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. Prior 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 encryption circuits and subexponential 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.

Our main construction can be based on functional encryption schemes that support a *single function key*, and where the encryption circuit grows sub-linearly in the circuit-size of the function. We further show that sublinear succinctness in circuit-size for single-key schemes can be traded with sublinear succinctness in the number of keys (also known as the *collusion-size*) for multi-key schemes. As a consequence, we obtain a new candidate IO construction based on the functional encryption scheme of Garg, Gentry, Halevi and Zhandry (TCC'16) under their assumptions on multi-linear graded encodings. We also show that, under the Learning with Errors assumption, our techniques imply that any indistinguishability obfuscator can be converted into one where the size of obfuscated circuits is twice that of the original circuit plus an additive overhead that is polynomial in its depth, input length, and the security parameter.

Our reduction highlights the importance of 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 appears 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, and by the NEC Corporation.

‡MIT. E-mail:vinodv@csail.mit.edu. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, the NEC Corporation, a Steven and Renee Finn Career Development Chair from MIT. This work was also sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and FA8750-11-2-0225.

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. While heuristic methods of obfuscation are widely practiced, our theoretical understanding of obfuscation is still in its early stages. Rather unfortunately, the most natural and intuitively appealing notion of obfuscation, namely *virtual-black-box* (VBB) obfuscation [Had00, 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 for a long time.

This changed dramatically with recent breakthrough results. First, Garg, Gentry, Halevi, Raykova, Sahai and Waters [GGH+13c] 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 the same functionality (namely, the same truth table) 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], known IO constructions prior to this work [GGH+13c, PST14, BR14, BGK+14, GLSW15, AB15, Zim15] were 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 solid cryptographic foundations?

1.1 This Work

In this work, aiming to make progress in the above direction, we show 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 the circuits computing the corresponding functions, just the size of their output, or the number of generated functional keys.

Functional encryption, on the face of it, seems much less powerful than IO and sure enough, it has not had nearly as many applications. Seemingly, IO derives its power from the fact that it allows *anyone* to compute meaningfully with a hidden object (say, a circuit) with no additional

[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].

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 with 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", as defined by Goldwasser, Kalai, Popa, Vaikuntanthan and Zeldovich [GKP+12].

Perhaps 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.*

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 (or even exponentially, assuming pseudo-random functions in NC^1).

We also show that the requirement for sub-linear dependence on circuit size can be traded, when moving to multi-key functional encryption schemes, with sub-linear 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).

Theorem 1.2 (informal)**.** *Assuming the existence of multi-key functional encryption schemes for all circuits, where encryption time is polynomial in the input-size and circuit-size, but sub-linear in the number of released keys, there exist single-key functional encryption schemes with sub-linear dependence on circuit-size.*

As a corollary of this transformation, relying on the recent functional encryption scheme of Garg et al. [GGHZ16], we obtain a new IO candidate constructions whose security is based on the same assumptions on multi-linear graded encodings (in their subepxonential version).

Corollary 1.3 (informal)**.** *Under a sub-exponential variant of the assumptions in [GGHZ16] on multi-linear graded encodings, there exists an IO construction.*

Another corollary that 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 function's circuit size plus some overhead in the circuit's depth.

Corollary 1.4 (informal)**.** *Assuming sub-exponential hardness of the Learning with Errors problem 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 succinctness properties required in Theorem 1.2 are known based on indistinguishability obfuscation [GGH+13c] 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. We hope that the construction would eventually lead to IO from more standard assumptions, or improved assumptions on multilinear graded encodings.[3] Indeed, in the past few years, we have seen

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

[3] Below, we mention subsequent work that has already partially fulfilled this hope.

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.

Gorbunov, Vaikuntanathan and Wee [GVW15] showed how to construct predicate encryption schemes for all circuits (with a-priori bounded depth) from sub-exponential hardness of the Learning with Errors problem (LWE). 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 1.2, and is yet another frontier in achieving indistinguishability obfuscation from LWE.

We next explain the main ideas standing behind our construction.

1.2 Main Ideas

Our starting point is a natural *input extension* approach: given an obfuscator $\mathcal{O}_{n-1}$ for circuits with input length $n-1$, design an obfuscator $\mathcal{O}_n$ for circuits with input length n. Intuitively, this way we can get obfuscation for circuits with arbitrary polynomial input length — recursively apply the input extension step polynomially many times. The base case is trivial — for circuits C with a single input bit, simply define the obfuscation $\mathcal{O}_1(C)$ to be the corresponding truth table $(C(0), C(1))$.

A crucial feature of any such input extension procedure is the blowup it incurs in complexity. Indeed, a trivial input extension procedure such as bit fixing:

$$\mathcal{O}_n(C(x_1, \ldots, x_n)) := \mathcal{O}_{n-1}(C(x_1, \ldots, x_{n-1}, 0) \circ C(x_1, \ldots, x_{n-1}, 1)) \ ,$$

blows up the obfuscation size, at each step, at least by a multiplicative factor of two, since the circuit obfuscated by $\mathcal{O}_{n-1}$ blows up at every step. Accordingly, such a procedure can only be applied logarithmically-many times (indeed, it is equivalent to simply writing the truth table of the circuit). To avoid such blowup, we must ensure that the total size of circuits obfuscated in each recursive step does not outgrow the size of previous circuits (except perhaps by an additive amount).

At high-level, this work is dedicated to developing such an input-extension procedure, based on functional encryption.

From token-based obfuscation to efficient input-extension. The basic idea behind our input extension procedure is founded on the concept of *token-based obfuscation* and its connection to function-hiding functional encryption. A token-based obfuscation algorithm consists of an obfuscation algorithm $\mathsf{Tok.Obf}(C)$ that given a circuit C produces an obfuscation $\widetilde{C}$ and a secret key SK. Unlike, the standard notion of obfuscation, which would allow evaluating $\widetilde{C}$ on any

input x to learn $C(x)$, here evaluation requires a token $\widetilde{x}$ corresponding to x. The token $\widetilde{x}$ can be generated by an encoding algorithm $\mathsf{Tok.Enc}(\mathsf{SK}, x)$ using the secret key SK (for simplicity of exposition, we shall assume that $\mathsf{Tok.Enc}$ is deterministic). Security is guaranteed against any adversary that does not posses the secret key and only gets the obfuscation $\widetilde{C}$, as well as an polynomial number of encoded inputs $\widetilde{x}$ of its choice. For the notion to be non-trivial, the complexity of $\mathsf{Tok.Enc}$ is required to only depend on the input x, and not on the circuit C.

Intuitively (and for now thinking about an obfuscation as an opaque black-box), token-based obfuscation suggests a simple input-extension procedure:

$$\begin{aligned}\mathcal{O}_n(C(x_1,\ldots,x_n)) :=& \mathsf{Tok.Obf}(C(x_1,\ldots,x_n)) \\ & \mathcal{O}_{n-1}(\mathsf{Tok.Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},0) \circ \mathsf{Tok.Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},1)) \ ;\end{aligned}$$

namely, to obfuscate a circuit C with n-bit inputs $(x_0,\ldots,x_n)$, obfuscate C using the token based obfuscation, and then use the obfuscator $\mathcal{O}_{n-1}$, to obfuscate a bit-fixing variant of the token generator

$$\mathsf{Tok.Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},0) \circ \mathsf{Tok.Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},1)$$

that given $(x_1,\ldots,x_{n-1})$ generates the two encodings corresponding to to fixing x_n to either 0 or 1.

Crucially, since the complexity of $\mathsf{Tok.Enc}(\mathsf{SK}, x)$ only grows with the encoded input $x \in \{0,1\}^n$, the circuit recursively obfuscated by $\mathcal{O}_{n-1}$ is now bounded, through all steps, by a fixed polynomial $\mathrm{poly}(n)$ in the input length n. Accordingly, unwinding the recursion, the complexity of $\mathcal{O}_n$ will now be bounded by $\mathrm{poly}(|C|) + n \cdot \mathrm{poly}(n)$.

From functional encryption to token-based obfuscation. As observed in [GKP+12, BKS16], token-based obfuscation can be constructed from any *private-key* functional encryption scheme that has a succinct encryption circuit. Concretely, they show that it is possible to harness the existing message-hiding of functional encryption to also guarantee function hiding. Here a functional key FSK_C is guaranteed to hide the circuit C and can be viewed as a token-based obfuscation of C. The encryption algorithm $\mathsf{Enc}(\mathsf{SK}, \cdot)$, with the corresponding private encryption key SK, is then viewed as the token generator.

Combined with the token-based input extension procedure, this suggests a strategy for constructing obfuscation based on (private-key) functional encryption with a succinct encryption circuit. Materializing this high-level strategy requires of course a more careful examination of the security guaranteed at each and every step. Assuming all the involved primitives satisfy an *ideal* (simulation-based) security guarantee, would indeed allow implementing this strategy and eventually lead to an ideal obfuscation guarantee (known as virtual black-box security). However, ideal security is known to be impossible for either (succinct) functional encryption or obfuscation [AGVW13, BGI+12].

The hope is that starting with a weaker indistinguishability-based guarantee for functional encryption would still allow to carry through the above strategy, leading to indistinguishability obfuscation. This turns out to encounter several difficulties, which eventually lead to our requirement of public-key functional encryption (rather than private-key), as well as our sub-exponential security requirement. We next overview these challenges and the way they are dealt with.

Under the hood. 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 circuit C with input $x_1,\ldots,x_n$, generate the functional key FSK_C and add an obfuscation

$$i\mathcal{O}(\mathsf{Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},0) \circ \mathsf{Enc}(\mathsf{SK}, x_1,\ldots,x_{n-1},1))$$

of the corresponding (bit fixing) encryption circuit. While this clearly satisfies the required functionality, it is not clear how to prove security based on IO. In fact, using ideas inspired by impossibility results for obfuscation [BGI+12, GK05, BCC+14], we 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 (see Section 5.1).

Our solution relies on public-key functional encryption. In the public-key setting, it is not known how to generically obtain function-hiding like in the private-key setting; rather, we shall enforce it explicitly in our construction using similar techniques to those used in the private-key setting [BS15].

Concretely, to obfuscate C, our obfuscation will once again consist of a functional key FSK_{C^*}, this time for an augmented circuit C^*, and an obfuscation $i\mathcal{O}(\mathsf{Enc}^*)$ for an augmented (bit fixing) encryption algorithm Enc^*. The circuit C^* will consist of two (plain) 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 C. The circuit C^* expects as input, not only an input x for C, but also a key SK_b. Given those, it decrypts the corresponding ciphertext CT_b, and applies the decrypted circuit to the input x. Accordingly, the encryption algorithm Enc^*, given input $x_1, \ldots, x_{n-1}$, will generate two (public-key) encryptions of $((x_1, \ldots, x_{n-1}, 0), \mathsf{SK}_b)$ and $((x_1, \ldots, x_{n-1}, 1), \mathsf{SK}_b)$, where in the real scheme b will always be set to say 0, and may take a different value during our analysis.

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 C_0 to one corresponding to a functionally-equivalent C_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 pair of encryptions

$$\mathsf{Enc}^*(x_1, \ldots, c_{n-1}) := \mathsf{Enc}(\mathsf{PK}, ((x_1, \ldots, x_{n-1}, 0), \mathsf{SK}_0)) \circ \mathsf{Enc}(\mathsf{PK}, ((x_1, \ldots, x_{n-1}, 1), \mathsf{SK}_0))$$

of some pre-selected input $(x_1, \ldots, x_{n-1})$. In this setting, we can employ a straight forward hybrid argument to show that the functional keys $(\mathsf{FSK}_{C_0^*}, \mathsf{FSK}_{C_1^*})$ corresponding to C_0 and C_1 are indistinguishable. Indeed, relying on the symmetric-key guarantee we can change CT_1 to encrypt C_1, and then relying on the FE guarantee we change Enc^* to encrypt SK_1 instead of SK_0, indeed we know that $C_0(x) = C_1(x)$. Then, we can symmetrically switch the other cipher to encrypt C_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 Enc^* 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.

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

Crucially, for this recursion to be efficient and not result in an obfuscation of exponential size, we must require that encrypting (corresponding to token generation) is simple enough. Indeed, as long as it only depends on the underlying plaintext, throughout we will have the invariant that the functions Enc^* 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 C_i^* for which keys are derived (except for the last one which depends on the size of the function C 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 PRFs in NC^1).

In terms of security, the exponential loss due to the use of probabilistic IO accumulates recursively: roughly, the indistinguishability gap δ_i for level i of the recursion is at most $2^i \cdot \delta_{i-1}$, requiring that all underlying cryptographic primitives are roughly $2^{-\Omega(n^2)}$-secure.

Is Private-Key FE Enough? As mentioned above, we show that instantiating our transformation with an arbitrary *private-key FE* scheme may result in an insecure IO scheme.

Proposition 1.1 (informal)**.** *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 1.2 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, we 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

We mention several concurrent and independent works:

- Ananth and Jain [AJ15] also show how to construct indistinguishability obfuscation from sub-exponentially secure public-key functional encryption. The two works take a somewhat different perspective 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 n-input FE to $(n+1)$-input FE, which is analogous to our recursive step of basing $(n+1)$-bit-input IO on n-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].

- Brakerski, Komargodski, and Segev [BKS16] show how to convert any (single-input) private-key functional encryption scheme into an $O(1)$-input private-key scheme (or doubly-logarithmic-input assuming sub-exponential security), which is not known to be sufficient to go all the way to IO polynomially large inputs.

- Ananth, Jain, and Sahai [AJS15] show how IO can be bootstrapped to always have linear-size overhead. By developing new techniques, they improve on the above Corollary 1.4 in two aspects. First, they avoid the LWE assumption. Second, they avoid the polynomial dependence of the obfuscated circuit-size on the depth of the original circuit.

- Ananth, Jain, and Sahai [AJS15] also show how to transform any collusion-resistant FE into a single-key FE scheme with succinct encryption circuits.

1.4 Followup Work

We mention several subsequent works that have relied on our result, or have extended it:

- Lin, Pass, Seth and Telang [LPST16b] show a different transformation from (public-key) functional encryption to IO. While their transformation shares much of the structure of our transformation, it has different features such as better (but still sub-exponential) security loss, and admits a very elegant description in the language of succinct randomized encodings [BGL+15, CHJV15, KLW15]. Their description of the transformation from public-key FE to IO shares much of the same high-level structure as the classical Goldreich-Goldwasser-Micali transformation from a pseudorandom generator to a pseudorandom function.

- Lin, Pass, Seth and Telang [LPST16a], in another work, introduce a relaxation of IO called Exponential Indistinguishability Obfuscation (XIO) that only requires that the size of an obfuscated circuit is sub-linear in the size of its truth table. Based on our result and the learning with errors (LWE) assumption, they show that this relaxation suffices for obtaining full-fledged IO.

- Lin [Lin16] shows how to construct IO from a concrete (but complex) assumption on *constant-degree* graded encodings [GGH13b], LWE, and a low-depth polynomial-stretch pseudorandom generator. Previously, all IO constructions from graded encodings required *polynomial degree*. The result relies on our transformation as an intermediate step.

- Lin and Vaikuntanathan [LV16], extending [Lin16], show how to construct IO using *constant-degree* graded encodings and a low-depth polynomial-stretch pseudorandom generator, but rather than rely on a complex assumption on the graded encodings, they rely on a natural generalization of the symmetric external Diffie-Hellman (SXDH) assumption. Their work constructs an FE scheme, and they use our result to generically transform this into an IO scheme.

- Bitansky, Nishimaki, Passelegue and Wichs [BNPW16] show how to obtain IO from sub-exponentially-secure *private-key* functional encryption and plain public-key encryption. In fact, they show how these primitives together imply a public-key functional encryption scheme and then invoke our transformation. It is still not known whether IO can be obtain from private-key functional encryption alone (without relying on any "public-key object").

- Garg, Pandey, Srinivasan and Zhandry [GPS16, GPSZ16] show how many of the applications of IO (such as the hardness of PPAD and multiparty key exchange) can be based directly on polynomially-secure functional encryption instead. Their reduction invokes a variant of our input-extension technique, but avoids the sub-exponential security loss.
- Li and Micciancio [LM16] and Garg and Srinivasan [GS16] independently show a construction of a many-key (collusion-resistant) functional encryption starting from any polynomially secure single-key functional encryption scheme with succinct encryption circuits. Such a transformation follows from our results as IO implies a collusion-resistant functional encryption scheme by the results of [GGH+13c], except that we lose a sub-exponential security factor that comes from invoking our transformation. The results of [LM16, GS16] avoid this loss. Together with our result that transforms a polynomially secure collusion-resistant FE into a single-key FE scheme with succinct encryption circuits. This shows that both variants of FE are in fact equivalent.

2 Definitions

We review basic concepts and present the basic definitions used throughout the paper.

2.1 Standard Computational Concepts

We rely on the standard notions of Turing machines and Boolean circuits.

- We say that a (uniform) Turing machine is PPT if it is probabilistic and runs in polynomial time.
- A polynomial-size (or just polynomial-size) circuit family $\mathcal{C}$ is a sequence of circuits $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$, such that each circuit C_λ is of polynomial size $\lambda^{O(1)}$ and has $\lambda^{O(1)}$ input and output bits.
- We follow the standard habit of modeling any efficient adversary strategy as a family of polynomial-size circuits. For an adversary $\mathcal{A}$ corresponding to a family of polynomial-size circuits $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we often omit the subscript λ, when it is clear from the context.
- We say that a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if it decays faster than any polynomial.
- Two ensembles of random variables $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are said to be computationally indistinguishable, denoted by $\mathcal{X} \approx_c \mathcal{Y}$, if for all polynomial-size distinguishers $\mathcal{D}$, there exists a negligible function ν such that for all λ,

$$|\Pr[\mathcal{D}(X_\lambda) = 1] - \Pr[\mathcal{D}(Y_\lambda) = 1]| \leq \nu(\lambda).$$

  For a concrete function δ, we denote by $\mathcal{X} \approx_\delta \mathcal{Y}$ the case that the above indistinguishability gap is bounded by $\delta^{\Omega(1)}$.

2.2 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 (FE.Setup, FE.Gen, FE.Enc, 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* $\{0,1\}^n$*,*

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

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

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

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

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

   We further say that FE *is* δ*-secure, for some concrete negligible function* $\delta(\cdot)$*, if for all polynomial-size 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 three PPT algorithms $(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ 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* $\{0,1\}^n$*,*

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

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

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

where $(\mathsf{PK}, \mathsf{FSK}_f) \leftarrow \mathsf{FE.Setup}(1^\lambda, f)$*.*

We further say that FE *is* δ*-secure, for some concrete negligible function* $\delta(\cdot)$*, if for all polynomial-size 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*, independent of the function* f*, chosen during the setup phase.*
   - *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* n, s, d*, are the input-size, circuit-size, and depth of the function* f *chosen during the setup phase,* poly *is a fixed polynomial, and* $\varepsilon < 1$ *is a constant, both independent of* f*.*

Remark 2.3 (Succinctness for general schemes). In the above definition of single-key FE, we define the succinctness of the scheme in terms of the concrete function that the setup algorithm gets as input. In more general functional encryption schemes, where the setup algorithm does not get in advance the function, it would get instead a bound on the parameters n, s, d, and succinctness is defined with respect to these bounds. (The constant ε, and polynomial poly, quantifying the compression are fixed and independent of the specific bounds.)

2.3 Indistinguishability Obfuscation

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

Definition 2.4 (Indistinguishability obfuscation)**.** *A PPT algorithm* $i\mathcal{O}$ *is said to be an* indistinguishability obfuscator *for a class of circuits* $\mathcal{C}$*, if it satisfies:*

1. **Functionality:** *for any $C \in \mathcal{C}$ and security parameter λ,*

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

2. **Indistinguishability:** *for any polynomial-size distinguisher $\mathcal{D}$ there exists a negligible function $\mu(\cdot)$, such that for any two circuits $C_0, C_1 \in \mathcal{C}$ that compute the same function and are of the same size:*

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

where the probability is over the coins of $\mathcal{D}$ and $i\mathcal{O}$.

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

2.4 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 2.5 (Puncturable PRFs). *Let n, k be polynomially bounded length functions. An efficiently computable family of functions*

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

associated with an efficient (probabilistic) key sampler $\mathsf{Gen}_{\mathcal{PRF}}$, is a puncturable PRF *if there exists a polynomial-time puncturing algorithm* Punc *that takes as input a key* K*, and a point x^*, and outputs a punctured key $\mathsf{K}\{x^*\}$, so that the following conditions are satisfied:*

1. **Functionality is preserved under puncturing:** *For every $x^* \in \{0,1\}^*$,*

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

2. **Indistinguishability at punctured points:** *for any polynomial-size distinguisher $\mathcal{D}$ there exists a negligible function $\mu(\cdot)$, such that for all $\lambda \in \mathbb{N}$, and any $x^* \in \{0,1\}^*$,*

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

where $\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda), \mathsf{K}\{x^*\} = \mathsf{Punc}(\mathsf{K}, x^*)$, and $u \leftarrow \{0,1\}^\lambda$.

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

2.5 Symmetric Encryption

A symmetric encryption scheme Sym consists of a tuple of two PPT algorithms (Sym.Enc, 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 2.6 (One-Time Symmetric Encryption)**.** *A pair of PPT algorithms* (Sym.Enc, 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 polynomial-size distinguisher* $\mathcal{D}$ *there exists a negligible function* $\mu(\cdot)$*, such that for all* $\lambda \in \mathbb{N}$*, and any equal size messages* m_0, m_1*,*

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

where $\mathsf{SK} \leftarrow \{0,1\}^\lambda$*.*

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

2.6 Randomized Encodings

We rely on the notion of randomized encodings from [IK00, AIK06]. Let $c \geq 1$ be an integer constant. A (c-local, decomposable) randomized encoding for a function $f : \{0,1\}^n \to \{0,1\}^m$ is a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^\mu$ with the following properties. Let $s_{\hat{f}}$ (resp. s_f) denote the size of the circuit computing $\hat{f}$ (resp. f).

- $\hat{f}(x;r) = (\hat{f}_1(x;r), \hat{f}_2(x;r), \ldots, \hat{f}_\mu(x;r))$ where each $\hat{f}_i$ depends on at most *a single bit* of x and c bits of r. We will write

$$\hat{f}(x;r) = (\hat{f}_1(x;r_{S_1}), \hat{f}_2(x;r_{S_2}), \ldots, \hat{f}_\mu(x;r_{S_\mu}))$$

  where S_i denotes the subset of bits of r that $\hat{f}_i$ depends on.

- μ and ρ are of size $s_f \cdot \mathrm{poly}(n, \lambda)$.

- There is a polynomial time decoder algorithm that, given $\hat{f}(x;r)$, outputs $f(x)$.

- There is a PPT simulator RE.Sim that takes as input $(1^\lambda, f(x))$ and outputs $\mathsf{SimOut}_{f(x)}$ such that no polynomial-size 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 all circuits, with (fully or weakly) succinct encryption.

- A $2^{-\tilde{\lambda}^\varepsilon}$-secure one-time symmetric encryption scheme Sym,

- A $2^{-\tilde{\lambda}^{\varepsilon}}$-secure puncturable pseudo-random function family $\mathcal{PRF}$.

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

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

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

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

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

Figure 1: The recursive obfuscation procedure.

Theorem 3.1. *$i\mathcal{O}$ is an indistinguishability obfuscator for al circuits.*

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

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

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.

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 fully 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 $\mathrm{poly}(|C_i|, |\mathsf{E}_{i-1}^0|, \lambda, n)$ 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 on the circuit-size of functions). Thus, overall the 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 polynomial-size distinguisher that works on obfuscations $i\mathcal{O}(C, 1^\lambda)$ for any circuit C of size $s(\lambda)$, defined on $\{0,1\}^{n(\lambda)}$.

Our goal is to show that for all $\lambda \in \mathbb{N}$,

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

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

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

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

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

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

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

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

Proof of Proposition 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

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

Recall that

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

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

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

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

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

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

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

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

In the first and third inequalities, we simply change the symmetrically encrypted plaintext in some CT_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 that the second and fourth equations hold; 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 subexponential IO in conjunction with subexponential puncturable PRFs are sufficient in this setting, which they formalize by *probabilistic IO* notion. For the sake of completeness, we next give the full argument.

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

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

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

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

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

Note that the difference between $\mathcal{H}_{\mathbf{x}-1}$ and $\mathcal{H}_{\mathbf{x}}$ is in the circuits encrypted in $\mathsf{CT}_i^0,\mathsf{CT}_i^1$: $\mathsf{E}_{\mathbf{x}-1}$ 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

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

  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 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 subexponentially 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 exist pseudo-random functions in* NC^1*, then this also holds with* $\mathrm{poly}(2^d)$*-dependence.*

Proof. 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 n, s are the input-size and circuit-size of the function f chosen in the setup phase, $\varepsilon < 1$ is some constant, and poly is a fixed polynomial, both independent of the function f. 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 fixed $\mathrm{poly}(n, \tilde{\lambda})$ 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,

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

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

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

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

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).

To deal with the case that the circuit-size of encryption is also $\Delta(d)$-dependent on the depth of the circuit, we rely on the following bootstrapping theorem, which gives a direct reduction to the case that this dependence does not exist.

Proposition 3.2 (follows from [ABSV15])**.** *For every $\epsilon < 1$, polynomial poly, and function Δ:*

- *If there is a single-key FE scheme that for circuits of input-size n, circuit-size s, and depth d, has an encryption circuit of size*

$$s^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda, \Delta(d)) \quad ,$$

- *If there is a single-key FE scheme that for circuits of input-size n, circuit-size s, and depth d, has an encryption circuit of size*

$$s^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda, \Delta(\mathsf{dep}_{\mathsf{prg}}(\lambda, t))) \quad ,$$

  where $\mathsf{dep}_{\mathsf{prg}}(\lambda, t)$ is the depth of a pseudo-random generator expanding λ bits to t bits, $t = s \cdot \mathrm{poly}(n, \lambda)$, and poly is some fixed polynomial.

In [ABSV15] a different version of this theorem is proven for the multi-key setting, which in particular, relies on the ability to compute pseudo-random functions in low depth, and not just pseudo-random generators. For completeness, we sketch the proof of Proposition 3.2 below.

Before that, let us show how it concludes the proof of Theorem 3.3. Indeed, a pseudo-random generator stretching $\tilde{\lambda}$ to $t = t(n, s, \tilde{\lambda})$ bits can always be computed in depth $\log(t) \cdot \mathrm{poly}(\lambda)$, by applying (say the GGM) PRF for each output bit. Since $t \leq s \cdot \mathrm{poly}(n, \tilde{\lambda})$, this depth is bounded by $\mathrm{poly}(n, \tilde{\lambda})$ for some fixed poly. This gives the required succinctness in the case of $\mathrm{poly}(d)$-dependence. Furthermore, given PRFs in NC^1, the depth of computing the pseudo-random generator reduces to $\log \log t$, in which case $\mathrm{poly}(2^d) = o(\lambda)$.

Proof of Proposition 3.2. Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be a single-key scheme with depth dependence as described in the proposition. Let $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ be a one time symmetric-key encryption scheme (Definition 2.6), where for plaintexts of size t, ciphertexts are also of size t, and decryption is done by a circuit of depth $d_{\mathsf{dec}} = \mathsf{dep}_{\mathsf{prg}}(\lambda, t)$ and size $s_{\mathsf{dec}} = O(t \cdot d_{\mathsf{dec}})$. (Such a one-time encryption scheme can be constructed by simply applying the the generator PRG, using the one-time key as the seed.)

We construct a new scheme $\mathsf{FE}^\star = (\mathsf{FE.Setup}^\star, \mathsf{FE.Enc}^\star, \mathsf{FE.Dec}^\star)$ that works as follows.

- $\mathsf{FE.Setup}^\star(1^\lambda, f)$ runs $\mathsf{FE.Setup}(1^\lambda, f^\star)$ to generate a key pair $(\mathsf{PK}, \mathsf{FSK}_{f^\star})$, where $f^\star$ is generated as follows:

1. sample a symmetric one-time encryption $\mathsf{CT} \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, 0^t)$, where t is defined below.
2. construct the circuit $f^\star$ which, on input a tuple $(b, x, s, \mathsf{SK}) \in \{0,1\} \times \{0,1\}^n \times \{0,1\}^\lambda \times \{0,1\}^\lambda$ work as follows:
   - If $b = 0$, output $e \leftarrow \hat{f}(x; r)$, where $r = \mathsf{PRG}(s)$. We let $t = |\hat{f}(x; r)|$.
   - If $b = 1$, output $e \leftarrow \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT})$.

- $\mathsf{FE.Enc}^\star(\mathsf{PK}, x)$ chooses a seed key $s \leftarrow \{0,1\}^\lambda$, and outputs $\mathsf{FCT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}; (0, x, s, \perp))$.
- $\mathsf{FE.Dec}^\star(\mathsf{FSK}_f, \mathsf{FCT})$ computes $e \leftarrow \mathsf{FE.Dec}(\mathsf{FSK}_{f^\star}, \mathsf{FCT})$ and runs the decoder of the randomized encoding on input e 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. During the encryption of an input x, the encryption algorithm of $\mathsf{FE}^\star$ is invoked on an input of size $O(n + \lambda)$. The circuit-size $s_{f^\star}$ of $f^\star$ is bounded by the time required to derive randomness and compute the randomized encoding of f, as well as decryption time:

$$s_{f^\star} \leq t \cdot \mathsf{dep}_{\mathsf{prg}}(\lambda, t) + s \cdot \mathrm{poly}(n, \lambda) + t \cdot \mathsf{dep}_{\mathsf{prg}}(\lambda, t) \ ,$$

where in our case $t \leq s \cdot \mathrm{poly}(n, \lambda)$ is bounded by the size of the randomized encoding time. Accordingly, and since the depth required to compute a randomized encoding is dominated by the depth of computing a linear stretch pseudo-random generator, the depth $d_{f^\star}$ of $f^\star$ is bounded by

$$d_{f^\star} \leq O(\mathsf{dep}_{\mathsf{prg}}(\lambda, t)) \ .$$

By the succinctness of FE, and restricting attention to the case that $d_{f^\star} \leq \mathrm{poly}(n, \lambda)$, the size of the encryption circuit in $\mathsf{FE}^\star$ is

$$s_{f^\star}^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda, \Delta(d_{f^\star})) = s^{1-\epsilon} \cdot \mathrm{poly}(n, \lambda, \Delta(\mathsf{dep}_{\mathsf{prg}}(\lambda, s \cdot \mathrm{poly}(n, \lambda)))) \ .$$

Security. We now sketch the proof of security, which proceeds by a sequence of hybrids. Given that we are in the public-kye setting, it is sufficient to consider the case when the adversary submits a single challenge pair (x_0, x_1).

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

$\mathcal{H}_1$**:** The challenger replaces CT with a symmetric encryption of the bits of $\widehat{f}(x_0; r)$ in the functional key for f, where $r = \mathsf{PRG}(s)$ 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 $(0, x_0, s, \perp)$. This hybrid is computationally indistinguishable from $\mathcal{H}_1$ by the security of the underlying functional encryption scheme.

$\mathcal{H}_3$**:** The challenger replaces the encryption CT in the function key with $\mathsf{Sym.Enc}(\mathsf{SK}, \widehat{f}(x_0; r))$ for a uniform r. $\mathcal{H}_3$ is computationally indistinguishable from $\mathcal{H}_2$ based on the security of PRG.

$\mathcal{H}_4$**:** 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. □

This completes the proof of Theorem 3.3.

□

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 3.3 (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 3.4. *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 Collusion-Succinct 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 [GGHZ16], we obtain an IO construction based on a subexponential variant of the assumptions in [GGHZ16] 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 subexponential 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 encryption circuit 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 encryption circuit 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 2.6). 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
$$\mathsf{FCT} \leftarrow \mathsf{FE.Enc}(\mathsf{PK}; (0, x, \mathsf{K}, \perp)) \ .$$
- $\mathsf{sFE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$ 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

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

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.

$\mathcal{H}_3$**:** The challenger replaces the encryption CT_i in the function key 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 [GGHZ16] satisfies collusion-succinctness and thus we obtain the following corollary

Corollary 4.1. *Under a subexponential variant of the assumptions in [GGHZ16] on multi-linear graded encodings, there exists an IO construction.*

5 On the Possibility of Using 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 of some extra information — a fixed key for a symmetric encryption scheme and a bit). Using techniques from the literature of unobfuscatable functions [BGI+12, GK05, BCC+14], we show how to 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 and break the resulting obfuscation scheme.

We next define symmetric-key FE and unobfuscatable functions, and then prove the above proposition. For simplicity, we restrict attention to fully-succinct schemes.

Symmetric-key FE. The definition of symmetric-key FE naturally restricts the public-key Definition 2.2. Concretely, there is one master 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* $\{0,1\}^n$*,*

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

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

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

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

   (a) *The adversary submits the challenge message-pair* $m_0, m_1 \in \{0,1\}^n$ *to the challenger.*

   (b) *The challenger executes* $\mathsf{FE.Setup}(1^\lambda)$ *to obtain* 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 polynomial-size adversaries the above indistinguishability gap* $\mu(\lambda)$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Auxiliary-input unobfuscatable functions. We now define a variant of auxiliary-input unobfuscatable functions and prove their existence based on indistinguishability obfuscation.

Definition 5.2 (CPA-secure auxiliary-input unobfuscatable functions, variant of [GK05, BCC+14])**.** *A CPA-secure auxiliary-input unobfuscatable function (UOF) scheme consists of four algorithms* $\mathsf{UOF} = (\mathsf{UOF.Gen}, \mathsf{UOF.Eval}, \mathsf{UOF.Dec}, \mathsf{UOF.Ext})$*, with the following syntax:*

- $Z \leftarrow \mathsf{UOF.Gen}(S; K, 1^\ell)$*: a PPT algorithm that given an input* S*, a secret key* $K \in \{0,1\}^\lambda$*, and a parameter* $\ell \in \mathbb{N}$*, samples an auxiliary input* Z*.*
- $y \leftarrow \mathsf{UOF.Eval}(x; K)$*: a PPT algorithm that given an input* $x \in \{0,1\}^{n(\lambda)}$ *and secret key* K*, samples some encoding* y *of* x*.*
- $\widetilde{x} \leftarrow \mathsf{UOF.Dec}(y; K)$*: a polynomial-time deterministic algorithm that given an encoding* y *and secret key* K*, decodes it to a value* $\widetilde{x} \in \{0,1\}^{n(\lambda)}$*.*
- $\widetilde{S} \leftarrow \mathsf{UOF.Ext}(C; Z)$*: a polynomial-time deterministic algorithm given a circuit* C *and auxiliary input* Z*, outputs some* $\widetilde{S}$*.*

We require that the scheme satisfies the following requirements.

1. **Non-Black-box learning from bounded circuits:** *For all* $\lambda \in \mathbb{N}$*, secret key* $K \in \{0,1\}^\lambda$*, secret* $S \in \{0,1\}^*$*, parameter* $\ell \leq 2^{n(\lambda)-1}$*, and* Z *in the support of* $\mathsf{UOF.Gen}(S; K, 1^\ell)$*, let* C *be a circuit of size at most* ℓ *such that for any* $x \in \{0,1\}^{n(\lambda)}$ *there exists* y *in the support of* $\mathsf{UOF.Eval}(x; K)$ *for which:*

$$\mathsf{UOF.Dec}(C(x); K) = \mathsf{UOF.Dec}(y; K) \ .$$

   Then the extractor can extract from C *the secret* S*:*

$$\mathsf{UOF.Ext}(C; Z) = S \ .$$

2. **Black-box unlearnability:** *For any polynomial-size oracle-aided adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\lambda)$*, such that for all* $\lambda \in \mathbb{N}$*, any* $S_0, S_1 \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, and* $\ell \leq 2^{n(\lambda)-1}$*,*

$$\left| \Pr_{\substack{b \leftarrow \{0,1\}, K \leftarrow \{0,1\}^\lambda \\ Z \leftarrow \mathsf{UOF.Gen}(S_b; K, 1^\ell)}} \left[\mathcal{A}^{\mathsf{UOF.Eval}(\cdot; K)}(Z, S_0, S_1) = b \right] - \frac{1}{2} \right| \leq \mu(\lambda) + 2^{-\ell} \ .$$

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

3. **Chosen plaintext attack (CPA) security:** *For any polynomial-size oracle-aided adversary* $\mathcal{A}$*, there exists a negligible function* $\mu(\lambda)$*, such that for all* $\lambda \in \mathbb{N}$*, any* $S \in \{0,1\}^{\mathrm{poly}(\lambda)}$*, any* $\ell \leq 2^{n(\lambda)-1}$*, and any* $x_0, x_1 \in \{0,1\}^{n(\lambda)}$

$$\left| \Pr_{\substack{b \leftarrow \{0,1\}, K \leftarrow \{0,1\}^\lambda \\ Z \leftarrow \mathsf{UOF.Gen}(S; K, 1^\ell) \\ y \leftarrow \mathsf{UOF.Eval}(x_b; K)}} \left[\mathcal{A}^{\mathsf{UOF.Eval}(\cdot; K)}(y, Z, S, x_0, x_1) = b \right] - \frac{1}{2} \right| \leq \mu(\lambda) + 2^{-\ell} \ .$$

   We say that the scheme satisfies δ*-CPA-security, for some concrete negligible function* $\delta(\cdot)$*, if for all polynomial-size adversaries the above indistinguishability gap* $\mu(\lambda)$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Proposition 5.2. *Assuming indistinguishability obfuscation and one-way functions, both* δ*-secure, there exists an unobfuscatable function scheme with* δ*-unlearnability and* δ*-CPA-security.*

Proof sketch. Roughly speaking, the idea behind the theorem builds on the same ideas as previous constructions of auxiliary-input unobfuscatable functions (the main difference being that here we also need to show CPA-security).

To construct the unobfuscatable function scheme, we rely on the following ingredients:

- A δ-CPA-secure symmetric encryption scheme $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$.
- A δ-secure pseudo-random function family $\mathcal{PRF}$ mapping $\left\{\{0,1\}^{n(\lambda)}\right\}_{\lambda\in\mathbb{N}}$ to $\{0,1\}$.
- A δ-secure indistinguishability obfuscator $i\mathcal{O}$.

The scheme is constructed as follows:

- A random key $K = (\mathsf{K}, \mathsf{SK}) \leftarrow \{0,1\}^\lambda$ consists of a random key for a PRF $\mathsf{K} \leftarrow \{0,1\}^{\lambda/2}$ and a random key $\mathsf{SK} \leftarrow \{0,1\}^{\lambda/2}$ for symmetric encryption.
- $\mathsf{UOF.Gen}(S; K, 1^\ell)$:
  - For all $i \in [2\ell] \subseteq \{0,1\}^{n(\lambda)}$, for some canonical embedding of $[2\ell]$ in $\{0,1\}^{n(\lambda)}$,[4] compute $Y_i = \mathsf{PRF}_\mathsf{K}(i)$.
  - Outputs an obfuscation $Z \leftarrow i\mathcal{O}(\mathsf{WD}[Y_1, \ldots, Y_{2\ell}, \mathsf{SK}, S], 1^\lambda)$ of the witness decryption circuit WD defined in Figure 3.
- $\mathsf{UOF.Eval}(x; K)$:
  - Compute $Y_x := \mathsf{PRF}_\mathsf{K}(x)$.
  - Output $y \leftarrow \mathsf{Sym.Enc}(Y_x; \mathsf{SK})$.
- $\mathsf{UOF.Dec}(y; K)$:
  - Output $\widetilde{x} \leftarrow \mathsf{Sym.Dec}(y; \mathsf{SK})$.
- $\widetilde{S} \leftarrow \mathsf{UOF.Ext}(C; Z)$:
  - Output $\widetilde{S} = Z(C)$.

Hardwired: bits $Y_1, \ldots, Y_{2\ell} \in \{0,1\}$, a symmetric decryption key $\mathsf{SK} \in \{0,1\}^{\lambda/2}$, and a secret S.

Input: a circuit $C\colon \{0,1\}^{n(\lambda)} \to \{0,1\}^{\mathrm{poly}(\lambda)}$ of size $|C| \leq \ell$.

1. If $\mathsf{Sym.Dec}(C(i)) = Y_i$ for all $i \in [2\ell]$, output S.
2. Else, return $\perp$.

Figure 3: The circuit $\mathsf{WD}[Y_1, \ldots, Y_{2\ell}, \mathsf{SK}, S]$.

Non-black-box learnability of the above scheme follows directly from it definition and functionality of the underlying primitives. Indeed, for any circuit C of size at most ℓ, assume that for all $x \in \{0,1\}^{n(\lambda)}$ there exists y in the support of $\mathsf{UOF.Eval}(x; K)$ for which:

$$\mathsf{UOF.Dec}(C(x); K) = \mathsf{UOF.Dec}(y; K) \ .$$

[4] Recall that $2\ell \leq 2^{n(\lambda)}$.

Then for all $i \in [2\ell]$

$$\mathsf{Sym.Dec}(C(i), k) = \mathsf{Sym.Dec}(\mathsf{Sym.Enc}(\mathsf{PRF}_{\mathsf{K}}(i); \mathsf{SK}); \mathsf{SK}) = Y_i \ .$$

In this case

$$\mathsf{UOF.Ext}(C, Z) = Z(C) = \mathsf{WD}(C) = S \ .$$

We now prove black-box unlearnability and CPA security. For this purpose, we first consider a hybrid version of the above scheme where $\mathsf{PRF}_{\mathsf{K}}$ is replaced with a truly random function (the scheme is no inefficient, but this is a mental experiment confined to the analysis). In particular, the bits $Y_1, \ldots, Y_{2\ell}$ are now truly random. We claim that the adversary's advantage in this hybrid world in either the black-box learning or CPA game changes at most by $\delta^{\Omega(1)}$. This follows directly from the pseudo-randomness of $\mathcal{PRF}$. Indeed, the real world and the hybrid world can be efficiently simulated given oracle access to $\mathsf{PRF}_{\mathsf{K}}$, in the real world, or the random function in the hybrid world, respectively.

We now observe that in this hybrid world, the probability that there exists a circuit C of size at most ℓ that information theoretically encodes the random string $Y_1, \ldots, Y_{2\ell}$ is at most $2^{-\ell}$. In this case, the circuit $\mathsf{WD}[Y_1, \ldots, Y_{2\ell}, \mathsf{SK}, S]$ always returns $\perp$. In particular, we can switch to yet another hybrid world, where the obfuscation of $\mathsf{WD}[Y_1, \ldots, Y_{2\ell}, \mathsf{SK}, S]$ is replaced with an obfuscation of a circuit $[Y_1, \ldots, Y_{2\ell}, \perp, \perp]$ that is independent of S and SK, and this, by the IO guarantee, changes the adversary's advantage by at most $\delta^{\Omega(1)} + 2^{-\ell}$.

In this hybrid world, black-box unlearnability is follows from the fact that Z and the oracle $\mathsf{UOF.Eval}$ are completely independent of the secret S. CPA-security follows directly from the CPA-security of the underlying symmetric encryption. Indeed, the oracle $\mathsf{UOF.Eval}$ can be efficiently simulated, given an oracle to $\mathsf{Sym.Enc}(\cdot; K)$.

This completes the proof sketch.

□

We are now ready to prove Proposition 5.1.

Proof sketch of Proposition 5.1. Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Gen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be a δ-selectively-secure symmetric-key functional encryption scheme. Let $\mathsf{UOF} = (\mathsf{UOF.Gen}, \mathsf{UOF.Eval}, \mathsf{UOF.Dec}, \mathsf{UOF.Ext})$ be an unobfuscatable function scheme with δ-unlearnability and δ-CPA-security.

We now define a new FE scheme FE^*. In what follows, we shall assume that encrypted messages consist of two parts $(x, x') \in \{0,1\}^{n(\lambda)} \times \{0,1\}^{\mathrm{poly}(\lambda)}$. Later, when considering the transformation from Section 3, x will represent the main part of the input and x' will represent the extra information (SK, b), which are both encrypted together under the FE scheme.

- A master key $\mathsf{MSK}^* = (\mathsf{MSK}, K)$ consists of a master secret-key MSK for FE, and secret key $K \leftarrow \{0,1\}^\lambda$ for UOF.
- $(\mathsf{FCT}, y) \leftarrow \mathsf{FE.Enc}^*(\mathsf{MSK}^*, x, x')$: given MSK^* and $(x, x') \in \{0,1\}^{n(\lambda)} \times \{0,1\}^{\mathrm{poly}(\lambda)}$, returns a ciphertext $\mathsf{FCT} \leftarrow \mathsf{FE.Enc}(\mathsf{MSK}, (x, x'))$, and an encoding $y \leftarrow \mathsf{UOF.Eval}(x; K)$.
- $(\mathsf{FSK}_f, Z) \leftarrow \mathsf{FE.Gen}^*(\mathsf{MSK}^*, f)$: given MSK^* and f, returns the functional key $\mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f)$ and auxiliary input $Z \leftarrow \mathsf{UOF.Gen}(\mathsf{MSK}; K, 1^\ell)$. Here $\ell = \ell(\lambda)$ is a polynomial that depends on the original FE and the fixed $\mathrm{poly}(\lambda, n(\lambda))$ bound on the running time of $\mathsf{UOF.Eval}$, which we specify below as part of the analysis.
- $\mathsf{FE.Dec}^*(\mathsf{FSK}_f, (\mathsf{FCT}, y)) := \mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$.

Correctness and succinctness. First, note that FE^* recovers the required functionality of a symmetric-key scheme. Indeed, ciphertexts and functional keys in the new scheme in particular include corresponding ciphertexts and functional keys with respect to the previous scheme.

Also, FE^* is succinct. Indeed, the new encryption algorithm consists of the previous encryption algorithm and an invocation $\mathsf{UOF.Eval}$ that is computed in fixed polynomial time in $n(\lambda), \lambda$.

Security. We now argue that the scheme FE^* is a secure symmetric-key FE just as the original scheme FE. For this purpose, we shall consider two hybrid schemes $\mathsf{FE}_1^*, \mathsf{FE}_2^*$. We will show that these are indistinguishable from FE^*, and that FE_2^* is as secure as the original scheme FE.

Hybrid scheme FE_1^*: Here when generating functions keys, $(\mathsf{FSK}_f, Z) \leftarrow \mathsf{FE.Gen}_1^*(\mathsf{MSK}^*, f)$ does not generate an auxiliary input $Z \leftarrow \mathsf{UOF.Gen}(\mathsf{MSK}; K, 1^\ell)$ as $\mathsf{FE.Gen}^*(\mathsf{MSK}^*, f)$ would. Instead, it invokes $\mathsf{UOF.Gen}(\mathsf{MSK}_1; K, 1^\ell)$ for some random MSK_1 that is independent of the master secret key MSK. We claim that

$$\mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}^*}(1^\lambda, b) \approx_\delta \mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}_1^*}(1^\lambda, b) \ .$$

This follows directly from the δ-unlearnability of UOF. Indeed, any adversary $\mathcal{A}$ that distinguishes between the above two experiments can violate unlearnability — the experiments can be perfectly simulated by emulating FE^* (respectively, FE^*) with an oracle to $\mathsf{UOF.Eval}(\cdot; K)$.

Hybrid scheme FE_2^*: Here when encrypting, $(\mathsf{FCT}, y) \leftarrow \mathsf{FE.Enc}_2^*(\mathsf{MSK}^*, (x, x'))$ does not generate an encoding of x, $y \leftarrow \mathsf{UOF.Eval}(x; K, 1^\ell)$, as $\mathsf{FE.Enc}_1^*(\mathsf{MSK}^*, (x, x'))$ (or $\mathsf{FE.Enc}^*(\mathsf{MSK}^*, (x, x'))$) would. Instead, it invokes $y \leftarrow \mathsf{UOF.Eval}(x_2; K, 1^\ell)$ for some random $x_2 \in \{0,1\}^{n(\lambda)}$ independent of x. We claim that

$$\mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}_1^*}(1^\lambda, b) \approx_\delta \mathsf{Expt}_{\mathcal{A}}^{\mathsf{FE}_2^*}(1^\lambda, b) \ .$$

This follows from the δ-CPA-security of UOF and a standard hybrid argument. Indeed, any adversary $\mathcal{A}$ that distinguishes between the above two experiments can violate CPA security in a hybrid experiment where $\mathsf{FE.Enc}_1^*(\mathsf{MSK}^*, (x, x'))$ for the first i ciphertexts and $\mathsf{FE.Enc}_2^*(\mathsf{MSK}^*, (x, x'))$ is invoked for the rest — these experiments can be perfectly simulated by emulating FE_1^* (respectively, FE_2^*) with an oracle to $\mathsf{UOF.Eval}(\cdot; K)$.

Hybrid scheme FE_2^* is secure: We argue that FE_2^* is as secure as the original scheme FE. Indeed, the view of the adversary in FE_2^* can be completely simulated from its view in FE by simulating UOF. Indeed, in FE_2^*, all algorithms in UOF are invoked on inputs that are independent of FE.

Insecurity of FE to IO transformation when instantiated with FE^*: Consider instantiating the transformation from FE to IO, given in Section 3, with FE^*, and let us assume that whenever $\mathsf{FE.Enc}^*$ is invoked to encrypt $(\mathbf{x}_{i-1}x_i, \mathsf{SK}_i, 0)$, we treat $(\mathsf{SK}_i, 0)$ as the extra information x' and $\mathbf{x}_{i-1}x_i$ as the main input x.

Recall that, after applying the transformation, the resulting obfuscation $\widetilde{\mathsf{E}}_n$ consists of functional keys $(\mathsf{FSK}_1^*, \ldots, \mathsf{FSK}_n^*)$ and a pair of initial encryptions, denoted by $\mathsf{E}_1^0 = (\mathsf{FCT}_1^0, \mathsf{FCT}_1^1)$. In our case, $\mathsf{FSK}_i^* = (\mathsf{FSK}_i, Z)$ where FSK_i is a functional key under the original scheme FE. We can then construct from the keys $(\mathsf{FSK}_1, \ldots, \mathsf{FSK}_{n-1})$ and initial encryptions E_1^0 a circuit C that given as input $x \in \{0,1\}^n$ outputs y in the support of $\mathsf{UOF.Eval}(x, K)$. In more detail, we can construct the evaluation circuit $r\mathcal{O}.\mathsf{Eval}(n-1, \widetilde{\mathsf{E}}_{n-1}, \cdot)$, which given $\mathbf{x}_{n-1} \in \{0,1\}^{n-1}$ outputs a pair of encryptions

$$\begin{aligned}&\{\mathsf{FCT}_n^{*x_n} \in \mathsf{FE.Enc}^*(\mathsf{MSK}^*, (\mathbf{x}_{n-1}x_n, \mathsf{SK}_n, 0))\}_{x_n \in \{0,1\}} = \\ &\{\mathsf{FCT}_n^{x_n} \in \mathsf{FE.Enc}(\mathsf{MSK}, (\mathbf{x}_{n-1}x_n, \mathsf{SK}_n, 0)), y_n^{x_n} \in \mathsf{UOF.Eval}(\mathbf{x}_{n-1}x_n, K)\}_{x_n \in \{0,1\}} \ .\end{aligned}$$

From this, we can easily construct the desired circuit C (by outputting the relevant $y \in \mathsf{UOF.Eval}(x, K)$ value). We further note that since functional decryption in FE^* invokes functional decryption in the original scheme FE (ignoring the extra information embedded in the ciphertexts by FE^*), the circuit C indeed only depends on $(\mathsf{FSK}_1, \ldots, \mathsf{FSK}_{n-1})$ and initial encryptions E_1^0 and not on the auxiliary input Z in each $\mathsf{FSK}_i^* = (Z, \mathsf{FSK}_i)$. Furthermore, note that the size of each FSK_i only depends on the security parameter λ, n, and the circuit size of $\mathsf{UOF.Eval}$. Accordingly, the size of the circuit C can be bounded by a fixed polynomial in λ and $n(\lambda)$, which dictates our choice of the bound ℓ.

It is now evident that given the circuit C and auxiliary input Z in the last key FSK^*, we can use $\mathsf{UOF.Ext}$ to extract the master secret key MSK_n and thus also learn SK_n, which completely reveals the obfuscated circuit (encrypted under SK in FSK_n). □

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

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

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

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

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

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

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

Proposition 5.3. *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 $\left\{(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i), \mathsf{SK}_i^1, 1)\right\}$ and hardwire the encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i)$. As in the original analysis, functionality is preserved, this time by the correctness of the puncturable symmetric-key FE. Then, when replacing the encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^0, 0), (\mathbf{x}, x_i)$ with and encryption of $(\mathbf{x}, x_i), \mathsf{SK}_i^1, 1), (\mathbf{x}, x_i)$, we rely on semantic security at punctured points. □

Acknowledgements

We thank Daniel Wichs for pointing out an error in a previous version of Section 5.1.

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.

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Achieving compactness generically: Indistinguishability obfuscation from non-compact functional encryption. *IACR Cryptology ePrint Archive*, 2015:730, 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.

[BGL+15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In *Symposium on Theory of Computing, STOC 2015*, 2015.

[BKS16] Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 852–880, 2016.

[BNPW16] Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From cryptomania to obfustopia through secret-key functional encryption. *IACR Cryptology ePrint Archive*, 2016:558, 2016.

[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.

[CHJV15] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Succinct garbling and indistinguishability obfuscation for RAM programs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 429–437, 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.

[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.

[GGH13b] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2013.

[GGH+13c] 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.

[GGHZ16] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 480–511, 2016.

[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.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *CRYPTO*, 2016.

[GPSZ16] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. *IACR Cryptology ePrint Archive*, 2016:102, 2016.

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

[GS16] Sanjam Garg and Akshayaram Srinivasan. Single-key to multi-key functional encryption with polynomial loss. Cryptology ePrint Archive, Report 2016/524, 2016. http://eprint.iacr.org/2016/524.

[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.

[Had00] Satoshi Hada. Zero-knowledge and code obfuscation. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, pages 443–457, 2000.

[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.

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 419–428, 2015.

[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.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, pages 28–57, 2016.

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. Cryptology ePrint Archive, Report 2016/561, 2016. http://eprint.iacr.org/2016/561.

[LPST16a] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In *Public-Key Cryptography - PKC 2016 - 19th IACR International Conference on Practice and Theory in Public-Key Cryptography, Taipei, Taiwan, March 6-9, 2016, Proceedings, Part II*, pages 447–462, 2016.

[LPST16b] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Output-compressing randomized encodings and applications. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, pages 96–124, 2016.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. Cryptology ePrint Archive, Report 2016/795, 2016. http://eprint.iacr.org/2016/795.

[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.

[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.