption circuit. In order to prove security, we crucially rely on the fact that in the public-key setting, encryptions of two inputs are indistinguishable *even given the encryption circuit* (which is simply the public key). This is not true any more with private-key FE since we cannot make the encryption circuit (or even an obfuscation of this circuit) public without ruining security. Indeed, we are able to show that instantiating our transformation with an arbitrary *private-key FE* scheme results in an insecure IO scheme.

Proposition I.1. *If there exists a succinct private-key functional encryption* FE*, then there also exists a succinct private-key functional encryption* FE^**, so that the transformation given by Theorem I.1 is insecure when instantiated with* FE^**.*

Complementing this negative result, we show that a notion of *puncturable* private-key FE suffices for our transformation. However, at this point do not know how to achieve this notion without relying on public-key schemes. (See Section V for more details). The real power of obfuscation manifests itself in transforming private-key schemes into public-key schemes [36], and for this reason, we believe that finding a (different) transformation from private-key FE to IO is a central open question.

C. Concurrent work

In a concurrent and independent work Ananth and Jain [37] also show how to construct indistinguishability obfuscation from sub-exponentially secure public-key functional encryption. The two works take a rather different approach to the problem. At high-level, Ananth and Jain show that any (sub-exponentially secure) public key functional encryption scheme can be converted into a multi-input functional encryption, a notion defined by Goldwasser et al. [38] that is known to imply indistinguishability obfuscation. The core step of their construction is a transformation from i-input FE to $(i+1)$-input FE, which is analogous to our recursive step of basing $i+1$-bit-input IO on i-bit-input IO. Our proof of security is perhaps more simple and concise, which we attribute to the fact that in each recursive step we fully exploit the expressive power of the IO guarantee, compared to the less expressive (multi-input) FE guarantee. In particular, we are able directly invoke previous techniques developed for IO, such as the concept of probabilistic IO [35].

In another concurrent work [39], Brakerski, Komargodski, and Segev, show how to convert any (single-input) private-key functional encryption scheme into an $O(1)$-input private-key scheme (or $O(\log\log\lambda)$-input assuming subexponential security), which is not known to be sufficient to go all the way to IO polynomially large inputs.

Organization: In Section II, we give the definitions of the cryptographic primitives used throughout the work, including functional encryption and indistinguishability obfuscation. In Section III, we describe the transformation from public-key functional encryption to indistinguishability obfuscation and analyze it. In Section IV, we give a general transformation from schemes that are succinct with respect to the number of derived keys to schemes that are succinct with respect to circuit-size, and obtain as a corollary a new IO construction based on [20]. Finally, in Section V, we present negative and positive results regarding the possibility of relying on private-key functional encryption schemes, rather than public-key ones.

II. Definitions

The cryptographic definitions in the paper follow the convention of modeling security against non-uniform adversaries. An efficient adversary $\mathcal{A}$ is modeled as a sequence of circuits $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, such that each circuit $\mathcal{A}_\lambda$ is of polynomial size $\lambda^{O(1)}$ with $\lambda^{O(1)}$ input and output bits. We often omit the subscript λ when it is clear from the context.

A. Functional Encryption

We recall the definition of public-key functional encryption (FE) with selective indistinguishability-based security [17, 18].

A public-key functional encryption scheme FE, for a function class $\mathcal{F}$ (represented by boolean circuits) and message space $\{0,1\}^*$, consists of four PPT algorithms ($\mathsf{FE.Setup}$, $\mathsf{FE.Gen}$, $\mathsf{FE.Enc}$, $\mathsf{FE.Dec}$) with the following syntax:

- $\mathsf{FE.Setup}(1^\lambda)$: Takes as input a security parameter λ in unary and outputs a (master) public key and a secret key $(\mathsf{PK}, \mathsf{MSK})$.
- $\mathsf{FE.Gen}(\mathsf{MSK}, f)$: Takes as input a secret key MSK, a function $f \in \mathcal{F}$ and outputs a functional key FSK_f.
- $\mathsf{FE.Enc}(\mathsf{PK}, m)$: Takes as input a public key PK, a message $m \in \{0,1\}^*$ and outputs an encryption of m. We shall sometimes address the randomness r used in encryption explicitly, which we denote by $\mathsf{FE.Enc}(\mathsf{PK}, m; r)$.
- $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: Takes as input a functional key FSK_f, a ciphertext CT and outputs $\widehat{m}$.

We next the define the required correctness and security properties.

Definition II.1 (Selectively-secure public-key FE)**.** *A tuple of PPT algorithms* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Gen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a selectively-secure public-key functional encryption scheme, for function class* $\mathcal{F}$*, and message space* $\{0,1\}^*$*, if it satisfies:*

1) **Correctness:** *for every* $\lambda, n \in \mathbb{N}$*, message* $m \in \{0,1\}^n$*, and function* $f \in \mathcal{F}$*, with domain* $\{0,1\}^n$*,*

$$\Pr\left[f(m) \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT}) \;\middle|\; \begin{array}{c}(\mathsf{PK}, \mathsf{MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda) \\ \mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f) \\ \mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}, m)\end{array}\right] = 1 \ .$$

2) **Selective-security:** *for any polysize adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\lambda)$ *such that for any* $\lambda \in \mathbb{N}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{FE}}_{\mathcal{A}} = \left|\mathsf{Pr}[\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \mathsf{Pr}[\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, 1) = 1]\right| \leq \mu(\lambda),$$

where for each $b \in \{0,1\}$ *and* $\lambda \in \mathbb{N}$ *the experiment* $\mathsf{Expt}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda, b)$*, modeled as a game between the challenger and the adversary* $\mathcal{A}$*, is defined as follows:*

a) *The adversary submits the challenge message-pair* $m_0, m_1 \in \{0,1\}^n$ *to the challenger.*
b) *The challenger executes* $\mathsf{FE.Setup}(1^\lambda)$ *to obtain* $(\mathsf{PK}, \mathsf{MSK})$*. It then executes* $\mathsf{FE.Enc}(\mathsf{PK}, m_b)$ *to obtain* CT*. The challenger sends* $(\mathsf{PK}, \mathsf{CT})$ *to the adversary.*
c) *The adversary submits function queries to the challenger. For any submitted function query* $f \in \mathcal{F}$ *defined over* $\{0,1\}^n$*, if* $f(m_0) = f(m_1)$*, the challenger generates and sends* $\mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f)$*. In any other case, the challenger aborts.*
d) *The output of the experiment is the output of* $\mathcal{A}$*.*

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

Single-key FE with succinct encryption.: In this work, we consider a special case of a single functional key for a function that is known in setup time, where we require that the encryption is succinct in some sense. This will be sufficient in our application.

Such a scheme FE, for a function class $\mathcal{F}$ (represented by boolean circuits) and message space $\{0,1\}^*$, consists of four PPT algorithms $(\mathsf{FE.Setup}, \mathsf{FE.Gen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ with the following syntax:

- $\mathsf{FE.Setup}(1^\lambda, f)$: takes as input a security parameter λ in unary and function $f \in \mathcal{F}$ and outputs a public key PK and a functional key FSK_f.
- $\mathsf{FE.Enc}(\mathsf{PK}, m)$: takes as input a public key PK, a message $m \in \{0,1\}^*$ and outputs an encryption of m. We shall sometimes address the randomness r used in encryption explicitly, which we denote by $\mathsf{FE.Enc}(\mathsf{PK}, m; r)$.
- $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: takes as input a functional key FSK_f, a ciphertext CT and outputs $\widehat{m}$.

We next the define the required correctness, security, and efficiency properties. While the first two are a special case of Definition II.1, they can be restated more simply.

Definition II.2 (Single-key, selectively-secure, public-key FE with succinct encryption)**.** *A tuple of PPT algorithms* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a single-key, selectively-secure, public-key functional encryption scheme with succinct encryption, for function class* $\mathcal{F}$*, and message space* $\{0,1\}^*$*, if it satisfies:*

1) **Correctness:** *for every* $\lambda, n \in \mathbb{N}$*, message* $m \in \{0,1\}^n$*, and function* $f \in \mathcal{F}$*, with domain* $\{0,1\}^n$*,*

$$\Pr\left[f(m) \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT}) \;\middle|\; \begin{array}{c}(\mathsf{PK}, \mathsf{FSK}_f) \leftarrow \mathsf{FE.Setup}(1^\lambda, f) \\ \mathsf{CT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}, m)\end{array}\right] = 1 \ .$$

2) **Selective security:** *for any polysize adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\lambda)$ *such that for any* $\lambda \in \mathbb{N}$*, any* $m_0, m_1 \in \{0,1\}^n$*, and function* $f \in \mathcal{F}$ *such that* $f(m_0) = f(m_1)$*,*

$$|\Pr\left[\mathcal{A}(\mathsf{PK}, \mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_0)) = 1\right] - \Pr\left[\mathcal{A}(\mathsf{PK}, \mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_1)) = 1\right]| \leq \mu(\lambda) \ ,$$

where $(\mathsf{PK}, \mathsf{FSK}_f) \leftarrow \mathsf{FE.Setup}(1^\lambda, f)$.
We further say that FE *is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for all polysize adversaries the above indistinguishability gap $\mu(\lambda)$ is smaller than $\delta(\lambda)^{\Omega(1)}$.*

3) **Succinct encryption:**
   - *Encryption is* **fully succinct** *if the size of the encryption circuit is bounded by $\mathrm{poly}(n, \lambda)$ for a fixed polynomial* poly.
   - *Encryption is* **weakly succinct** *if the size of the encryption circuit is bounded by $s^{1-\varepsilon} \cdot \mathrm{poly}(n, \lambda, 2^d)$ where $s = \max_{f \in \mathcal{F}_n} |f|$, $d = \max_{f \in \mathcal{F}_n} \mathsf{dep}(f)$, $\mathcal{F}_n \subseteq \mathcal{F}$ is the subset of functions defined on $\{0,1\}^n$,* poly *is a fixed polynomial, and $\varepsilon < 1$ is a constant.*

B. Indistinguishability Obfuscation

We define indistinguishability obfuscation (IO) with respect to a give class of circuits. The definition is formulated as in [1].

Definition II.3 (Indistinguishability obfuscation)**.** *A PPT algorithm $i\mathcal{O}$ is said to be an* indistinguishability obfuscator *for a class of circuits $\mathcal{C}$, if it satisfies:*

1) **Functionality:** *for any $C \in \mathcal{C}$ and security parameter λ,*
$$\Pr_{i\mathcal{O}}\left[\forall x : i\mathcal{O}(C, 1^\lambda)(x) = C(x)\right] = 1 \ .$$
2) **Indistinguishability:** *for any polysize distinguisher $\mathcal{D}$ there exists a negligible function $\mu(\cdot)$, such that for any two circuits $C_0, C_1 \in \mathcal{C}$ that compute the same function and are of the same size:*
$$\left|\Pr[\mathcal{D}(i\mathcal{O}(C_0, 1^\lambda)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(C_1, 1^\lambda)) = 1]\right| \leq \mu(\lambda) \ ,$$
where the probability is over the coins of $\mathcal{D}$ and $i\mathcal{O}$.
We further say that $i\mathcal{O}$ is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for all polysize distinguishers the above indistinguishability gap $\mu(\lambda)$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

C. Puncturable Pseudorandom Functions

We consider a simple case of the puncturable pseudo-random functions (PRFs) where any PRF may be punctured at a single point. The definition is formulated as in [6], and is satisfied by the GGM [40] PRF [41, 42, 43].

Definition II.4 (Puncturable PRFs)**.** *Let n, k be polynomially bounded length functions. An efficiently computable family of functions*
$$\mathcal{PRF} = \left\{\mathsf{PRF}_\mathsf{K} : \{0,1\}^* \to \{0,1\}^\lambda \;\middle|\; \mathsf{K} \in \{0,1\}^{k(\lambda)}, \lambda \in \mathbb{N}\right\} \ ,$$
associated with an efficient (probabilistic) key sampler $\mathsf{Gen}_{\mathcal{PRF}}$, is a puncturable PRF *if there exists a poly-time puncturing algorithm* Punc *that takes as input a key* K*, and a point x^*, and outputs a punctured key $\mathsf{K}\{x^*\}$, so that the following conditions are satisfied:*

1) **Functionality is preserved under puncturing:** *For every $x^* \in \{0,1\}^*$,*
$$\Pr_{\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda)}\left[\forall x \neq x^* : \mathsf{PRF}_\mathsf{K}(x) = \mathsf{PRF}_{\mathsf{K}\{x^*\}}(x) \;\middle|\; \mathsf{K}\{x^*\} = \mathsf{Punc}(\mathsf{K}, x^*)\right] = 1 \ .$$
2) **Indistinguishability at punctured points:** *for any polysize distinguisher $\mathcal{D}$ there exists a negligible function $\mu(\cdot)$, such that for all $\lambda \in \mathbb{N}$, and any $x^* \in \{0,1\}^*$,*
$$|\Pr[\mathcal{D}(x^*, \mathsf{K}\{x^*\}, \mathsf{PRF}_\mathsf{K}(x^*)) = 1] - \Pr[\mathcal{D}(x^*, \mathsf{K}\{x^*\}, u) = 1]| \leq \mu(\lambda) \ ,$$
where $\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda), \mathsf{K}\{x^*\} = \mathsf{Punc}(\mathsf{K}, x^*)$, and $u \leftarrow \{0,1\}^\lambda$.
We further say that $\mathcal{PRF}$ is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for all polysize distinguishers the above indistinguishability gap $\mu(\lambda)$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

D. Symmetric Encryption

A symmetric encryption scheme Sym consists of a tuple of two PPT algorithms $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$. The encryption algorithm takes as input a symmetric key $\mathsf{SK} \in \{0,1\}^\lambda$, where λ is the security parameter and a message $m \in \{0,1\}^*$ of polynomial size in the security parameter, and outputs is a ciphertext CT. The decryption algorithm takes as input $(\mathsf{SK}, \mathsf{CT})$, and outputs the decrypted message m. For this work we only require one-time security.

Definition II.5 (One-Time Symmetric Encryption)**.** *A pair of PPT algorithms* $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ *is a one-time symmetric encryption scheme for message space* $\{0,1\}^*$ *if it satisfies:*

1) **Correctness:** *For every security parameter* λ *and message* $m \in \{0,1\}^*$*,*

$$\Pr\left[\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}) = m \;\middle|\; \begin{array}{c} \mathsf{SK} \leftarrow \{0,1\}^\lambda \\ \mathsf{CT} \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, m) \end{array}\right] = 1 \ .$$

2) **Indistinguishability:** *for any polysize distinguisher* $\mathcal{D}$ *there exists a negligible function* $\mu(\cdot)$*, such that for all* $\lambda \in \mathbb{N}$*, and any equal size messages* m_0, m_1*,*

$$|\Pr[\mathcal{D}(\mathsf{Sym.Enc}(\mathsf{SK}, m_0)) = 1] - \Pr[\mathcal{D}(\mathsf{Sym.Enc}(\mathsf{SK}, m_1)) = 1]| \leq \mu(\lambda) \ ,$$

where $\mathsf{SK} \leftarrow \{0,1\}^\lambda$*.*
We further say that Sym *is* δ*-secure, for some concrete negligible function* $\delta(\cdot)$*, if for all polysize distinguishers the above indistinguishability gap* $\mu(\lambda)$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

E. Randomized Encodings

We rely on the notion of randomized encodings from [44, 45]. Let $c \geq 1$ be an integer constant. A (c-local, decomposable) randomized encoding for a function $f : \{0,1\}^n \to \{0,1\}^m$ is a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^\mu$ with the following properties. Let $s_{\hat{f}}$ (resp. s_f) denote the size of the circuit computing $\hat{f}$ (resp. f).

- $\hat{f}(x;r) = (\hat{f}_1(x;r), \hat{f}_2(x;r), \ldots, \hat{f}_\mu(x;r))$ where each $\hat{f}_i$ depends on at most *a single bit* of x and c bits of r. We will write
$$\hat{f}(x;r) = (\hat{f}_1(x; r_{S_1}), \hat{f}_2(x; r_{S_2}), \ldots, \hat{f}_\mu(x; r_{S_\mu}))$$
where S_i denotes the subset of bits of r that $\hat{f}_i$ depends on.
- μ and ρ are of size $s_f \cdot \mathrm{poly}(n, \lambda)$.
- There is a polynomial time decoder algorithm that, given $\hat{f}(x;r)$, outputs $f(x)$.
- There is a PPT simulator $\mathsf{RE.Sim}$ that takes as input $(1^\lambda, f(x))$ and outputs $\mathsf{SimOut}_{f(x)}$ such that no polysize adversary can distinguish between the distributions $\{\hat{f}(x;r)\}_{x \in \{0,1\}^*}$ and the distribution $\{\mathsf{SimOut}_{f(x)}\}_{x \in \{0,1\}^*}$.

Such randomized encodings can be constructed from one-way functions [46]. Furthermore, each $\hat{f}_i$ can be computed by a shallow circuit whose depth is determined by the depth in which a linear stretch PRG can be computed (over strings of length λ) [45].

III. The Transformation

In this section, we describe the transformation and analyze it.

Ingredients: We rely on the following primitives:

- A $2^{-\tilde{\lambda}^\varepsilon}$-secure single-key, selectively-secure, public-key functional encryption scheme FE for P/poly, for a single key with (fully or weakly) succinct encryption.
- A $2^{-\tilde{\lambda}^\varepsilon}$-secure one-time symmetric encryption scheme Sym,
- A $2^{-\tilde{\lambda}^\varepsilon}$-secure puncturable pseudo-random function family $\mathcal{PRF}$.

where $\tilde{\lambda}$ is the security parameter and $\varepsilon < 1$.

The obfuscator $i\mathcal{O}$: Given a circuit $C : \{0,1\}^n \to \{0,1\}$ and security parameter λ, the obfuscator $i\mathcal{O}(C, 1^\lambda)$, computes a new security parameter $\tilde{\lambda} = \omega((n^2 + \log\lambda)^{1/\varepsilon})$, and invokes a recursive obfuscation procedure $r\mathcal{O}.\mathsf{Obf}(n, C, 1^{\tilde{\lambda}})$. In general, the recursive obfuscation procedure $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ extends obfuscation for circuits with $i-1$ bits to obfuscation for circuits with i bits. To this end, it generates an obfuscation of an encryption circuit E_i that takes a prefix $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ and generates two encyptions of each possible continuation $\mathbf{x}0$ or $\mathbf{x}1$. The procedure is given in Figure 1. A corresponding recursive evaluation procedure $r\mathcal{O}.\mathsf{Eval}$ is described right after.

$$r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$$

Input: An input length $i \in \mathbb{N}$, a circuit $C_i : \{0,1\}^i \to \{0,1\}^*$, and security parameter $\tilde{\lambda}$.

1) If $i = 1$, output $(C_i(0), C_i(1))$.
2) Otherwise, generate:
   - Symmetric encryption keys $(\mathsf{SK}_i^0, \mathsf{SK}_i^1) \leftarrow \{0,1\}^{\tilde{\lambda}} \times \{0,1\}^{\tilde{\lambda}}$.
   - Symmetric encryptions $(\mathsf{CT}_i^0, \mathsf{CT}_i^1) \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}_i^0, C_i) \times \mathsf{Sym.Enc}(\mathsf{SK}_i^1, C_i)$.
   - A circuit f_i defined for $(\mathbf{x}_i, \mathsf{SK}, \beta) \in \{0,1\}^i \times \{0,1\}^{\tilde{\lambda}} \times \{0,1\}$ by
   $$f_i(\mathbf{x}_i, \mathsf{SK}, \beta) = U(\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i^\beta), \mathbf{x}_i) \ ,$$
   where $U(\cdot,\cdot)$ is the universal circuit.
   - Public key and functional key $(\mathsf{PK}_i, \mathsf{FSK}_i) \leftarrow \mathsf{FE.Setup}(1^{\tilde{\lambda}}, f_i)$.
   - Seed $\mathsf{K}_i \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^{\tilde{\lambda}})$ for a puncturable pseudo random function.
   - A circuit E_{i-1}^0 defined for any $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ by
   $$\mathsf{E}_{i-1}^0(\mathbf{x}_{i-1}) = \left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}_{i-1}, x_i), \mathsf{SK}_i^0, 0); \mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}_{i-1}, x_i))\right\}_{x_i \in \{0,1\}} \ ,$$
   padded to some size $\ell(\tilde{\lambda})$ for some polynomial $\ell(\cdot)$ determined in the analysis.
   - An obfuscation
   $$\widetilde{\mathsf{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathsf{E}_{i-1}^0, 1^{\tilde{\lambda}}) \ .$$
3) Output $\widetilde{\mathsf{E}}_i := (\widetilde{\mathsf{E}}_{i-1}, \mathsf{FSK}_i)$.

Figure 1. The recursive obfuscation procedure.

Theorem III.1. *$i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

Functionality: The evaluation of the obfuscated $i\mathcal{O}(C, 1^\lambda) = \widetilde{\mathsf{E}}_n$ on input $\mathbf{x} \in \{0,1\}^n$ is done by invoking the recursive evaluation procedure $r\mathcal{O}.\mathsf{Eval}(n, \widetilde{\mathsf{E}}_n, \mathbf{x})$. This procedure gradually constructs an encryption FCT_n of $\mathbf{x}$. At step i, given encryptions $(\mathsf{FCT}_i^0, \mathsf{FCT}_i^1)$ of $(\mathbf{x}_{i-1}, 0)$ and $(\mathbf{x}_{i-1}, 1)$ it chooses $\mathsf{FCT}_i^{x_i}$ and decrypts with FSK_i to compute $(\mathsf{FCT}_{i+1}^0, \mathsf{FCT}_{i+1}^1)$ or $C(\mathbf{x}_n)$ in the very last step. The procedure is given in Figure 2.

$$r\mathcal{O}.\mathsf{Eval}(i, \widetilde{\mathsf{E}}_i, \mathbf{x}_i)$$

Input: An input length $i \in \mathbb{N}$, an obfuscation $\widetilde{\mathsf{E}}_i = (\widetilde{\mathsf{E}}_{i-1}, \mathsf{FSK}_i)$, and prefix $\mathbf{x}_i \in \{0,1\}^i$.

1) If $i = 1$, parse $\widetilde{\mathsf{E}}_1 = (\mathsf{FCT}_1^0, \mathsf{FCT}_1^1)$, and output $\mathsf{FCT}_1^{\mathbf{x}_1}$.
2) Otherwise, compute $(\mathsf{FCT}_i^0, \mathsf{FCT}_i^1) = r\mathcal{O}.\mathsf{Eval}(\widetilde{\mathsf{E}}_{i-1}, \mathbf{x}_{i-1})$, where $\mathbf{x}_{i-1}$ are the first $i-1$ bits of $\mathbf{x}_i$.
3) Output $\mathsf{FE.Dec}(\mathsf{FSK}_i, \mathsf{FCT}_i^{x_i})$.

Figure 2. The recursive evaluation procedure.

Functionality follows readily by the correctness of the functional encryption scheme FE and the symmetric encryption scheme Sym. Indeed, each $\mathsf{FCT}_i^{x_i}$ is an encryption of $\mathbf{x}_i$, the ith prefix of $\mathbf{x}$; in particular, $\mathsf{FCT}_n^{x_n}$ encrypts $\mathbf{x} = \mathbf{x}_n$. Thus, the last decryption operation results in $C(\mathbf{x})$.

Efficiency: For simplicity, let us first assume that the encryption is full succinct. In the full version of this work, we extend the analysis to the case of sub-linear dependence on the circuit size and even exponential dependence on circuit-depth.

Note that the running time of each invocation of $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ is bounded by some polynomial $\mathrm{poly}(|C_i|, |\mathsf{E}_i^0|, \lambda, n)$ plus the running time of the recursive call to $r\mathcal{O}.\mathsf{Obf}(i-1, \cdots)$ (and poly is fixed independently of i). Second, note that the obfuscated circuit C_i is C when $i = n$, and E_i^0 for any $i \in [n-1]$. It is left to see that the maximal size of any circuit E_i^0, $\max_i |\mathsf{E}_i^0|$ is bounded by some fixed polynomial $\mathrm{poly}(n, \lambda)$. Indeed, each such circuit computes two encryptions of $i + \lambda + 1$ bits and a pseudo-random function to derive randomness for this operation. Here we invoke the assumption that the size of the encryption circuit only depends on the size of the plaintext and the security parameter (and not say on the number of keys in the system, or output length of functional keys). Thus, overall the time to obfuscate (and size of the resuting obfuscation) is bounded by a fixed polynomial $\mathrm{poly}(|C|, \lambda)$ as required.

A. Security Analysis

Let $s(\cdot), n(\cdot)$ be any two polynomially-bounded functions and $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ be any poly-size distinguisher that works on obfuscations $i\mathcal{O}(C, 1^\lambda)$ for any circuit circuit C of size $s(\lambda)$, defined on $\{0,1\}^{n(\lambda)}$.

Our goal is to show that

$$\delta_{i\mathcal{O}}(\lambda) := \max_{C_0, C_1} \left| \Pr\left[\mathcal{D}(i\mathcal{O}(C_0, 1^\lambda)) = 1\right] - \Pr\left[\mathcal{D}(i\mathcal{O}(C_1, 1^\lambda)) = 1\right] \right| =$$
$$\max_{C_0, C_1} \left| \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(n, C_0, 1^{\tilde{\lambda}})) = 1\right] - \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(n, C_1, 1^{\tilde{\lambda}})) = 1\right] \right| \leq 2^{-\omega(\log \lambda)} \ ,$$

where C_0 and C_1 are any two circuits defined on $\{0,1\}^{n(\lambda)}$ of the same functionality and size $s(\lambda)$.

For every every $\lambda \in \mathbb{N}$, define $\delta_{n(\lambda)} := \delta_{i\mathcal{O}}(\lambda)$ and for $1 \leq i < n(\lambda)$, define

$$\delta_i := \max_{C_0, C_1, z} \left| \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i, C_0, 1^{\tilde{\lambda}}), z) = 1\right] - \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i, C_1, 1^{\tilde{\lambda}}), z) = 1\right] \right| \ ,$$

where C_0 and C_1 are any two circuits defined on $\{0,1\}^i$ of the same functionality and size $\ell(\tilde{\lambda})$.

Proposition III.1. *$\delta_1 = 0$ and for any $i \in \{2, \ldots, n(\lambda)\}$, $\delta_i \leq 2^{i-1} \cdot O(\delta_{i-1} + 2^{-\Omega(\tilde{\lambda}^\varepsilon)})$.*

Before proving the proposition, note that it concludes the security analysis since it implies

$$\begin{aligned}
\delta_{i\mathcal{O}}(\lambda) = \ & \delta_n \leq \\
& 2^{n-1} \cdot O(\delta_{n-1} + 2^{-\Omega(\tilde{\lambda}^\varepsilon)}) \leq \\
& 2^{n-1} \cdot O(2^{-\Omega(\tilde{\lambda}^\varepsilon)}) + 2^{n-1} \cdot 2^{n-2} \cdot O(\delta_{n-2} + 2^{-\Omega(\tilde{\lambda}^\varepsilon)}) \leq \\
& \qquad \vdots \\
& \left(\sum_{i=1}^{n} \prod_{j=1}^{i} 2^{n-j} \right) \cdot O(2^{-\Omega(\tilde{\lambda}^\varepsilon)}) \leq \\
& O(n \cdot 2^{n^2/2}) \cdot O(2^{-\Omega(\tilde{\lambda}^\varepsilon)}) \leq \\
& O(n \cdot 2^{n^2/2}) \cdot O(2^{-\omega(n^2 + \log \lambda)}) = \\
& 2^{-\omega(\log \lambda)} \ .
\end{aligned}$$

Proof of Proposition III.1: First, to see that $\delta_1 = 0$, note that for any C defined on $\{0,1\}$,

$$r\mathcal{O}.\mathsf{Obf}(1, C, 1^{\tilde{\lambda}}) = (C(0), C(1))$$

by definition, and thus for any two C_0, C_1 with the same functionality

$$r\mathcal{O}.\mathsf{Obf}(1, C_0, 1^{\tilde{\lambda}}) \equiv r\mathcal{O}.\mathsf{Obf}(1, C_1, 1^{\tilde{\lambda}}) \ .$$

We now prove the main part of the proposition. Fix $i \in \{2, \ldots, n(\lambda)\}$, and let C_0, C_1 be any two circuits defined on $\{0,1\}^i$ of eqaul size $\ell(\tilde{\lambda})$ and fix any auxiliary z. Our goal is to show that

$$\left|\Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i, C_0, 1^{\tilde{\lambda}}), z) = 1\right] - \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i, C_1, 1^{\tilde{\lambda}}), z) = 1\right]\right| \leq 2^{i-1} \cdot O(\delta_{i-1} + 2^{-\Omega(\tilde{\lambda}^{\varepsilon})}) \ .$$

Recall that

$$r\mathcal{O}.\mathsf{Obf}(i, C_b, 1^{\tilde{\lambda}}) = \left(\widetilde{\mathsf{E}}_{i-1}, \mathsf{FSK}_i\right) \ ,$$

where $\widetilde{\mathsf{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathsf{E}^0_{i-1}, 1^{\tilde{\lambda}})$ and E^0_{i-1} is a circuit that has $(\mathsf{PK}_i, \mathsf{SK}^0_i, \mathsf{K}_i)$ hardwired, and which, on input $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$, computes two encryptions

$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}_{i-1}, x_i), \mathsf{SK}^0_i, 0); \mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}_{i-1}, x_i))\right\}_{x_i \in \{0,1\}} \ ,$$

and FSK_i is a functional decryption that has two hardwired syemmetric encryptions CT^0_i and CT^1_i both of the circuit C_b; FSK_i corresponds to the function that decrypts according to the key specified in the plaintext.

For every three bits $\beta, \gamma_0, \gamma_1 \in \{0,1\}$, we consider a hybrid experiment $\mathcal{H}^{\gamma_0,\gamma_1}_{\beta}$ where

- $\widetilde{\mathsf{E}}_{i-1}$ is an obfuscation of E^{β}_{i-1} that encrypts $(\mathsf{SK}^{\beta}_i, \beta)$, rather than always encrypting $(\mathsf{SK}^0_i, 0)$. (The circuit is independent of $\mathsf{SK}^{1-\beta}_i$.)
- CT^0_i encrypts C_{γ_0} and CT^1_i encrypts C_{γ_1}. (It may be that $\gamma_0 \neq \gamma_1$.)

Note that $\mathcal{H}^{0,0}_0$ and $\mathcal{H}^{1,1}_0$ exactly correspond to obfuscating either C_0 or C_1. We show that

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}(\mathcal{H}^{0,0}_0) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{H}^{0,1}_0) = 1\right]\right| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ , \\
&\left|\Pr\left[\mathcal{D}(\mathcal{H}^{0,1}_0) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{H}^{0,1}_1) = 1\right]\right| \leq 2^{i-1} \cdot O(\delta_{i-1} + 2^{-\Omega(\tilde{\lambda}^{\varepsilon})}) \ , \\
&\left|\Pr\left[\mathcal{D}(\mathcal{H}^{0,1}_1) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{H}^{1,1}_1) = 1\right]\right| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ , \\
&\left|\Pr\left[\mathcal{D}(\mathcal{H}^{1,1}_1) = 1\right] - \Pr\left[\mathcal{D}(\mathcal{H}^{1,1}_0) = 1\right]\right| \leq 2^{i-1} \cdot O(\delta_{i-1} + 2^{-\Omega(\tilde{\lambda}^{\varepsilon})}) \ .
\end{aligned}$$

In the first and third inequalities, we simply change the symmetrically encrypted plaintext in some CT^b_i where only the key SK^{1-b}_i is present. Thus the inequalities follow from the (one-time) symmetric encryption guarantee.

We now show equations two and four; concretely, we focus on the second equation, and the forth is proven using a similar argument. Recall again that the difference between $\mathcal{H}^{0,1}_0$ and $\mathcal{H}^{0,1}_1$ is in the obfuscated $\widetilde{\mathsf{E}}_{i-1}$. In the first, the circuit E^0_{i-1}, which always puts SK^0_i in the plaintext, is obfuscated, and in the second E^1_{i-1}, which always puts SK^1_i in the plaintext, is obfuscated. The key to the indistinguishability behind the hybrids is that the output of the two circuits on any point $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ is indistinguishable even given the two circuits themselves as long as the randomness used to generate the output is not revealed. Indeed, because the circuits encrypted in $\mathsf{CT}^0_i, \mathsf{CT}^1_0$ compute the same function, FSK_i does not allow distinguishing between the two cases and we can invoke the FE guarantee. Canetti, Lin, Tessaro, and Vaikuntanathan [35] show that sub-exponential IO in conjunction with sub-exponential puncturable PRFs are sufficient in this setting, which they formalize by *probabilistic IO* notion. For the sake of completeness, we next give the full argument.

We consider a sequence of $2^{i-1}+1$ hybrids $\{\mathcal{H}_{\mathbf{x}}\}_{\mathbf{x} \in \{0,\ldots,2^{i-1}\}}$, where we naturally identify integers in $[2^{i-1}]$ with strings in $\{0,1\}^{i-1}$. In $\mathcal{H}_{\mathbf{x}}$, both CT^0_i and CT^1_i encrypt the same circuit $\mathsf{E}_{\mathbf{x}}(\mathbf{x}')$ that computes $\mathsf{E}^0_{i-1}(\mathbf{x}')$ for all $\mathbf{x}' > \mathbf{x}$ and $\mathsf{E}^1_{i-1}(\mathbf{x}')$ for all $\mathbf{x}' \leq \mathbf{x}$; the circuit $\mathsf{E}_{\mathbf{x}}$ is padded to size $\ell(\tilde{\lambda})$.

We first note that E_0 computes the same function as E^0_{i-1} and that $\mathsf{E}_{2^{i-1}}$ computes the same function as E^1_{i-1}, and thus

$$\left|\Pr\left[\mathcal{D}(\mathcal{H}_0^{0,1})=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}_0)=1\right]\right|\leq\delta_{i-1}\ ,$$
$$\left|\Pr\left[\mathcal{D}(\mathcal{H}_{2^{i-1}})=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}_0^{0,1})=1\right]\right|\leq\delta_{i-1}\ .$$

We now show that for any $\mathbf{x}\in[2^{i-1}]$,

$$|\Pr\left[\mathcal{D}(\mathcal{H}_{\mathbf{x}-1})=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}_{\mathbf{x}})=1\right]|\leq O(\delta_{i-1}+2^{-\Omega(\tilde{\lambda}^\varepsilon)})\ .$$

Note that the difference between $\mathcal{H}_{\mathbf{x}-1}$ and $\mathcal{H}_{\mathbf{x}}$ is in the circuits encrypted in $\mathsf{CT}^0_i,\mathsf{CT}^1_i$: $\mathsf{E}_{\mathbf{x}-1}$ in $\mathcal{H}_{\mathbf{x}-1}$ and $\mathsf{E}_{\mathbf{x}}$ in $\mathcal{H}_{\mathbf{x}}$. Further note that these two circuits only differ on $\mathbf{x}$: the first returns $\mathsf{E}^0_{i-1}(\mathbf{x})$ whereas the second returns $\mathsf{E}^1_{i-1}(\mathbf{x})$. We consider the following sub-hybrids:

- $\mathcal{G}_1$: instead of $\mathsf{E}_{\mathbf{x}-1}$, $\mathsf{CT}^0_i,\mathsf{CT}^1_i$ both encrypt $\mathsf{E}'_{\mathbf{x}-1}$ that has

  $$\mathsf{E}_{\mathbf{x}-1}(\mathbf{x})=\mathsf{E}^0_{i-1}(\mathbf{x})=\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^0_i,0);\mathsf{PRF}_{\mathsf{K}}i(\mathbf{x},x_i))\ \middle|\ x_i\in\{0,1\}\right\}$$

  hardwired as well as a punctured key $\mathsf{K}_i\{(\mathbf{x},x_i)\}$ used to generate all other encryptions. The circuit is padded to size $\ell(\tilde{\lambda})$.
  Since $\mathsf{E}_{\mathbf{x}-1}$ and $\mathsf{E}'_{\mathbf{x}-1}$ compute the same function:

  $$|\Pr\left[\mathcal{D}(\mathcal{H}_{\mathbf{x}-1})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}_1)=1\right]|\leq\delta_{i-1}\ .$$

- $\mathcal{G}_2$: Here we replace the hardwired

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^0_i,0);\mathsf{PRF}_{\mathsf{K}}i(\mathbf{x},x_i))\ \middle|\ x_i\in\{0,1\}\right\}$$

  so that instead of using the pseudo-randomness $\mathsf{PRF}_{\mathsf{K}}i(\mathbf{x},x_i)$, true randomness r is used

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^0_i,0);r)\ \middle|\ x_i\in\{0,1\}\right\}\ .$$

  By pseudo-randomness at punctured points

  $$|\Pr\left[\mathcal{D}(\mathcal{G}_1)=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}_2)=1\right]|\leq 2^{-\Omega(\tilde{\lambda}^\varepsilon)}\ .$$

- $\mathcal{G}_3$: Here we replace the hardwired

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^0_i,0);r)\ \middle|\ x_i\in\{0,1\}\right\}$$

  to encrypt $(\mathsf{SK}^1_i,1)$ instead of $(\mathsf{SK}^0_i,0)$:

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^1_i,1);r)\ \middle|\ x_i\in\{0,1\}\right\}\ .$$

  Since, CT^0_i and CT^1_i encrypt circuits C_0 and C_1, respectively, with the exact same functionality, we can apply the FE guarantee to deduce

  $$|\Pr\left[\mathcal{D}(\mathcal{G}_2)=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}_3)=1\right]|\leq 2^{-\Omega(\tilde{\lambda}^\varepsilon)}\ .$$

- $\mathcal{G}_{2'}$: reverses $\mathcal{G}_2$, we replace the hardwired

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^1_i,1);r)\ \middle|\ x_i\in\{0,1\}\right\}$$

  with

  $$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i,((\mathbf{x},x_i),\mathsf{SK}^1_i,1);\mathsf{PRF}_{\mathsf{K}}i(\mathbf{x},x_i))\ \middle|\ x_i\in\{0,1\}\right\}\ .$$

  By pseudo-randomness at punctured points

  $$|\Pr\left[\mathcal{D}(\mathcal{G}_3)=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}_{2'})=1\right]|\leq 2^{-\Omega(\tilde{\lambda}^\varepsilon)}\ .$$

- Denote by $\mathsf{E}'_{\mathbf{x}}$ the circuit $\mathsf{E}'_{\mathbf{x}-1}$ after the above changes to the hardwired encryption. Note that $\mathsf{E}'_{\mathbf{x}}$ and $\mathsf{E}_{2^{i-1}}$ compute the same function, we deduce

$$|\Pr[\mathcal{D}(\mathcal{G}_{2'}) = 1] - \Pr[\mathcal{D}(\mathcal{H}_{\mathbf{x}}) = 1]| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ .$$

Overall,

$$|\Pr[\mathcal{D}(\mathcal{H}_{\mathbf{x}-1}) = 1] - \Pr[\mathcal{D}(\mathcal{H}_{\mathbf{x}}) = 1]| \leq O(\delta_{i-1} + 2^{-\Omega(\tilde{\lambda}^{\varepsilon})}) \ ,$$

as required, which completes the proof of the proposition.

Remark III.2. *Formally, we have defined PRF puncturing at a single point, where as in the above argument we need to puncture in $(\mathbf{x}, x_i)$ for both $x_i \in \{0,1\}$. One can naturally define puncturing at two points, or simply go through the above hybrids separately for each $x_i \in \{0,1\}$.*

The padding parameter: $\ell(\tilde{\lambda})$ is chosen to account for the maximal-size circuit considered in any of the above hybrids. ■

Extended efficiency analysis: So far, we have analyzed the efficiency of our obfuscator, assuming that the running time of the functional encryption algorithm is fully succinct, namely, bounded by some fixed polynomial $\mathrm{poly}(n, \tilde{\lambda})$ in the total input size n and the security parameter $\tilde{\lambda}$, independently of circuit-size $|C|$, circuit-depth d, or output-size m of functions (here $\tilde{\lambda} = \omega((n^2 + \log \lambda)^{1/\varepsilon})$, for the security parameter λ). In the full version of this work, we show that the efficiency of the transformation is maintained, even if the encryption algorithm depends on the circuit size $|C|$ (and in-particular the output size $m \leq |C|$), but only sublinearly, namely it is bounded by $|C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda})$, for any constant ε. Furthermore, we show that a slight variant of the transformation maintains efficiency even when encryption depends exponentially on the circuit depth d.

B. IO with Linear Overhead

In this section, we observe that our technique, combined with known results from the literature, implies that any IO scheme can be turned into an IO scheme where the size of an obfuscation of a circuit C of depth d is of size $2|C| + \mathrm{poly}(d, n, \lambda)$, assuming LWE.

The basic observation is that a single iteration of our transformation, i.e. running $r\mathcal{O}.\mathsf{Obf}(n, C, 1^{\tilde{\lambda}})$, results in an obfuscation $\widetilde{\mathsf{E}}_{n-1}$ of a circuit E_{n-1}, generating FE encryptions of inputs, plus a functional key FSK_n for the function f_n that performs decryption and evaluation of the circuit C. In particular:

- the size of the circuit E_{n-1} is dominated by the complexity of FE encryption,
- the function f_n can be represented by $2|C|$ bits, consisting of two one-time encryptions of $|C|$. (For example, using a PRG that expends λ bits to $|C|$ bits as a one-time pad.)

We can then rely on the following result by Boneh et al.

Proposition III.2 (FE with succinct keys [22])**.** *Assuming subexponential LWE, there exists a single-key, public-key, functional encryption scheme, where the size of the encryption circuit and of a functional key are both $m \cdot \mathrm{poly}(n, \lambda, d)$, for classes of circuits with inputs and outputs of size n and m, and maximal depth d. (Functional decryption, requires also the (public) description of the function.)*

Obfuscating E_{n-1} with any IO scheme, and plugging-in the above FE scheme, we deduce:

Corollary III.3. *Assuming subexponential LWE and IO, there exists IO such that, given any circuit $C : \{0,1\}^n \to \{0,1\}$ of size s and depth d, a corresponding obfuscation is of size $2s + \mathrm{poly}(n, d, \lambda)$.*

IV. IO FROM THE GGHZ FUNCTIONAL ENCRYPTION

In this section, we show how to transform any *collusion-succinct* functional encryption scheme into a (circuit) succinct functional encryption scheme (according to Definition II.2), which in particular is suitable for our IO transformation. In a *collusion-succinct* FE scheme, the ciphertexts could grow polynomially with the input length, the maximum circuit-size supported by the scheme, and the security parameter, but they grow *sub-linearly* with the number of collusions (derived functional keys) that the scheme can handle. Applying this transformation to the functional encryption scheme from the work of Garg, Gentry, Halevi and Zhandry [20], we obtain an IO construction based on a subexponential variant of the assumptions in [20] on multi-linear graded encodings.

We now turn to describe the transformation that is similar to several randomized-encoding-based bootstrapping schemes from the literature [26, 47, 48]. For simplicity, we describe everything in terms of polynomial security. The transformation can be naturally scaled for the case of sub-exponential security.

Proposition IV.1. *For every $\epsilon = \epsilon(\lambda, N) < 1$:*

- *If there is a (selectively secure) FE scheme for circuits of size at most $s = s(\lambda)$ with $n = n(\lambda)$ inputs, secure against the release of $N = N(\lambda)$ functional keys, with ciphertexts of size $N^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda, s)$,*
- *Then, there is a (selectively secure) FE scheme for circuits of size at most $s = s(\lambda)$ secure against the release of $N = N(\lambda)$ functional keys with ciphertexts of size $(s \cdot N)^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda)$.*

In particular, for constant ϵ, we get a transformation from any weakly collusion-succinct to a weakly circuit succinct scheme. For $\epsilon = 1 - \log_N \mathrm{poly}(n, \lambda)$, we get a transformation from a fully collusion-succinct to a fully circuit succinct scheme.

Proof: Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Gen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be a collusion-succinct functional encryption scheme. Let $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ be a one time symmetric-key encryption scheme (Definition II.5). We construct a scheme $\mathsf{sFE} = (\mathsf{sFE.Setup}, \mathsf{sFE.KeyGen}, \mathsf{sFE.Enc}, \mathsf{sFE.Dec})$ that works as follows.

- $\mathsf{sFE.Setup}(1^\lambda)$ runs $\mathsf{FE.Setup}(1^\lambda)$ to generate a key pair $(\mathsf{MSK}, \mathsf{PK})$.
- $\mathsf{sFE.KeyGen}(\mathsf{MSK}, f)$ picks a uniformly random tag $\tau \leftarrow \{0,1\}^\lambda$, symmetric encryptions $\mathsf{CT}_i \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}_i, 0)$ each under a random key $\mathsf{SK}_i \leftarrow \{0,1\}^\lambda$, and constructs a sequence of circuits $\{g_i := g_{f,\tau,\mathsf{CT}_i}\}_{i\in[\mu]}$ which, on input a tuple $(b, x, \mathsf{K}, \mathsf{SK}) \in \{0,1\} \times \{0,1\}^n \times \{0,1\}^\lambda \times \{0,1\}^\lambda$ work as follows:
  - If $b = 0$,
    - Let S_i be the subset of random bits on which $\hat{f}_i(\cdot,\cdot)$ depends.
    - For $j \in S_i$, compute $r_j = \mathsf{PRF}_\mathsf{K}(\tau||j)$,
    - output $e_i \leftarrow \hat{f}_i(x; r_{S_i})$.
  - If $b = 1$,
    - output
    $$e_i \leftarrow \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i)$$

  The functional key for f, denoted sFSK_f, is the set of keys for all the circuits $\{g_{f,\tau,\mathsf{CT}_i}\}_{i=1,2,\ldots,\mu}$ where μ is the output length of the randomized encoding.
- $\mathsf{sFE.Enc}(\mathsf{PK}, x)$ chooses a random PRF key $\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda)$, and outputs
$$\mathsf{FCT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}; (0, x, \mathsf{K}, \perp)) \ .$$
- $\mathsf{sFE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$ parses $\mathsf{sFSK}_f = (\mathsf{FSK}_{g_1}, \ldots, \mathsf{FSK}_{g_\mu})$, computes $e_i \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_{g_i}, \mathsf{FCT})$ and runs the decoder of the randomized encoding on input $(e_1, e_2, \ldots, e_\mu)$ to get $f(x)$.

Correctness follows directly from that of the functional encryption scheme FE and the randomized encoding scheme. We now analyze the efficiency and security of the scheme.

Efficiency: In order to issue N keys in the scheme sFE, we issue $N \cdot \mu = N \cdot s_f \cdot \mathrm{poly}(n, \lambda)$ keys in the underlying scheme FE. Each such key is issued for a circuit g_i of size $\mathrm{poly}(\lambda, n)$. During the encryption of an input x, the encryption algorithm of sFE is invoked on an input of size $n + O(\lambda)$.

Thus, by the collusion-succinctness guarantee of FE, the size of the encryption circuit in sFE is

$$(N \cdot s_f \cdot \mathrm{poly}(n, \lambda))^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda) = (N \cdot s_f)^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda) \ .$$

Remark IV.1. *The collusion-succinct encryption size may also depend on $2^{O(d)}$ where d is the maximal depth of circuits in the class, provided that there exist PRFs in NC^1. Indeed, in the above, the depth of any g_i is dominated by the depth of computing a PRF on a tag of size $O(\lambda)$ and the depth of $\hat{f}_i$ which can be computed in depth $O(\log \log \lambda)$, assuming PRFs in NC^1.*

Security: We now sketch the proof of security, which proceeds by a sequence of hybrids. For simplicity, we consider the case when the adversary submits a single key query for a function f and a single challenge pair (x_0, x_1). The argument can be easily generalized to the case of multiple keys.

$\mathcal{H}_0$*:* This corresponds to the real experiment where the challenger sends an encryption of x_0 to the adversary.

$\mathcal{H}_1$*:* The challenger replaces $\mathsf{CT} = (\mathsf{CT}_1, \ldots, \mathsf{CT}_\mu)$ with a symmetric encryption of the bits of $\widehat{f}(x_0; r)$ in the functional key for f, where $r = (\mathsf{PRF}_K(\tau||1), \ldots, \mathsf{PRF}_K(\tau||\mu))$ is the randomness for the encoding. $\mathcal{H}_1$ is computationally indistinguishable from $\mathcal{H}_0$ based on the semantic security of the symmetric encryption scheme.

$\mathcal{H}_2$*:* The challenge ciphertext will consist of an encryption of $(1, x_0, \perp, \mathsf{SK})$ instead of $(0, x_0, \mathsf{K}, \perp)$. This hybrid is computationally indistinguishable from $\mathcal{H}_1$ by the security of the underlying functional encryption scheme.

$\mathcal{H}_3$*:* For every function query f, the challenger replaces the encryption CT_i in all the functional keys with $\mathsf{Sym.Enc}(\mathsf{SK}_i, \widehat{f}_i(x_0; r_{S_i}))$ for a uniform $r = r_{1,\ldots,\rho}$. $\mathcal{H}_3$ is computationally indistinguishable from $\mathcal{H}_2$ based on the security of the PRF.

$\mathcal{H}_4$*:* The challenger replaces $\widehat{f}(x_0; r)$ in the ciphertext hardwired in the functional key for f by $\widehat{f}(x_1; r)$. $\mathcal{H}_4$ is computationally indistinguishable from $\mathcal{H}_3$ based on the security of randomized encodings and the fact that $f(x_0) = f(x_1)$.

Observing that this hybrid can also be reached symmetrically from a real experiment where x_1 is encrypted, shows indistinguishability, and finishes our proof sketch.

■

The functional encryption scheme of Garg, Gentry, Halevi, and Zhandry [20] satisfies collusion-succinctness and thus we obtain the following corollary

Corollary IV.2. *Under a subexponential variant of the assumptions in [20] on multi-linear graded encodings, there exists an IO construction.*

V. On the Possibility of Basing the Transformation on Symmetric-Key FE

Our transformation in Section III and its proof of security rely on any public-key functional encryption (with proper succinctness). Nevertheless, it may seem that this is just limitation of our proof, and using any symmetric-key scheme instead may be possible. In Section V-A, we show that this is not the case, and that for some symmetric-key schemes our transformation will be insecure. This means that to base IO on symmetric key FE in our transformation one must require additional properties of the symmetric-key scheme. In Section V-B, we formalize a *puncturing property* that is sufficient.

A. Impossibility of Instantiation with Any Symmetric-Key Scheme

We show:

Proposition V.1. *If there exists a succinct symmetric-key functional encryption FE, then there also exists a succinct symmetric-key functional encryption FE^*, so that the transformation given by Theorem III.1 is insecure when instantiated with FE^*.*

To understand the idea behind the above proposition, recall that the core of our transformation is a (recursive) obfuscation of a circuit that given any input $x \in \{0,1\}^n$, produces an FE encryption of x (and some fixed key for a symmetric encryption). However, using similar ideas to those of Barak et al. [1], we can construct a symmetric-key FE scheme where encryption is *unobfuscatable* in the sense that given any encryption circuit as above, it is possible to recover the entire symmetric key.

The formal definitions of symmetric-key FE and unobfuscatable functions as well as the proof of Proposition V.1 are deferred to the full version of this work.

B. Puncturable Symmetric-Key FE is Sufficient

In the previous section, we have shown that it is not possible to instantiate our transformation with any symmetric-key FE scheme. In this section, we give a criterion for symmetric key FE schemes that is sufficient for our transformation to go through. While at this point, we only know how to satisfy this criterion based on public-key FE, it may be constructed directly, without going through public-key FE. (Of course that eventually it does imply the existence of public-key FE, as it leads to IO.)

Specifically, we define puncturable symmetric-key FE, where it is possible puncture the master secret key MSK on a pair of messages m_0, m_1 such that it still allows to encrypt any $m \notin \{m_0, m_1\}$, but does not allow to distinguish encryptions of m_0 and m_1, in the presence of a functional secret-key (that does not separate m_0 and m_1. We restrict the definition to the case of a single functional key, which is sufficient for our purpose.

Definition V.1 (Puncturable symmetric FE)**.** *A single-key symmetric-key functional encryption scheme* FE *is said to be puncturable if there exists an additional algorithms* $\mathsf{FE.Punc}$*,* $\mathsf{FE.PEnc}$ *with the following two properties:*

1) **Correctness:** *For any two equal-length messages* m_0, m_1*, any* MSK *in the support of* $\mathsf{FE.Setup}$*, and any* $m \notin \{m_0, m_1\}$*, it holds that*

$$\mathsf{FE.PEnc}(\mathsf{MSK}\{m_0, m_1\}, m; r) = \mathsf{FE.Enc}(\mathsf{MSK}, m; r) \ ,$$

where $\mathsf{MSK}\{m_0, m_1\} \leftarrow \mathsf{FE.Punc}(\mathsf{MSK}, m_0, m_1)$.

2) **Semantic security at punctured points:** *For any two equal-length messages* m_0, m_1*, and any* f *such that* $f(m_0) = f(m_1)$*:*

$$\{\mathsf{FSK}_f, \mathsf{MSK}\{m_0, m_1\}, \mathsf{FE.Enc}(\mathsf{MSK}, m_0)\} \approx_c \{\mathsf{FSK}_f, \mathsf{MSK}\{m_0, m_1\}, \mathsf{FE.Enc}(\mathsf{MSK}, m_1)\} \ ,$$

where $\mathsf{MSK} \leftarrow \mathsf{FE.Setup}(1^\lambda)$*,* $\mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f)$*, and* $\mathsf{MSK}\{m_0, m_1\} \leftarrow \mathsf{FE.Punc}(\mathsf{MSK}, m_0, m_1)$.

Proposition V.2. *The public-key FE scheme in the transformation given by Theorem III.1 can be replaced by a puncturable symmetric-key FE scheme.*

Proof sketch: The only difference is in the proof of Proposition III.1. When moving from hybrid $\mathcal{H}_{\mathbf{x}-1}$ to hybrid $\mathcal{G}_1$ not only do we puncture the PRF key at $\mathbf{x}$, but we also puncture the master encryption key (now the secret MSK) at $\left\{(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i), \mathsf{SK}_i^1, 1)\right\}$ and hardwire the encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i)$. As in the original analysis, functionality is preserved, this time by the correctness of the puncturable symmetric-key FE. Then, when replacing the encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i)$ with and encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^1, 1), (\mathbf{x}, x_i)$, we rely on semantic security at punctured points. ■

REFERENCES

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, "On the (im)possibility of obfuscating programs," *J. ACM*, vol. 59, no. 2, p. 6, 2012.

[2] S. Goldwasser and Y. T. Kalai, "On the impossibility of obfuscation with auxiliary input," in *FOCS*. IEEE Computer Society, 2005, pp. 553–562.

[3] N. Bitansky, R. Canetti, H. Cohn, S. Goldwasser, Y. T. Kalai, O. Paneth, and A. Rosen, "The impossibility of obfuscation with auxiliary input or a universal simulator," in *CRYPTO*, 2014, pp. 71–89.

[4] S. Garg, C. Gentry, S. Halevi, A. Sahai, M. Raikova, and B. Waters, "Candidate indistinguishability obfuscation and functional encryption for all circuits," in *FOCS*, 2013.

[5] S. Goldwasser and G. N. Rothblum, "On best-possible obfuscation," in *TCC*, 2007, pp. 194–213.

[6] A. Sahai and B. Waters, "How to use indistinguishability obfuscation: deniable encryption, and more." in *STOC*, D. B. Shmoys, Ed. ACM, 2014, pp. 475–484.

[7] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev, "One-way functions and (im)perfect obfuscation," in *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 2014, pp. 374–383.

[8] R. Pass, K. Seth, and S. Telang, "Indistinguishability obfuscation from semantically-secure multilinear encodings," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, ser. Lecture Notes in Computer Science, J. A. Garay and R. Gennaro, Eds., vol. 8616. Springer, 2014, pp. 500–517.

[9] C. Gentry, A. B. Lewko, A. Sahai, and B. Waters, "Indistinguishability obfuscation from the multilinear subgroup elimination assumption," in *FOCS 2015*, 2015.

[10] Z. Brakerski and G. N. Rothblum, "Virtual black-box obfuscation for all circuits via generic graded encoding," in *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, Y. Lindell, Ed., vol. 8349. Springer, 2014, pp. 1–25.

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

[12] B. Applebaum and Z. Brakerski, "Obfuscating circuits via composite-order graded encoding," in *TCC*, 2015.

[13] J. Zimmerman, "How to obfuscate programs directly," in *Eurocrypt*, 2015.

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

[15] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, 2004, pp. 506–522.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, ser. Lecture Notes in Computer Science, R. Cramer, Ed., vol. 3494. Springer, 2005, pp. 457–473.

[17] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: a new vision for public-key cryptography," *Commun. ACM*, vol. 55, no. 11, pp. 56–64, 2012.

[18] A. O'Neill, "Definitional issues in functional encryption," Cryptology ePrint Archive, Report 2010/556, 2010.

[19] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich, "Reusable garbled circuits and succinct functional encryption," Cryptology ePrint Archive, Report 2012/733, 2012.

[20] S. Garg, C. Gentry, S. Halevi, and M. Zhandry, "Fully secure functional encryption without obfuscation," 2014.

[21] J. Coron, T. Lepoint, and M. Tibouchi, "New multilinear maps over the integers," *IACR Cryptology ePrint Archive*, vol. 2015, p. 162, 2015.

[22] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, pp. 533–556.

[23] B. Waters, "A punctured programming approach to adaptively secure functional encryption," *IACR Cryptology ePrint Archive*, vol. 2014, p. 588, 2014.

[24] E. Boyle, K.-M. Chung, and R. Pass, "On extractability obfuscation," in *TCC*, 2014, pp. 52–73.

[25] A. Sahai and H. Seyalioglu, "Worry-free encryption: functional encryption with public keys," in *ACM CCS*, 2010, pp. 463–472.

[26] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Functional encryption with bounded collusions via multi-party computation," in *CRYPTO*, August 2012, pp. 162–179.

[27] ——, "Attribute-based encryption for circuits," in *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, 2013, pp. 545–554.

[28] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Functional encryption: New perspectives and lower bounds," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, ser. Lecture Notes in Computer Science, R. Canetti and J. A. Garay, Eds., vol. 8043. Springer, 2013, pp. 500–518.

[29] S. Gorbunov, V. Vaikuntanathan, and H. Wee, "Predicate encryption for circuits from LWE," *IACR Cryptology ePrint Archive*, vol. 2015, p. 29, 2015.

[30] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," *J. Cryptology*, vol. 26, no. 2, pp. 191–224, 2013.

[31] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," in *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, ser. Lecture Notes in Computer Science, D. H. Lee and X. Wang, Eds., vol. 7073. Springer, 2011, pp. 21–40.

[32] Z. Brakerski and G. Segev, "Function-private functional encryption in the private-key setting," in *TCC*, 2015.

[33] U. Feige and A. Shamir, "Zero knowledge proofs of knowledge in two rounds," in *CRYPTO*, 1989, pp. 526–544.

[34] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *STOC*, 1990, pp. 427–437.

[35] R. Canetti, H. Lin, S. Tessaro, and V. Vaikuntanathan, "Obfuscation of probabilistic circuits and applications," in *TCC*, 2015.

[36] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[37] P. Ananth and A. Jain, "Indistinguishability obfuscation from compact functional encryption," in *Crypto*, 2015.

[38] S. Goldwasser, S. D. Gordon, V. Goyal, A. Jain, J. Katz, F. Liu, A. Sahai, E. Shi, and H. Zhou, "Multi-input functional encryption," in *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, pp. 578–602.

[39] Z. Brakerski, I. Komargodski, and G. Segev, "From single-input to multi-input functional encryption in the private-key setting," *IACR Cryptology ePrint Archive*, vol. 2015, p. 158, 2015.

[40] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions." *J. ACM*, vol. 33, no. 4, pp. 792–807, 1986.

[41] D. Boneh and B. Waters, "Constrained pseudorandom functions and their applications," in *ASIACRYPT (2)*, ser. Lecture Notes in Computer Science, K. Sako and P. Sarkar, Eds., vol. 8270. Springer, 2013, pp. 280–300.

[42] A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias, "Delegatable pseudorandom functions and applications." in *CCS*, A. Sadeghi, V. D. Gligor, and M. Yung, Eds. ACM, 2013, pp. 669–684.

[43] E. Boyle, S. Goldwasser, and I. Ivan, "Functional signatures and pseudorandom functions." in *PKC*, ser. Lecture Notes in Computer Science, H. Krawczyk, Ed., vol. 8383. Springer, 2014, pp. 501–519.

[44] Y. Ishai and E. Kushilevitz, "Randomizing polynomials: A new representation with applications to round-efficient secure computation," in *FOCS*. IEEE Computer Society, 2000, pp. 294–304.

[45] B. Applebaum, Y. Ishai, and E. Kushilevitz, "Computationally private randomizing polynomials and their applications," *Computational Complexity*, vol. 15, no. 2, pp. 115–162, 2006.

[46] A. C. Yao, "How to generate and exchange secrets (extended abstract)," in *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*. IEEE Computer Society, 1986, pp. 162–167.

[47] B. Applebaum, "Bootstrapping obfuscators via fast pseudorandom functions," in *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, ser. Lecture Notes in Computer Science, P. Sarkar and T. Iwata, Eds., vol. 8874. Springer, 2014, pp. 162–172.

[48] P. Ananth, Z. Brakerski, G. Segev, and V. Vaikuntanathan, "The trojan method in functional encryption: From selective to adaptive security, generically," in *CRYPTO*, 2015.