Indistinguishability Obfuscation from Functional Encryption*

Nir Bitansky† Vinod Vaikuntanathan‡

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.

*An extended abstract of this paper will appear in the proceedings of FOCS 2015.

†MIT. E-mail: nirbitan@csail.mit.edu. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119.

‡MIT. E-mail: vinodv@csail.mit.edu. Research 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.

Contents

1 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 [BGI+12], was shown to have strong limitations [BGI+12, GK05, BCC+14]. 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 [GGH+13b] demonstrated a candidate obfuscation algorithm for all circuits, and conjectured that it satisfies an apparently weak notion of *indistinguishability obfuscation* (IO) [BGI+12, GR07], 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 [SW14], 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 [PST14, GLSW15], all known IO constructions [GGH+13b, PST14, BR14, BGK+14, GLSW15, AB15, Zim15] are still based on the hardness of little-studied problems on multi-linear maps [GGH13a]. Thus, an outstanding foundational question in cryptography is:

Can we base indistinguishability obfuscation on strong cryptographic foundations?

1.1 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 [BCOP04, SW05, BSW12, O'N10], 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" [GKP+12].

Rather surprisingly, we show:

Theorem 1.1 (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.*

[1] Strictly speaking, we need the assumption that IO exists, plus a very mild (and minimal) complexity-theoretic assumption that NP $\neq$ ioBPP [KMN+14].

[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.

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. We can further relax the above to allow the encryption to also depend polynomially on circuit-depth. Assuming (sub-exponential) puncturable pseudo-random functions in NC^1, even let encryption 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. [GGHZ14], 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 [CLT15].

Corollary 1.2. *Under a subexponential variant of the assumptions in [GGHZ14] 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 [BGG+14] is that obfuscation size can always be reduced to linear in the size circuit plus some overhead in the circuit's depth.

Corollary 1.3. *Assuming subexponential LWE and IO, there exists IO such that an obfuscation of any circuit C is of size $2|C| + \mathrm{poly}(n, \mathsf{dep}(C), \lambda)$.*

Interpretation. Functional encryption schemes satisfying the ciphertext compactness required in Theorem 1.1 are also known based on indistinguishability obfuscation [GGH+13b, Wat14] or the stronger notion of differing-inputs obfuscation [BCP14]. 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 [SS10, GVW12, GVW13]. The state of the art scheme based on a standard assumption is that of Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [GKP+12] 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 [AGVW13, GKP+12]. 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 [GVW15] 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 [KSW13, AFV11, GVW15] for more details). Strengthening this to "full attribute hiding" will give us a functional encryption scheme that satisfies the requirements of Theorem 1.1, and is yet another frontier in achieving indistinguishability obfuscation from LWE.

1.2 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 [GKP+12], and generalized in [BS15], 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 [GKP+12] 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. [BGI+12], 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 [BS15] going back to the classic two-key paradigm [FS89, NY90].

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 [BS15]. 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 [CLTV15] 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 $\mathsf{FSK}_1, \dots, \mathsf{FSK}_n$ 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 $x_1 \dots x_{i-1}x_i$ between the two encryptions of $x_1 \dots x_{i-1}0$ and $x_1 \dots x_{i-1}1$ 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 also polynomially on the depth, or even exponentially 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 1.1. *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 1.1 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 5 for more details). The real power of obfuscation manifests itself in transforming private-key schemes into public-key schemes [DH76], and for this reason, we believe that finding a (different) transformation from private-key FE to IO is a central open question.

1.3 Concurrent work

In a concurrent and independent work Ananth and Jain [AJ15] 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. [GGG+14] 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 [CLTV15].

In another concurrent work [BKS15], 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.

2 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.

2.1 Functional Encryption

We recall the definition of public-key functional encryption (FE) with selective indistinguishability-based security [BSW12, O'N10].

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 2.1 (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* $\{0,1\}^n$*,*

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

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*

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

where for each $b \in \{0,1\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}}(1^\lambda, b)$, 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 2.1, they can be restated more simply.

Definition 2.2 (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 $\{0,1\}^n$,*

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

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 $f(m_0) = f(m_1)$,*

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

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** *with* $\Delta(d)$*-dependence on the depth if the size of the encryption circuit is bounded by* $s^{1-\varepsilon} \cdot \mathrm{poly}(n, \lambda, \Delta(d))$ *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.*

2.2 Indistinguishability Obfuscation

We define indistinguishability obfuscation (IO) with respect to a give class of circuits. The definition is formulated as in [BGI$^+$12].

Definition 2.3 (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*,*

$$\Pr_{i\mathcal{O}} \left[\forall x : i\mathcal{O}(C, 1^\lambda)(x) = C(x) \right] = 1 \enspace .$$

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:*

$$\left| \Pr[\mathcal{D}(i\mathcal{O}(C_0, 1^\lambda)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(C_1, 1^\lambda)) = 1] \right| \le \mu(\lambda) \enspace ,$$

   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* $\delta(\lambda)^{\Omega(1)}$*.*

2.3 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 [SW14], and is satisfied by the GGM [GGM86] PRF [BW13, KPTZ13, BGI14].

Definition 2.4 (Puncturable PRFs)**.** *Let* n, k *be polynomially bounded length functions. An efficiently computable family of functions*

$$\mathcal{PRF} = \left\{ \mathsf{PRF}_{\mathsf{K}} : \{0,1\}^* \to \{0,1\}^\lambda \;\middle|\; \mathsf{K} \in \{0,1\}^{k(\lambda)}, \lambda \in \mathbb{N} \right\} \enspace ,$$

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* $x^* \in \{0,1\}^*$*,*

$$\Pr_{\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda)} \left[\forall x \neq x^* : \mathsf{PRF}_{\mathsf{K}}(x) = \mathsf{PRF}_{\mathsf{K}\{x^*\}}(x) \;\middle|\; \mathsf{K}\{x^*\} = \mathsf{Punc}(\mathsf{K}, x^*) \right] = 1 \enspace .$$

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 $x^* \in \{0,1\}^*$,*

$$|\Pr[\mathcal{D}(x^*, \mathsf{K}\{x^*\}, \mathsf{PRF}_{\mathsf{K}}(x^*)) = 1] - \Pr[\mathcal{D}(x^*, \mathsf{K}\{x^*\}, u) = 1]| \le \mu(\lambda) \ ,$$

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 $\delta(\lambda)^{\Omega(1)}$.

2.4 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 2.5 (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,*

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

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 $\delta(\lambda)^{\Omega(1)}$.*

2.5 Randomized Encodings

We rely on the notion of randomized encodings from [IK00, AIK06]. Let $c \ge 1$ 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 [Yao86]. 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) [AIK06].

3 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.

Theorem 3.1. *$i\mathcal{O}$ 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.

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 Section 3.2, 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

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

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
   $$f_i(\mathbf{x}_i, \mathsf{SK}, \beta) = U(\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i^{\beta}), \mathbf{x}_i) \enspace ,$$
   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
   $$\mathsf{E}_{i-1}^0(\mathbf{x}_{i-1}) = \left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}_{i-1}, x_i), \mathsf{SK}_i^0, 0); \mathsf{PRF}_{\mathsf{K}_i}(\mathbf{x}_{i-1}, x_i))\right\}_{x_i \in \{0,1\}} \enspace ,$$
   padded to some size $\ell(\tilde{\lambda})$ for some polynomial $\ell(\cdot)$ determined in the analysis.
   - An obfuscation
   $$\widetilde{\mathsf{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathsf{E}_{i-1}^0, 1^{\tilde{\lambda}}) \enspace .$$
3. Output $\widetilde{\mathsf{E}}_i := (\widetilde{\mathsf{E}}_{i-1}, \mathsf{FSK}_i)$.

Figure 1: The recursive obfuscation procedure.

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

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.

time to obfuscate (and size of the resuting obfuscation) is bounded by a fixed polynomial $\mathrm{poly}(|C|, \lambda)$ as required.

3.1 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

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

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 $1 \le i < n(\lambda)$, define

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

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

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

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

Proof of Proposition 3.1. 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

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

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_i^0 and CT_i^1 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 $\mathcal{H}_\beta^{\gamma_0,\gamma_1}$ where

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

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

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

In the first and third inequalities, we simply change the symmetrically encrypted plaintext in some CT_i^b where only the key SK_i^{1-b} 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 $\mathcal{H}_0^{0,1}$ and $\mathcal{H}_1^{0,1}$ is in the obfuscated $\widetilde{\mathsf{E}}_{i-1}$. In the first, the circuit E_{i-1}^0, which always puts SK_i^0 in the plaintext, is obfuscated, and in the second E_{i-1}^1, which always puts SK_i^1 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 $\mathsf{CT}_i^0, \mathsf{CT}_0^1$ 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 [CLTV15] show that sub-exponential IO in conjunction with sub-exponential puncuturable 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 $\{\mathcal{H}_\mathbf{x}\}_{\mathbf{x}\in\{0,\dots,2^{i-1}\}}$, where we naturally identify integers in $[2^{i-1}]$ with strings in $\{0,1\}^{i-1}$. In $\mathcal{H}_\mathbf{x}$, both CT_i^0 and CT_i^1 encrypt the same circuit $\mathsf{E}_\mathbf{x}(\mathbf{x}')$ that computes $\mathsf{E}_{i-1}^0(\mathbf{x}')$ for all $\mathbf{x}' > \mathbf{x}$ and $\mathsf{E}_{i-1}^1(\mathbf{x}')$ for all $\mathbf{x}' \le \mathbf{x}$; 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_{i-1}^0 and that $\mathsf{E}_{2^{i-1}}$ computes the same function as E_{i-1}^1, and thus

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

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

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

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

- $\mathcal{G}_1$: instead of $\mathsf{E}_{\mathbf{x}-1}$, $\mathsf{CT}_i^0, \mathsf{CT}_i^1$ both encrypt $\mathsf{E}'_{\mathbf{x}-1}$ that has
$$\mathsf{E}_{\mathbf{x}-1}(\mathbf{x}) = \mathsf{E}^0_{i-1}(\mathbf{x}) = \left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); \mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}, x_i)) \;\middle|\; x_i \in \{0,1\}\right\}$$
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:
$$|\Pr[\mathcal{D}(\mathcal{H}_{\mathbf{x}-1}) = 1] - \Pr[\mathcal{D}(\mathcal{G}_1) = 1]| \leq \delta_{i-1} \ .$$
- $\mathcal{G}_2$: Here we replace the hardwired
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); \mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}, x_i)) \;\middle|\; x_i \in \{0,1\}\right\}$$
so that instead of using the pseudo-randomness $\mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}, x_i)$, true randomness r is used
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); r) \;\middle|\; x_i \in \{0,1\}\right\} \ .$$
By pseudo-randomness at punctured points
$$|\Pr[\mathcal{D}(\mathcal{G}_1) = 1] - \Pr[\mathcal{D}(\mathcal{G}_2) = 1]| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ .$$
- $\mathcal{G}_3$: Here we replace the hardwired
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); r) \;\middle|\; x_i \in \{0,1\}\right\}$$
to encrypt $(\mathsf{SK}_i^1, 1)$ instead of $(\mathsf{SK}_i^0, 0)$:
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^1, 1); r) \;\middle|\; x_i \in \{0,1\}\right\} \ .$$
Since, CT_i^0 and CT_i^1 encrypt circuits C_0 and C_1, respectively, with the exact same functionality, we can apply the FE guarantee to deduce
$$|\Pr[\mathcal{D}(\mathcal{G}_2) = 1] - \Pr[\mathcal{D}(\mathcal{G}_3) = 1]| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ .$$
- $\mathcal{G}_{2'}$: reverses $\mathcal{G}_2$, we replace the hardwired
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^1, 1); r) \;\middle|\; x_i \in \{0,1\}\right\}$$
with
$$\left\{\mathsf{FE.Enc}(\mathsf{PK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^1, 1); \mathsf{PRF}_{\mathsf{K}} i(\mathbf{x}, x_i)) \;\middle|\; x_i \in \{0,1\}\right\} \ .$$
By pseudo-randomness at punctured points
$$|\Pr[\mathcal{D}(\mathcal{G}_3) = 1] - \Pr[\mathcal{D}(\mathcal{G}_{2'}) = 1]| \leq 2^{-\Omega(\tilde{\lambda}^{\varepsilon})} \ .$$
- 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 3.2. 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. □

3.2 Extended Efficiency Analysis

So far, we have analyzed the efficiency of our obfuscator, assuming that the functional encryption scheme is fully succinct, namely, the running time of the encryption algorithm is 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, circuit-depth, or output-size of functions (here $\tilde{\lambda} = \omega((n^2 + \log \lambda)^{1/\varepsilon})$, for the security parameter λ). In this section, we show how the efficiency of our transformation can still be maintained assuming weak succinctness where there is some dependence on these parameters.

Theorem 3.3. *Assuming the existence of a sub-exponentially secure public-key functional encryption scheme for all circuits that is weakly succinct with* $\mathrm{poly}(d)$*-dependence on depth, there exists indistinguishability obfuscation for all circuits. If there exists puncturable pseudorandom functions in NC*^{1}*, then this also holds with* $\mathrm{poly}(2^d)$*-dependence.*

We first show that efficiency is still guaranteed if the size of the encryption circuit (and ciphertext size) can grow sub-linearly with the circuit size of functions. Namely, encryption size is at most

$$s^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \ ,$$

where $s = \max_{f \in \mathcal{F}_n} |f|$, and $\varepsilon < 1$ is some constant, and poly is any fixed polynomial. For simplicity of exposition, we will first show that this holds for some ε related to the underlying cryptographic primitives. We will then observe that it can be made to hold for any $\varepsilon < 1$.

First, we note that the size of each circuit f_i for which a functional key FSK_i is derived is bounded by

$$|f_i| \leq |\mathsf{E}_i^0|^c \cdot \mathrm{poly}(\tilde{\lambda}) \ ,$$

where poly is some fixed polynomial and c is some constant that depends on the efficiency of symmetric decryption, and application of the universal circuit.

Recall that E_i^0 first derives pseudo-randomness using a puncturable PRF, and then encrypts with respect to PK_{i+1} a plaintext of size at most $n + \tilde{\lambda}$. By our assumption on the succinctness of the scheme, the size of the encryption circuit is bounded by $|f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda})$, which also dominates the size of deriving randomness using a PRF (up to $\mathrm{poly}(n, \tilde{\lambda})$ factors).

Using the above bound on each f_i, we can now bound the size of each circuit E_i^0 as follows

$$|\mathsf{E}_i^0| \leq |f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \leq |\mathsf{E}_{i+1}^0|^{c(1-\varepsilon)} \cdot \mathrm{poly}(n, \tilde{\lambda}) \ .$$

Also,

$$|\mathsf{E}_{n-1}^0| \leq |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \ ,$$

where C is the obfuscated circuit.

It follows that,

$$|\mathsf{E}_i^0| \leq |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \cdot \prod_{j=0}^{n-i-1} \left(\mathrm{poly}(n, \tilde{\lambda})\right)^{(c(1-\varepsilon))^j} \ .$$

Now, provided that $c(1-\varepsilon) < 1$, for any $k, p \in \mathbb{N}$,

$$\prod_{j=0}^{k} p^{(c(1-\varepsilon))^j} = p^{\sum_{j=0}^{k} (c(1-\varepsilon))^j} \leq p^{\frac{1}{1-c(1-\varepsilon)}} \ .$$

We conclude that

$$\max_i |\mathsf{E}_i^0| \leq |C|^{1-\varepsilon} \cdot \left(\mathrm{poly}(n, \tilde{\lambda})\right)^{\frac{1}{1-c(1-\varepsilon)}+1} \ .$$

Efficiency now follows, for any $\varepsilon < 1 - \frac{1}{c}$, as in the case of total independence of the circuit size.

Remark 3.4 (Efficiency for any $\varepsilon < 1$). Looking more closely at the complexity of f_i we observe that we can assure that $c = 1 + o(1)$. Indeed, c accounts for symmetric decryption and the application of a universal circuit. First, recall that the overhead of universal circuits is known to be quasi-linear [Val76]. Second, note that symmetric-decryption can be done in time linear in the plaintext and polynomial in the security parameter using pseudo-random generators that are linear in their output size and polynomial in the security parameter (which in turn can be constructed from any PRF).

We next show that we can also allow polynomial dependence on the circuit depth. Concretely, we shall assume that encryption size is bounded by

$$s^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, d) \ \ ,$$

where $s = \max_{f \in \mathcal{F}_n} |f|$, $d = \max_{f \in \mathcal{F}_n} \mathsf{dep}(f)$, and $\varepsilon < 1$.

Throughout, we shall assume that the encryption circuit is shallow. Concretely, we can always make it shallow by computing a randomized encoding of the encryption circuit (Section 2.5). The concrete depth is determined by the depth in which a linear-stretch PRG can be computed, which is some fixed $\mathrm{poly}(\tilde{\lambda})$. We also note that this adaptation to the encryption scheme does not increase the size of the encryption circuit except by a fixed $\mathrm{poly}(\tilde{\lambda})$ factor.

We now go through a similar calculation to the one before (where we did not assume depth dependence), while now also analyzing the depth of the involved circuits rather than only their size.

First, we bound the size and depth of each circuit f_i for which a functional key FSK_i is derived:

$$|f_i| \le |\mathsf{E}_i^0| \cdot \mathrm{poly}(\tilde{\lambda}) \ \ ,$$
$$\mathsf{dep}(f_i) \le \mathsf{dep}(\mathsf{E}_i^0) + \mathrm{poly}(\tilde{\lambda}) \ \ .$$

Both bounds account for decryption and the application of a universal circuit. The depth of the universal circuit can be linear in the depth of the circuit E_i^0 which it emulates [CH85]. Decryption depth is determined by the depth of computing a PRG that stretches $\tilde{\lambda}$ bits to at most $|\mathsf{E}_i^0|$ bits. This can be done in depth $\mathrm{poly}(\tilde{\lambda})$ (by invoking the PRF for each of the output bits).

Recall that E_i^0 first derives pseudo-randomness using a puncturable PRF, and then encrypts with respect to PK_{i+1} a plaintext of size at most $n + \tilde{\lambda}$. By our assumption on the succinctness of the scheme, the size of the encryption circuit is bounded by $|f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, \mathsf{dep}(f_{i+1}))$, which also dominates the size of deriving randomness using a PRF (up to $\mathrm{poly}(n, \tilde{\lambda})$ factors). Thus:

$$|\mathsf{E}_i^0| \le |f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, \mathsf{dep}(f_{i+1})) \le |\mathsf{E}_{i+1}^0|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, \mathsf{dep}(\mathsf{E}_{i+1}^0))$$
.

According to our assumption on the encryption algorithm it is of depth $\mathrm{poly}(\tilde{\lambda})$ and the depth of applying a puncturable PRF to an input of size $O(n)$ is $\mathrm{poly}(n, \tilde{\lambda})$ and overal:

$$\mathsf{dep}(\mathsf{E}_i^0) \le \mathrm{poly}(n, \tilde{\lambda}) \ \ .$$

This implies that

$$|\mathsf{E}_i^0| \le |\mathsf{E}_{i+1}^0|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \ \ .$$

This reduces the analysis to the previous case (with no dependence on the depth).

We next show that we can also allow exponential dependence on the circuit depth if we assume puncturable PRFs in NC^1. Here we can only obfuscate circuits in NC^1, and then bootstrap to get IO for arbitrary circuits as in previous works [GGH+13b, CLTV15].

Concretely, we shall assume that encryption size is bounded by

$$s^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^d) \ ,$$

where $s = \max_{f \in \mathcal{F}_n} |f|$, $d = \max_{f \in \mathcal{F}_n} \mathsf{dep}(f)$, and $\varepsilon < 1$.

Throughout, we shall assume that the encryption circuit is shallow. Indeed, we can always make it shallow by computing a randomized encoding of the encryption circuit (Section 2.5). The concrete depth is determined by the depth in which a linear-stretch PRG can be computed. Since we assume PRFs in NC^1, we can assume such a PRG of depth $O(\log \tilde{\lambda})$. We note that this adaptation to the encryption scheme does not increase the size of the encryption circuit except by a $\mathrm{poly}(\tilde{\lambda})$ factor.

We now go through a similar calculation to the one before (where we did not assume depth dependence), while now also analyzing the depth of the involved circuits rather than only their size.

First, we bound the size and depth of each circuit f_i for which a functional key FSK_i is derived:

$$|f_i| \le |\mathsf{E}_i^0| \cdot \mathrm{poly}(\tilde{\lambda}) \ ,$$
$$\mathsf{dep}(f_i) \le O(\mathsf{dep}(\mathsf{E}_i^0) + \log\log|\mathsf{E}_i^0|).$$

Both bounds account for decryption and the application of a universal circuit. The depth of the universal circuit can be linear in the depth of the circuit E_i^0 which it emulates [CH85]. Decryption depth is determined by the depth of computing a PRG that stretches $\tilde{\lambda}$ bits to at most $|\mathsf{E}_i^0|$ bits. Assuming PRFs in NC^1 this can be done in depth $\log\log|\mathsf{E}_i^0|$ (by invoking the PRF for each of the $|\mathsf{E}_i^0|$ output bits).

Recall that E_i^0 first derives pseudo-randomness using a puncturable PRF, and then encrypts with respect to PK_{i+1} a plaintext of size at most $n+\tilde{\lambda}$. By our assumption on the succinctness of the scheme, the size of the encryption circuit is bounded by $|f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(f_{i+1})})$, which also dominates the size of deriving randomness using a PRF (up to $\mathrm{poly}(n, \tilde{\lambda})$ factors). The depth of encryption is $O(\log \tilde{\lambda})$ and, under the assumption that the puncturable PRF is in NC^1, the depth of deriving randomness is of depth is $O(\log\log|\mathsf{E}_i^0|)$.

Using the above bounds on the depth and size of each f_i, we can now bound the size and depth of each circuit E_i^0 as follows

$$\begin{aligned}
&|\mathsf{E}_i^0| \le |f_{i+1}|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(f_{i+1})}) \le |\mathsf{E}_{i+1}^0|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(\mathsf{E}_{i+1}^0)}) \\
&\mathsf{dep}(\mathsf{E}_i^0) \le O(\log \tilde{\lambda} + \log\log|\mathsf{E}_i^0|) \le \\
&O(\log \tilde{\lambda} + \log\log(|\mathsf{E}_{i+1}^0|) + \log\log \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(\mathsf{E}_{i+1}^0)})) \le \\
&O(\log n + \log \tilde{\lambda} + \log \mathsf{dep}(\mathsf{E}_{i+1}^0)) \ .
\end{aligned}$$

Also,

$$\begin{aligned}
\mathsf{dep}(\mathsf{E}_{n-1}^0) \le O(\log n + \log \tilde{\lambda} + \log \mathsf{dep}(C)) \le O(\log n + \log \tilde{\lambda}) \ , \\
|\mathsf{E}_{n-1}^0| \le |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(C)}) \le |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \ ,
\end{aligned}$$

where C is the obfuscated circuit (of depth $O(\log n)$).

It follows that,

$$\begin{aligned}
\mathsf{dep}(\mathsf{E}_i^0) &\le O(\log \tilde{\lambda} + \log \mathsf{dep}(\mathsf{E}_{n-1}^0)) \le O(\log n + \log \tilde{\lambda}) \ , \\
|\mathsf{E}_i^0| &\le |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(\mathsf{E}_{i+1}^0)}) \cdot \prod_{j=0}^{n-i-1} \left(\mathrm{poly}(n, \tilde{\lambda}, 2^{\mathsf{dep}(\mathsf{E}_{i+1}^0)})\right)^{(1-\varepsilon)^j} \\
&\le |C|^{1-\varepsilon} \cdot \mathrm{poly}(n, \tilde{\lambda}) \cdot \prod_{j=0}^{n-i-1} \left(\mathrm{poly}(n, \tilde{\lambda})\right)^{(1-\varepsilon)^j}
\end{aligned}$$

.

We conclude that

$$\max_i \mathsf{dep}(\mathsf{E}_i^0) \leq O(\log n + \log \tilde{\lambda}) \ ,$$
$$\max_i |\mathsf{E}_i^0| \leq |C|^{1-\varepsilon} \cdot \left(\mathrm{poly}(n, \tilde{\lambda})\right)^{\frac{1}{\varepsilon}+1} \ .$$

Efficiency follows as in the case of total independence of the circuit size.

3.3 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 3.2 (FE with succinct keys [BGG+14])**.** *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 3.5. *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 $2s + \mathrm{poly}(n, d, \lambda)$.*

4 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 2.2), 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 [GGHZ14], we obtain an IO construction based on a subexponential variant of the assumptions in [GGHZ14] 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 [GVW12, App14, ABSV15]. For simplicity, we describe everything in terms of polynomial security. The transformation can be naturally scaled for the case of sub-exponential security.

Proposition 4.1. *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 2.5). 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 4.1. 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 [GGHZ14] satisfies collusion-succinctness and thus we obtain the following corollary

Corollary 4.2. *Under a subexponential variant of the assumptions in [GGHZ14] on multi-linear graded encodings, there exists an IO construction.*

5 On the Possibility of Basing the Transformation on Symmetric-Key FE

Our transformation in Section 3 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 5.1, 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 5.2, we formalize a *puncturing property* that is sufficient.

5.1 Impossibility of Instantiation with Any Symmetric-Key Scheme

We show:

Proposition 5.1. *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 3.1 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. [BGI+12], 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.

We first define symmetric-key FE and unobfuscatable functions.

Symmetric-key FE. The definition of symmetric-key FE naturally restricts the public-key Definition 2.2. Concretely there is one mater symmetric-key MSK that is used for encryption, decryption, and key-derivation.

Definition 5.1 (Selectively-secure symmetric-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 symmetric-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 $\{0,1\}^n$,*

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

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*

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

where for each $b \in \{0,1\}$ and $\lambda \in \mathbb{N}$ the experiment $\mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}}(1^\lambda, b)$, 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 MSK. It then executes $\mathsf{FE.Enc}(\mathsf{MSK}, m_b)$ to obtain CT. The challenger sends 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 adversary may also submit encryption queries. For any query m , the challenger generates and sends $\mathsf{FE.Enc}(\mathsf{MSK}, m)$ to the adversary. (Encryption queries and key queries can be interleaved.)*

   (e) *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 $\delta(\lambda)^{\Omega(1)}$.

Unobfuscatable functions. We define unobfuscatable functions and state several useful facts.

Definition 5.2 (Unobfuscatable family)**.** *An efficiently computable family of functions $\mathcal{F} = \{f_K\}_{K \in \{0,1\}^n, n \in \mathbb{N}}$ is unobfuscatable if it is:*

1. **Non-Black-box learnable:** *There exists a poly time extractor E, such that for any $K \in \{0,1\}^n, n \in \mathbb{N}$ and any circuit C that computes f_K,*

$$E(C) = K \enspace .$$

2. **Black-box unlearanble:** *For any poly-size oracle-aided adversary $\mathcal{A}$ and all $n \in \mathbb{N}$, there exists a negligible function $\mu(\lambda)$, such that*

$$\Pr_{K \leftarrow \{0,1\}^n}\left[\mathcal{A}^{f_K}(1^n) = K\right] \leq \mu(n) \enspace .$$

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

Theorem 5.3 ([BGI+12])**.** *Assuming one-way functions, there exists unobfuscatable function families. Furthermore, there exist pseudo-random function families that are unobfuscatable.*

For our purposes we will consider strong unobfuscatable functions where a part of the key is even semantically secure.

Definition 5.4 (Strongly unobfuscatable family)**.** *An efficiently computable family of functions $\mathcal{F} = \{f_{K,S}\}_{K,S \in \{0,1\}^n, n \in \mathbb{N}}$ is strongly unobfuscatable if it is:*

1. **Non-Black-box learnable:** *There exists a poly time extractor E, such that for any $K, S \in \{0,1\}^n, n \in \mathbb{N}$ and any circuit C that computes $f_{K,S}$,*

$$E(C) = S \enspace .$$

2. **Black-box strongly unlearanble:** *For any poly-size oracle-aided adversary $\mathcal{A}$ and all $n \in \mathbb{N}$, there exists a negligible function $\mu(\lambda)$, such that for any S_0, S_1*

$$\left| \Pr_{K \leftarrow \{0,1\}^n}\left[\mathcal{A}^{f_{K,S_0}}(S_0, S_1) = 1\right] - \Pr_{K \leftarrow \{0,1\}^n}\left[\mathcal{A}^{f_{K,S_1}}(S_0, S_1) = 1\right] \right| \leq \mu(n) \enspace .$$

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

Unobfsucatable families directly imply strongly unobfuscatable ones, and likewise unobfuscatable psuedo-random families imply strongly unobfsucatable ones. For instance, we can extend any unobfsucatable family $\{f_K\}$ to $\{g_{K,K',S}\}$ where K is a key for the original family and K' is a seed for a PRF f', and each function is given by

$$g_{K,K',S}(x,y) = \begin{cases} f_K(x), & y = 0^{|K|}; \\ S & y = K; \\ f'_{K'}(x,y), & \text{o.w.} \end{cases}$$

Assuming f, f' are pseudo-random clearly so is the new function family g. Also, given any circuit that computes $g_{K,K',S}$, one can derive a circuit that computes f_K, extract the key K and then obtain S. Also, any black-box learner $\mathcal{A}^{g_{K,K',S_b}}(S_0, S_1)$ that can predict b, must query its oracle on an input (x, y), where $y = K$ and can be translated into a learner for f_K.

We are now ready to prove Proposition 5.1.

Proof sketch of Proposition 5.1. Let FE be a (succinct) symmetric-key FE scheme, with encryption algorithm by $\mathsf{FE.Enc}(\mathsf{MSK}, x; r)$, key derivation algorithm $\mathsf{FE.Gen}(\mathsf{MSK}, f)$, as well as functional decryption $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$. Let $\mathcal{F} = \{f_{K,S}\}$ be a strongly unobfuscatable family of pseudorandom functions. We define a new FE scheme FE^* as follows:

- A master key $\mathsf{MSK}^* = (K, \mathsf{MSK})$ consists of a random $K \leftarrow \{0,1\}^n$, and a master secret-key MSK for FE.

- $\mathsf{FE.Enc}^*(\mathsf{MSK}^*, x; r) = (\mathsf{FCT}, y)$, where $\mathsf{FCT} = \mathsf{FE.Enc}(\mathsf{MSK}, x; r)$, $y = f_{K,\mathsf{MSK}}(x)$.

- $\mathsf{FE.Gen}^*(\mathsf{MSK}^*, f) = \mathsf{FSK}_f$, where $\mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f)$.

- $\mathsf{FE.Dec}^*(\mathsf{FSK}_f, (\mathsf{FCT}, y)) = \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$.

First, note that FE^* recovers the required functionality of a symmetric-key decryption scheme, ignoring the second part y of ciphertexts. Also, FE^* is succinct since the unobfuscatable family $\mathcal{F}$ can be computed in fixed polynomial time. Further, we claim that the scheme FE^* is a secure symmetric-key FE just as the original scheme FE. To see this, note that by strong unobfuscatability, we can indistinguishably move to an alternative scheme FE^*_1, where encryption sets $y = f_{K,\mathsf{MSK}_1}(x)$ for MSK_1 that is completely independent of MSK. Indeed, defining $\mathsf{MSK}_0 = \mathsf{MSK}$, and $\mathsf{FE}^*_0 = \mathsf{FE}^*$, we note that given an oracle to f_{K,MSK_b} and MSK_b, it is possible to exactly simulate FE^*_b. Furthermore, FE^*_1 is clearly as secure as the original FE, as the addition of y is independent of the underlying key MSK, and pseudorandom given the plaintext x. It follows that $\mathsf{FE}^*_0 = \mathsf{FE}^*$ is also secure. (Also, note that if FE is δ secure and $\mathcal{F}$ is δ-strongly-unobfuscatable, then FE^* is δ-secure.)

At the same time, we claim that given *any* circuit C that computes $\mathsf{FE.Enc}^*(\mathsf{MSK}^*, x)$, derandomized in some arbitrary way (for instance, using a puncturable PRF as in our transformation, it is possible to recover MSK and thus break FE^*. Indeed, any such code can be used to derive a code that computes $f_{K,\mathsf{MSK}}$ from which it is possible to efficiently extract MSK by the non-black-box learnability property.

As mentioned above, the transformation in Section 3 actually provides the attacker with a circuit that computes $\mathsf{FE.Enc}^*(\mathsf{MSK}^*, (x, \mathsf{SK}))$, where SK is a key for a standard symmetric encryption scheme. We can, however, construct the above FE^* so to append only $f_{K,\mathsf{MSK}}(x)$, rather than $f_{K,\mathsf{MSK}}(x, \mathsf{SK})$. Instantiating our transformation with such a scheme would allow to obtain MSK, and thus also SK, and completely reveal the obfuscated circuit (which is encrypted under SK). □

5.2 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 5.5 (Puncturable symmetric FE). *A single-key symmetric-key functional encryption scheme* FE *is said to be puncturable if there exists an additional algorithms* FE.Punc, 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* 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* $f(m_0) = f(m_1)$*:*

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

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 5.2. *The public-key FE scheme in the transformation given by Theorem 3.1 can be replaced by a puncturable symmetric-key FE scheme.*

Proof sketch. The only difference is in the proof of Proposition 3.1. 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 $\{(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i), \mathsf{SK}_i^1, 1)\}$ 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

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In *TCC*, 2015.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. The trojan method in functional encryption: From selective to adaptive security, generically. In *CRYPTO*, 2015.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *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*, volume 7073 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2011.

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 500–518. Springer, 2013.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Crypto*, 2015.

[App14] Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In Palash Sarkar and Tetsu Iwata, editors, *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*, volume 8874 of *Lecture Notes in Computer Science*, pages 162–172. Springer, 2014.

[BCC+14] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. In *CRYPTO*, pages 71–89, 2014.

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe 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*, pages 506–522, 2004.

[BCP14] Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *TCC*, pages 52–73, 2014.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran 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*, pages 533–556, 2014.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC*, volume 8383 of *Lecture Notes in Computer Science*, pages 501–519. Springer, 2014.

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit 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*, pages 221–238, 2014.

[BKS15] Zvika Brakerski, Ilan Komargodski, and Gil Segev. From single-input to multi-input functional encryption in the private-key setting. *IACR Cryptology ePrint Archive*, 2015:158, 2015.

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014, San Diego, CA, USA, February 24-26, 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 1–25. Springer, 2014.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In *TCC*, 2015.

[BSW12] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Commun. ACM*, 55(11):56–64, 2012.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (2)*, volume 8270 of *Lecture Notes in Computer Science*, pages 280–300. Springer, 2013.

[CH85] Stephen A. Cook and H. James Hoover. A depth-universal circuit. *SIAM J. Comput.*, 14(4):833–839, 1985.

[CLT15] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. *IACR Cryptology ePrint Archive*, 2015:162, 2015.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *TCC*, 2015.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[FS89] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In *CRYPTO*, pages 526–544, 1989.

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng 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*, pages 578–602, 2014.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, Mariana Raikova, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.

[GGHZ14] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Fully secure functional encryption without obfuscation, 2014.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562. IEEE Computer Society, 2005.

[GKP+12] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. Cryptology ePrint Archive, Report 2012/733, 2012.

[GLSW15] Craig Gentry, Allison B. Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. In *FOCS 2015*, 2015.

[GR07] Shafi Goldwasser and Guy N. Rothblum. On best-possible obfuscation. In *TCC*, pages 194–213, 2007.

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO*, pages 162–179, August 2012.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 545–554, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. *IACR Cryptology ePrint Archive*, 2015:29, 2015.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *FOCS*, pages 294–304. IEEE Computer Society, 2000.

[KMN$^+$14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon 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*, pages 374–383. IEEE Computer Society, 2014.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *CCS*, pages 669–684. ACM, 2013.

[KSW13] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *J. Cryptology*, 26(2):191–224, 2013.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 500–517. Springer, 2014.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *ACM CCS*, pages 463–472, 2010.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, 2005.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *STOC*, pages 475–484. ACM, 2014.

[Val76] Leslie G. Valiant. Universal circuits (preliminary report). In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing, May 3-5, 1976, Hershey, Pennsylvania, USA*, pages 196–203, 1976.

[Wat14] Brent Waters. A punctured programming approach to adaptively secure functional encryption. *IACR Cryptology ePrint Archive*, 2014:588, 2014.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pages 162–167. IEEE Computer Society, 1986.

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In *Eurocrypt*, 2015.