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 previously 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 functional 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. 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.

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

†Tel Aviv University. E-mail:nirbitan@tau.ac.il. Member of the Check Point Institute of Information Security. Supported by the Alon Young Faculty Fellowship and by Len Blavatnik and the Blavatnik Family foundation. Part of this research was done while at MIT and 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 used in practice, 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 [Had00, BGI+12, GK05, BCC+14]. Furthermore, except for very restricted function classes (see, for example, [Can97, LPS04, Wee05, HMS07, HRSV11, CRV10]), no candidate construction with any form of meaningful security was known for a long time.

This changed dramatically when Garg, Gentry, Halevi, Raykova, Sahai and Waters [GGH+13b] demonstrated a candidate obfuscation algorithm for all circuits, and conjectured that it satisfies an apparently weak notion of *indistinguishability obfuscation* (IO) [BGI+12, GR07], requiring only that the obfuscations of any two circuits of the same size and 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+13b, 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], it is possible to produce functional keys FSK_f for functions f (represented as circuits throughout this paper). Given an encryption of an input x, computed using a 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 help. In contrast, FE does allow us to encrypt circuits[2] but to evaluate the circuit on an input,

[1] Strictly speaking, we need the assumption that IO exists, plus a very mild (and minimal) complexity-theoretic assumption that NP $\not\subseteq$ ioBPP [KMN+14].

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

one needs a functional 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, Vaikuntanathan and Zeldovich [GKP+13].

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 functional 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 a sub-linear dependence on the number of derived keys. Informally, in such multi-key schemes, it is guaranteed that an adversary in the possession of a set of functional keys, learns nothing on the encrypted message beyond the combination of function outputs corresponding to the set of keys.

In fact, we show a generic bootstrapping theorem that captures both the transformation from the multi-key setting to the single-key setting, and the removal of depth dependence in single-key schemes.

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, or single-key functional encryption schemes with sub-linear dependence on circuit-size and polynomial dependence on circuit-depth, there exist single-key functional encryption schemes with sub-linear dependence on circuit-size (and no further dependence on circuit-depth).*

This transformation, in particular, allows obtaining new IO candidates from exsiting multi-key functional encryption schemes such as the one by Garg et al. [GGHZ16] (in its subepxonentially-hard version).

Another corollary that follows from our techniques and 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 circuit-depth.

Corollary 1.3 (informal)**.** *Assuming sub-exponential hardness of the Learning with Errors problem and IO, there exists IO such that an obfuscation of any circuit C of depth d and input length n is of size $2|C| + \mathrm{poly}(n, d, \lambda)$.*

Interpretation. Functional encryption schemes satisfying the succinctness properties required in Theorem 1.2 were previously known based on indistinguishability obfuscation [GGH+13b] 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

[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+13] 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, AIKW13, GKP+13]. 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:

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

that first creates an $(n-1)$-bit input circuit with a doubly-long output (by taking two copies of the original circuit and fixing the last bit x_n to either 0 or 1) blows up the obfuscation size, at each step, by at least a multiplicative factor of two. 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 possess the secret key and only gets the obfuscation $\widetilde{C}$, as well as a 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 $(x_1,\ldots,x_n)$, 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 two encodings corresponding 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 $\mathcal{O}_{n-1}$ (and then $\mathcal{O}_{n-2}$ and so on) is now bounded, through all steps, by a fixed polynomial $\mathrm{poly}(n)$ in the input length n. 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+13, BS15], token-based obfuscation can be constructed from any *symmetric-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 (symmetric-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 certain difficulties, which eventually lead to our requirement of public-key functional encryption (rather than symmetric-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 (symmetric-key) function-hiding functional encryption scheme, to obfuscate any circuit C with input $\mathbf{x} = x_1 \ldots x_n$, 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. This clearly satisfies the required functionality, but it is not clear how to prove security based on IO. While the function-hiding guarantee of symmetric-key FE holds in the presence of an encryption *oracle*, it may not hold when the adversary is presented with an actual circuit implementing this oracle. In particular, an indistinguishability obfuscation of the encryption circuit, which includes the secret encryption key, may potentially leak the secret key. (As a matter of fact, it may even be that the encryption function is *unobfuscatable* [BGI+12] in the sense that any circuit implementation thereof would leak the secret key).

To overcome the above difficulty, we would like to obtain FE with a more robust function hiding guarantee that holds even in the presence of an indistinguishability obfuscation of the encryption algorithm. Perhaps the first thought that comes to mind is to use *public-key* function-hiding FE, in which case the encryption circuit can be published without compromising security. However, it is easy to see that public-key function-hiding FE is too strong in the sense that it directly implies IO, which is the very thing we are trying to construct.[4]

Instead, we define a new notion of *puncturable functional encryption* that lies somewhere between the notions of symmetric-key and public-key for function-hiding FE. In particular, this notion may not provide full-fledged function hiding in the presence of the encryption key, but does satisfy certain weaker properties. Once we define this notion, we show how to obtain it from plain (non-function-hiding) public-key FE, and show that the guaranteed security properties are already enough to prove the security of the IO input-extension procedure described above.

Puncturable FE. Roughly speaking, the notion of puncturable FE (PFE) satisfies two properties: *key indistinguishability* and *puncturing*. The first property asserts that for any two functions f_0 and f_1, of the same size and functionality, it is possible to generate a fake functional key $\mathsf{FSK}^\star = \mathsf{FSK}^\star_{f_0,f_1}$ together with two fake encryption keys $\mathsf{EK}^\star_0, \mathsf{EK}^\star_1$ so that a real encryption key EK and a functional key FSK_{f_b} corresponding to f_b are indistinguishable from the fake functional key $\mathsf{FSK}^\star$ together with the corresponding fake encryption key $\mathsf{EK}^\star_b$:

$$\mathsf{EK}, \mathsf{FSK}_{f_0} \approx_c \mathsf{EK}^\star_0, \mathsf{FSK}^\star, \qquad \mathsf{EK}^\star_1, \mathsf{FSK}^\star \approx_c \mathsf{EK}, \mathsf{FSK}_{f_1} \ .$$

The gap between such an FE and a function hiding FE is that given the fake functional key $\mathsf{FSK}^\star$ it may still be possible to distinguish $\mathsf{EK}^\star_0$ from $\mathsf{EK}^\star_1$ (indeed this is the case in our scheme).

The second property, called puncturing, is meant to bridge this gap. Our notion of puncturing is inspired by that of *puncturable pseudorandom functions* [BW13, KPTZ13, BGI14] and roughly says the following. First, for any message x, encryptions under the two fake keys are indistinguishable

$$\mathsf{Enc}(\mathsf{EK}^\star_0, x), \mathsf{FSK}^\star \approx_c \mathsf{Enc}(\mathsf{EK}^\star_1, x), \mathsf{FSK}^\star \ .$$

Moreover, there is an efficient way to generate a so called punctured version $\mathsf{EK}^\star_0\{x\}, \mathsf{EK}^\star_1\{x\}$ of the keys $\mathsf{EK}^\star_0, \mathsf{EK}^\star_1$, so that the above indistinguishability holds even in the presence of *both* punctured keys. In terms of functionality, the punctured keys still allow to compute encryptions for all messages $x' \neq x$.

Why is puncturable FE enough? While the above puncturing requirement still does not say that the fake keys $\mathsf{EK}^\star_0$ and $\mathsf{EK}^\star_1$ are indistinguishable, it is sufficient for showing that indistinguishability obfuscations $i\mathcal{O}(\mathsf{Enc}(\mathsf{EK}^\star_0, \cdot))$ and $i\mathcal{O}(\mathsf{Enc}(\mathsf{EK}^\star_1, \cdot))$ of the corresponding encryption algorithms are in fact indistinguishable (even in the presence of the fake functional key $\mathsf{FSK}^\star$), provided that the underlying puncturable FE and IO are subexponentially-secure. This

[4] Roughly speaking, an obfuscation of a circuit C consists of a functional key for C and the public-encryption algorithm. Now, anyone could encrypt any input x, and obtain $C(x)$ by invoking functional decryption.

is shown using a hybrid technique commonly used in the context of IO [BGL+15, CHJV15, KLW15, CLTV15].

Roughly speaking, we can move from an obfuscation of one circuit to an obfuscation of the other through a sequence of hybrids $\{\mathcal{H}_x\}_{x\in\{0,1\}^n}$ ranging over all messages x of length n. In $\mathcal{H}_x$, the obfuscated circuit uses the key $\mathsf{EK}_1^\star$ to encrypt all messages x' that are lexicographically smaller than x, and uses the key $\mathsf{EK}_0^\star$ to encrypt all other messages. Going from one hybrid to the next, we consider a circuit where both keys are punctured at the point x, and the encryption of x is hardwired. The new circuit computes the exact same function as the original one, and thus this change cannot be detected when the circuit is (indistinguishability) obfuscated. Now, we can switch an encryption under one key to an encryption under the other relying on the security of the puncturable PFE.

The distinguishing gap between the hybrids additively grows as we move between one hybrid to the next, and overall degrades by a factor of 2^n. This is indeed the reason we need to start from subexponentially secure IO and puncturable FE (where the security parameter will be an appropriate polynomial in n).

The final construction. Overall, our input-extension step has the following form. To generate an obfuscation $i\mathcal{O}_n(C)$ of a circuit C with n-bit input $\mathbf{x} = x_1 \dots x_n$, generate a (real) functional key FSK_C together with a (real) encryption key EK and add an obfuscation

$$i\mathcal{O}_{n-1}(\mathsf{Enc}(\mathsf{EK}, x_1, \dots, x_{n-1}, 0) \circ \mathsf{Enc}(\mathsf{EK}, x_1, \dots, x_{n-1}, 1))$$

of the corresponding (bit fixing) encryption circuit, defined on $n-1$ bits. To evaluate the circuit on input $\mathbf{x}$, the evaluator runs the obfuscation on the $(n-1)$-bit prefix of $\mathbf{x}$ and chooses the encryption $\mathsf{Enc}(\mathsf{EK}, \mathbf{x})$ according to the last bit of $\mathbf{x}$. Then it simply performs functional decryption.

Indistinguishability between obfuscations of two functionally-equivalent circuits C_0 and C_1 is shown by first using the key-indistinguishability property to switch real keys $\mathsf{EK}, \mathsf{FSK}_{C_0}$, corresponding to an obfuscation of C_0, into fake keys $\mathsf{EK}_0, \mathsf{FSK}^\star$. Then, we switch to the fake keys $\mathsf{EK}_1, \mathsf{FSK}^\star$ using the puncturing property following the hybrid argument outlined above. Finally, we can switch back to real keys $\mathsf{EK}, \mathsf{FSK}_{C_1}$ corresponding to an encryption of C_1, using key indistinguishability again.

In terms of security, each invocation of the input extension step incurs a multiplicative loss of 2^n in the distinguishing gap, and overall when extending a trivial one-bit-input IO to n-bit-input IO, the overall degradation is roughly $2^{\sum_{i\in[n]} i} \approx 2^{n^2}$.

Constructing puncturable FE and the need for public-key encryption. Finally, we describe how to convert any public-key FE scheme FE into a new scheme PFE that is puncturable. The construction relies on similar techniques to those used for function hiding in the symmetric-key setting [BS15].

Concretely, to generate a key for a function f, we use the underlying scheme FE to generate a functional key FSK_g, for an augmented function g. The circuit $g = g[\mathsf{CT}_0, \mathsf{CT}_1]$ has two (plain) symmetric-key encryptions $\mathsf{CT}_0, \mathsf{CT}_1$, under two independently chosen secret keys $\mathsf{SK}_0, \mathsf{SK}_1$, hardwired into its code. In the actual scheme, both ciphertexts encrypt the circuit f. The function g expects as input, not only an input x for f, but also a key SK_b. Given those, it decrypts the corresponding ciphertext CT_b, and applies the decrypted circuit to the input x.

The (real) encryption key EK then consists of the encryption key $\mathsf{EK}_{\mathsf{FE}}$ of the underlying FE along with a secret key SK_b for a randomly chosen b, as well as a key K for a pseudorandom function. Encryption is done using the encryption of the underlying FE, encrypting not only the message x, but also the secret key SK_b; any randomness r required for the encryption algorithm is derived by applying the pseudorandom function to x, namely $r = \mathsf{PRF}_{\mathsf{K}}(x)$.

A fake encryption key $\mathsf{EK}_0^\star$ is distributed identically to a real key EK, consisting of $\mathsf{EK}_{\mathsf{FE}}, \mathsf{SK}_{b_0}, \mathsf{K}$, for a random bit b_0. The second key $\mathsf{EK}_1^\star$ only differs in the symmetric key, which is chosen to be the second key SK_{b_1}, where $b_1 = 1 - b_0$. As expected, the fake functional key $\mathsf{FSK}^\star$ corresponds now to $g[\mathsf{CT}_0, \mathsf{CT}_1]$ where CT_{b_0} encrypts f_0 and CT_{b_1} encrypts f_1 (rather than both encrypting the same function).

Proving key indistinguishability and puncturing naturally extends the ideas used to prove function-hiding in [BS15]. (There, the adversary, never actually sees the encryption key, and is only given an oracle that computes encryptions.) Key indistinguishability, follows directly from the security of the symmetric encryption scheme; indeed, the only difference between the real keys $\mathsf{EK}, \mathsf{FSK}_{f_0}$ and fake keys $\mathsf{EK}_0^\star, \mathsf{FSK}^\star$ is that in the latter the ciphertext CT_{b_1} embedded in $\mathsf{FSK}^\star$ encrypts f_1 instead of f_0. Since $\mathsf{EK}_0^\star$ only includes the secret key SK_{b_0} and is independent of SK_{b_1}, semantic security applies. A symmetric argument holds for the real keys $\mathsf{EK}, \mathsf{FSK}_{f_1}$ and fake keys $\mathsf{EK}_1^\star, \mathsf{FSK}^\star$.

We then need to prove that encryptions of any message x under the fake key $\mathsf{EK}_0^\star$ are indistinguishable from ones under $\mathsf{EK}_1^\star$, even in the presence of a punctured version of the keys. Let us first understand why such indistinguishability holds when the adversary does not obtain any encryption keys. In this case, which is analogous to [BS15], we can rely on the security of the underlying FE. Specifically, recall that the only difference between encryptions under the two keys is that when encrypting with $\mathsf{EK}_0^\star$, we use the underlying encryption to encrypt (x, SK_{b_0}), whereas when encrypting under $\mathsf{EK}_0^\star$, we encrypt (x, SK_{b_1}). Note, however, that since f_0 and f_1 compute the same function, the function $g[\mathsf{CT}_0, \mathsf{CT}_1]$ does distinguish between two such inputs. Indeed, in the first case it outputs $f_0(x)$ and in the second $f_1(x)$. This, in particular, means that two such encryptions are indistinguishable.

It is left to show that we can produce a punctured version of the encryption keys $\mathsf{EK}_0^\star\{x\}$, $\mathsf{EK}_1^\star\{x\}$ such that the above would hold even in their presence. This basically implies that we have to: (1) generate a punctured version of the encryption key $\mathsf{EK}_{\mathsf{FE}}$ in the underlying (non-function-hiding) functional encryption scheme, and (2) generate a punctured version of the pseudorandom functional key K. The second requirement can be dealt with easily relying on puncturable pseudorandom functions [BW13, KPTZ13, BGI14] — these are exactly pseudorandom functions where it is possible to puncture the secret key at any point x, so that given the punctured key $\mathsf{K}\{x\}$ the value $r = \mathsf{PRF}_{\mathsf{K}}(x)$ is pseudorandom.

The question is how to satisfy the first requirement, namely, ensure puncturing for the functional encryption key $\mathsf{EK}_{\mathsf{FE}}$. This is exactly where we rely on public-key functional encryption. Indeed, with public-key functional encryption the encryption key $\mathsf{EK}_{\mathsf{FE}} = \mathsf{PK}$ is trivially puncturable. That is, given the key PK itself (unchanged) indistinguishability is still guaranteed. This is in contrast to symmetric-key FE where in the presence of the encryption key, indistinguishability may no longer hold.[5]

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, under a similar assumption on the running-time of the encryption algorithm (which they term *compactness*). The two works take a somewhat different perspective to the problem. At high-level, Ananth and Jain show that functional encryption schemes as above can be converted into a multi-input functional encryption, a notion defined by Goldwasser et al. [GGG+14] that is

[5] The subsequent work [KNT17] shows how to construct a puncturable symmetric-key FE directly from plain symmetric-key FE, thereby allowing to base the entire transformation on symmetric-key FE.

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 proof techniques developed for IO.

- Brakerski, Komargodski, and Segev [BKS16] show how to convert any (single-input) symmetric-key functional encryption scheme into an $O(1)$-input symmetric-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 [AJS17] show how IO can be bootstrapped to always have linear-size overhead. By developing new techniques, they improve on the above Corollary 1.3 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.

- A progression of works [Lin16, LV16, AS17, Lin17, LT17] has shown how to reduce the degree of multi-linear maps required for constructing IO (assuming also the existence of appropriate local pseudorandom generators). The core of these works is a construction of FE, which is then bootstrapped to IO using our transformation.

- Bitansky, Nishimaki, Passelegue and Wichs [BNPW16] show how to obtain IO from sub-exponentially-secure *symmetric-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. Subsequently, Kitagawa, Nishimaki, and Tanaka [KNT17] removed the assumption of public-key encryption, showing that subexponentially-secure symmetric-key functional encryption implies public-key functional encryption.

- Garg, Pandey, Srinivasan and Zhandry [GPS16, GPSZ17] show how many of the applications of IO (such as the hardness of PPAD and multiparty key exchange) can be based instead directly on polynomially-secure functional encryption. Their reduction invokes a variant of our input-extension technique, but avoids the sub-exponential security loss. An general framework that captures these works was subsequently introduced by Liu and Zhandry [LZ17].
- Li and Micciancio [LM16] and Garg and Srinivasan [GS16] independently show a construction of a multi-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+13b], except that we lose a sub-exponential security factor that comes from invoking our transformation. The results of [LM16, GS16] avoid this loss. This, together with our result that transforms a polynomially secure collusion-resistant FE into a single-key FE scheme with succinct encryption circuits, 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 Turing machine is PPT if it is probabilistic and runs in polynomial time.
- For (a deterministic and time-bounded) algorithm A, we denote by $A(\cdot)$ the corresponding boolean circuit (computing the same function). We denote by $A(x, \cdot)$ the circuit where a prefix of the input is fixed to x.
- A 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 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. When we write $\mathcal{A}(x)$ for a circuit $\mathcal{A}$, without specifying the length of x, we assume that $\mathcal{A}$ only reads an ℓ-bit prefix of x, where ℓ is the input-size of the circuit $\mathcal{A}$.
- A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if $f(\lambda) = \lambda^{-\omega(1)}$.
- For random variables X and Y, distinguisher $\mathcal{D}$, and $0 < \mu < 1$, we write $X \approx_{\mathcal{D},\mu} Y$ if

$$|\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(Y) = 1]| \leq \mu.$$

- 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 $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for all λ,

$$X_\lambda \approx_{\mathcal{D},\mu(\lambda)} Y_\lambda \ .$$

For a concrete function δ, we denote by $\mathcal{X} \approx_\delta \mathcal{Y}$ the case that the above distinguishing gap is bounded by $\delta^{\Omega(1)}$. That is, for every polynomial-size distinguisher $\mathcal{D}$ there exists a constant $\alpha \leq 1$ such that $X_\lambda \approx_{\mathcal{D},\delta(\lambda)^\alpha} Y_\lambda$.

- Our constructions rely on primitives that are subexponentially-secure, or more generally δ-secure for some negligible δ. Typically, in the literature, this addresses adversaries of subexponential size that break the underlying system with subexponentially small probability. Like many other works in the context of indistinguishability obfuscation, for our purpose, it will be sufficient to require that polynomial-size (rather than subexponential-size) adversaries break the system with subexponentially-small probability.

2.2 Single-Key FE with Succinct Encryption

In this work, we consider a restricted notion of single-key functional encryption schemes where the function is known in setup time. This is in contrast to the typical stronger definition in the literature where the function is not known in setup time and the encryption key and a master decryption key are generated independently of the function. (Note that since we aim to construct IO from functional encryption, considering such a weaker notion only strengthens the result.) Furthermore, we will require certain succinctness of the encryption circuit, which will play an essential role in our constructions.

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 and 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 and a ciphertext CT and outputs $\widehat{m}$.

We next define the required correctness, security, and efficiency properties.

Definition 2.1 (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* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$*, 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* $f(m_0) = f(m_1)$*,*

$$\mathsf{PK}, \mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_0) \approx_{\mathcal{A},\mu} \mathsf{PK}, \mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_1) \ ,$$

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* $\delta(\lambda)$*, if for all polynomial-size adversaries the above distinguishing gap* $\mu(\lambda)$ *is smaller than* $\delta(\lambda)^{\Omega(1)}$.

3. **Succinct encryption circuit:** *there exists a polynomial* Φ *and a constant* $0 < \varepsilon \leq 1$*, such that for any input-size* n*, circuit-size* s*, and* s*-size function* $f : \{0,1\}^n \to \{0,1\}^*$*,*

$$|\mathsf{FE.Enc}(\mathsf{PK}, \cdot)| \leq s^{1-\varepsilon} \cdot \Phi(n, \lambda) \ ,$$

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

- *Encryption is* **fully succinct** *if* $\varepsilon = 1$.
- *Encryption is* **weakly depth-succinct** *if*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq s^{1-\varepsilon} \cdot \Phi(n, d, \lambda) \ .$$

- *Encryption is* **very weakly depth-succinct** *if*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq s^{1-\varepsilon} \cdot \Phi(n, 2^d, \lambda) \ .$$

In Section 4, we show that any weakly-depth-succinct scheme implies a succinct scheme. Furthermore, assuming the existence of pseudorandom functions in NC^1 this extends to very-weakly-depth-succinct schemes. Accordingly, throughout most of the paper, we restrict attention to succinct schemes (and do not address depth).

2.3 Indistinguishability Obfuscation

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

Definition 2.2 (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* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ *there exists a negligible function* $\mu(\lambda)$*, such that for any two circuits* $C_0, C_1 \in \mathcal{C}$ *that compute the same function and are of the same size:*

$$i\mathcal{O}(C_0, 1^\lambda) \approx_{\mathcal{D},\mu} i\mathcal{O}(C_1, 1^\lambda) \ ,$$

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

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

2.4 Puncturable Pseudorandom Functions

We consider a simple case of puncturable pseudorandom 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] pseudorandom function [BW13, KPTZ13, BGI14].

Definition 2.3 (Puncturable PRFs). *Let $k(\lambda)$ be a polynomially-bounded length function. 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 $\mathsf{K}\{x^*\}$ of the same size, so that the following conditions are satisfied:*

1. **Punctured-key correctness:** *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 $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function $\mu(\lambda)$, such that for all $\lambda \in \mathbb{N}$, and any $x^* \in \{0,1\}^*$,*

$$\mathsf{K}\{x^*\}, \mathsf{PRF}_{\mathsf{K}}(x^*) \approx_{\mathcal{D},\mu} \mathsf{K}\{x^*\}, r \ ,$$

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

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

Remark 2.4. The definition requires that punctured keys are of the same size as plain keys and that the same evaluation algorithm PRF works for both. We note that this requirement is w.l.o.g by appropriate padding.

Remark 2.5. The original [GGM86] construction of pseudorandom functions is described for a fixed input length. Here we define and use pseudorandom functions over arbitrary strings. A slight variant of the [GGM86] construction is known to achieve this [Gol01] and admits exactly the same puncturing properties (and proof of security).

2.5 One-Time Symmetric Encryption with Local Decryption

A one-time symmetric encryption scheme Sym with local decryption consists of a tuple of two PPT algorithms $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ with the following syntax:

- $\mathsf{Sym.Enc}(\mathsf{SK}, m)$ takes as input a symmetric key $\mathsf{SK} \in \{0,1\}^\lambda$, where λ is the security parameter, and a message $m \in \{0,1\}^n$, and outputs a size-n ciphertext CT.
- $\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i, i)$ takes as input the key SK a ciphertext bit CT_i and the index $i \in [n]$, and outputs the decrypted message bit m_i.

Throughout, we interpret $\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT})$, where no set of bits is specified as decrypting the entire ciphertext $\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}) = \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_1, 1), \ldots, \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_n, n)$.

Definition 2.6 (One-time symmetric encryption with local decryption). *A pair of deterministic polynomial-time algorithms $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ is a one-time symmetric encryption scheme for message space $\{0,1\}^*$ if it satisfies:*

1. **Correctness:** *for every security parameter* $\lambda \in \mathbb{N}$*, message length* $n \in \mathbb{N}$*, message* $m \in \{0,1\}^n$*, and index* $i \in [n]$*,*

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

   We say that Sym *is* **shallow** *if it has a decryption circuit of depth* $\log \lambda$*.*

2. **One-time indistinguishability:** *for any polynomial-size distinguisher* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ *there exists a negligible function* $\mu(\lambda)$*, such that for all* $\lambda \in \mathbb{N}$*, and any equal size messages* m_0, m_1*,*

$$\mathsf{Sym.Enc}(\mathsf{SK}, m_0) \approx_{\mathcal{D},\mu} \mathsf{Sym.Enc}(\mathsf{SK}, m_1) \ ,$$

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

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

Such encryption schemes follow from any pseudorandom function (or local pseudorandom generator). Given pseudorandom functions in NC^1 (which are known under various standard assumptions [BR17]), there exist such schemes that are also shallow.

Fact 2.7 ([GGM86, HILL99])**.** *Assuming (*δ*-secure) one-way functions, there exists a (*δ*-secure) one-time symmetric encryption with local decryption. Assuming pseudorandom functions in* NC^1*, the scheme is shallow.*

2.6 Decomposable Randomized Encodings

We rely on the notion of decomposable randomized encodings (REs) from [Yao86, IK00, AIK04, AIK06]. There are different variants of REs in the literature differing in their security and decomposition (or locality) properties. Here we define the properties that will be used in our constructions, which are satisfied by Yao's garbled circuit [Yao86] in conjunction with appropriate pseudorandom functions.

Such a scheme RE consists of two polynomial-time algorithms $(\mathsf{RE.Enc}, \mathsf{RE.Dec})$ with the following syntax:

- $\mathsf{RE.Enc}(f, 1^\lambda)$: takes as input a circuit $f : \{0,1\}^n \to \{0,1\}^*$ and a security parameter 1^λ and outputs a new encoder circuit $\hat{f} : \{0,1\}^n \times \{0,1\}^\lambda \to \{0,1\}^*$.
- $\mathsf{RE.Dec}(\hat{y})$ takes as input an encoding $\hat{y}$ and outputs y.

We next the define the required correctness, security, and decomposability properties.

Definition 2.8 (Decomopsable randomized encoding)**.** *A pair of polynomial-time algorithms* $\mathsf{RE} = (\mathsf{RE.Enc}, \mathsf{RE.Dec})$ *is a decomposable randomized encoding if it satisfies:*

1. **Correctness:** *for every* $\lambda, n \in \mathbb{N}$*, circuit* $f : \{0,1\}^n \to \{0,1\}^*$*, and input* $x \in \{0,1\}^n$*,*

$$\Pr\left[\mathsf{RE.Dec}(\hat{f}(x; r)) = f(x) \;\middle|\; \begin{array}{c} \hat{f} \leftarrow \mathsf{RE.Enc}(f, 1^\lambda) \\ r \leftarrow \{0,1\}^\lambda \end{array}\right] = 1 \ .$$

2. **Privacy:** *there exists a PPT simulator* $\mathsf{RE.Sim}$ *such that for any polynomial-size adversary* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$*, there exists a negligible function* $\mu(\lambda)$ *such that for any* $\lambda, n \in \mathbb{N}$*, any* $f : \{0,1\}^n \to \{0,1\}^m$*, and* $x \in \{0,1\}^n$*,*

$$\hat{f}(x;r) \approx_{\mathcal{A},\mu} \mathsf{RE.Sim}(f, f(x)) \ ,$$

where $\hat{f} \leftarrow \mathsf{RE.Enc}(f, 1^\lambda)$ *and* $r \leftarrow \{0,1\}^\lambda$*.*

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

3. **Decomopsability:** *there exists a polynomial* Φ *such that for any input size* $n \in \mathbb{N}$ *and size-*s *circuit* $f : \{0,1\}^n \to \{0,1\}^*$*, the encoder circuit* $\hat{f} \leftarrow \mathsf{RE.Enc}(f, 1^\lambda)$ *can be decomposed into* $\ell = s \cdot \Phi(n, \lambda)$ *circuits*

$$\hat{f}(x;r) = (\hat{f}_1(x;r), \ldots, \hat{f}_\ell(x;r)) \ ,$$

each of size $\Phi(n, \lambda)$ *and with single output bit.*

We say that RE *is* **shallow** *if the depth of each* $\hat{f}_i$ *is* $\log \lambda$*.*

Such randomized encodings can be constructed from one-time symmetric encryption (Definition 2.6) and pseudorandom functions (which can in turn be constructed from one-way functions) based on Yao's garbled circuit. Assuming pseudorandom functions in NC^1, the randomized encodings are shallow.

Fact 2.9 ([Yao86, AIK06])**.** *Assuming (*δ*-secure) one-way functions there exists a (*δ*-secure) decomposable randomized encoding. Assuming also pseudorandom functions in* NC^1*, it is shallow.*

3 The Transformation

In this section, we describe the transformation from FE to IO and analyze it. The transformation consists of two steps. First, in Section 3.1, we define and construct a notion of puncturable functional encryption with a succinct encryption circuit that satisfies a weak form of function hiding. Then, in Section 3.2, we construct IO from such FE schemes.

3.1 Puncturable Functional Encryption

We now define the abstraction of Puncturable Functional Encryption (PFE) that will be used in our transformation.

Such a scheme PFE, for a function class $\mathcal{F}$ (represented by boolean circuits) and message space $\{0,1\}^*$, consists of three polynomial-time algorithms $(\mathsf{PFE.Setup}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec})$, the first probabilistic and the other two deterministic, with the following syntax:

- $\mathsf{PFE.Setup}(1^\lambda, f)$: takes as input a security parameter λ in unary and function $f \in \mathcal{F}$ with domain $\{0,1\}^n$, and outputs an encryption key EK and a functional key FSK_f.
- $\mathsf{PFE.Enc}(\mathsf{EK}, m)$: takes as input an encryption key EK and a message $m \in \{0,1\}^n$ and outputs a ciphertext CT encrypting m.
- $\mathsf{PFE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: takes as input a functional key FSK_f and a ciphertext CT and outputs $\widehat{m}$.

We next define the required correctness, security, and efficiency properties. To define the security properties, we require the existence of two additional algorithms:

- $\mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$: a PPT algorithm that takes as input the security parameter λ, and two functions $f_0, f_1 \in \mathcal{F}$ with domain $\{0,1\}^n$ and outputs a pair of keys $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star)$ and a fake functional key $\mathsf{FSK}^\star$.
- $\mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, m)$: a deterministic polynomial-time algorithm that takes as input keys $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star)$ and a message $m \in \{0,1\}^n$ and outputs a punctured keys $(\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\})$.

Definition 3.1 (Puncturable functional encryption)**.** $\mathsf{PFE} = (\mathsf{PFE.Setup}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec})$ *is a puncturable 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 $\{0,1\}^n$,*

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

2. **Punctured-key Correctness:** *for every $\lambda, n \in \mathbb{N}$, message $m \in \{0,1\}^n$, and functions $f_0, f_1 \in \mathcal{F}$, and $b \in \{0,1\}$,*

$$\Pr\left[\forall m' \in \{0,1\}^n \setminus \{m\} : \mathsf{PFE.Enc}(\mathsf{EK}_b^\star\{m\}, m') = \mathsf{PFE.Enc}(\mathsf{EK}_b^\star, m')\right] = 1 \ ,$$

   where $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$ and $(\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}) = \mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, m)$.

3. **Key indistinguishability:** *for any polynomial-size adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function $\mu(\lambda)$, such that for any $\lambda \in \mathbb{N}$, equal-size functions $f_0, f_1 \in \mathcal{F}$, and $b \in \{0,1\}$:*

$$\mathsf{EK}_b, \mathsf{FSK}_{f_b} \approx_{\mathcal{A},\mu} \mathsf{EK}_b^\star, \mathsf{FSK}^\star \ ,$$

   where $(\mathsf{EK}_b, \mathsf{FSK}_{f_b}) \leftarrow \mathsf{PFE.Setup}(1^\lambda, f_b)$ and $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$.

4. **Indistinguishability at punctured points:** *for any polynomial-size adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function $\mu(\lambda)$, such that for any $\lambda, n \in \mathbb{N}$, $m \in \{0,1\}^n$, and equal-size functions $f_0, f_1 \in \mathcal{F}$ such that $f_0(m) = f_1(m)$,*

$$\begin{aligned}&\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}, \mathsf{FSK}^\star, \mathsf{PFE.Enc}(\mathsf{EK}_0^\star, m) \approx_{\mathcal{A},\mu} \\ &\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}, \mathsf{FSK}^\star, \mathsf{PFE.Enc}(\mathsf{EK}_1^\star, m) \ ,\end{aligned}$$

   where $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$ and $(\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}) = \mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, m)$.

   We further say that PFE is δ-secure, for some concrete negligible function $\delta(\lambda)$, if for all polynomial-size adversaries the above distinguishing gaps $\mu(\lambda)$ are smaller than $\delta(\lambda)^{\Omega(1)}$.

5. **Succinct encryption circuit:** *there exists a polynomial Φ and a constant $0 < \varepsilon \leq 1$, such that for any input-size n, circuit-size s, s-size functions $f, f' : \{0,1\}^n \to \{0,1\}^*$, $m \in \{0,1\}^n$, and $b \in \{0,1\}$,*

$$|\mathsf{PFE.Enc}(\mathsf{EK}_b^\star, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}_b^\star\{m\}, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)| \leq s^{1-\varepsilon} \cdot \Phi(n, \lambda) \ ,$$

   where $(\mathsf{EK}, \mathsf{FSK}) \leftarrow \mathsf{PFE.Setup}(1^\lambda, f)$, $(\mathsf{EK}_0^\star, \mathsf{EK}_0^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f, f')$, $(\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}) \leftarrow \mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_0^\star, m)$.

   Encryption is **fully succinct** *if $\varepsilon = 1$.*

Puncturable FE from single-key FE. We now show that any single-key, selectively-secure public-key FE scheme (Definition 2.1) implies a puncturable FE scheme that has essentially the same succinctness properties as the original scheme.

Ingredients. We rely on the following primitives:

- A single-key, selectively-secure, public-key functional encryption scheme FE for all circuits.
- A one-time symmetric encryption scheme Sym.
- A puncturable pseudo-random function family $\mathcal{PRF}$.

The constructed scheme PFE consists of the algorithms

$$(\mathsf{PFE.Setup}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec}, \mathsf{PFE.Setup}^\star, \mathsf{PFE.Punc})$$

described in Figure 1.

Theorem 3.2. PFE *is a puncturable functional encryption for all circuits. If* FE *is succinct so is* PFE. *If* $\mathsf{FE}, \mathsf{Sym}, \mathcal{PRF}$ *are all* δ*-secure so is* PFE.

Proof. We prove that the constructed scheme satisfies the properties of a PFE (Definition 3.1).

Correctness: Fix a security parameter $\lambda \in \mathbb{N}$, message $m \in \{0,1\}^n$, and $f \in \mathcal{F}$ with domain $\{0,1\}^n$. By the functionality of FE and correctness of Sym,

$$\begin{aligned}\mathsf{PFE.Dec}(\mathsf{FSK}_f, \mathsf{FCT}) = \ & \mathsf{FE.Dec}(\mathsf{FSK}_F, \mathsf{FCT}) = \\ & F(m, \mathsf{SK}_\beta, \beta) = \\ & U(\mathsf{Sym.Dec}(\mathsf{SK}_\beta, \mathsf{CT}_\beta), m) = \\ & U(f, m) = f(m) \ ,\end{aligned}$$

where $(\mathsf{EK}, \mathsf{FSK}_f) \leftarrow \mathsf{PFE.Setup}(1^\lambda, f)$, $\mathsf{FCT} = \mathsf{PFE.Enc}(\mathsf{EK}, m)$, $\mathsf{EK} = (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_\beta, \beta)$, F is the function underlying FSK_f, and $\mathsf{CT}_0, \mathsf{CT}_1$ are the ciphertexts corresponding to F.

Punctured-key correctness: Fix a security parameter $\lambda \in \mathbb{N}$, message $m \in \{0,1\}^n$, and $f_0, f_1 \in \mathcal{F}$ with domain $\{0,1\}^n$. By the punctured-key correctness of $\mathcal{PRF}$, for any $m' \in \{0,1\}^n \setminus \{m\}$ and $b \in \{0,1\}$,

$$\begin{aligned}\mathsf{PFE.Enc}(\mathsf{EK}_b^\star\{m\}, m') = \ & \mathsf{FE.Enc}(\mathsf{PK}, (m', \mathsf{SK}_{\beta_b}, \beta_b); \mathsf{PRF}_{\mathsf{K}\{m\}}(m')) = \\ & \mathsf{FE.Enc}(\mathsf{PK}, (m', \mathsf{SK}_{\beta_b}, \beta_b); \mathsf{PRF}_{\mathsf{K}}(m')) = \\ & \mathsf{PFE.Enc}(\mathsf{EK}_b^\star, m') \ ,\end{aligned}$$

where $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$, $\mathsf{EK}_b^\star = (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_{\beta_b}, \beta_b)$, $(\mathsf{EK}_0^\star\{m\}, \mathsf{EK}_1^\star\{m\}) = \mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, m)$, and $\mathsf{K}\{m\} = \mathsf{Punc}(\mathsf{K}, m)$.

Key indistinguishability: Fix any polynomial-size $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$. By the indistinguishability of Sym, there exists a negligible $\mu_{\mathsf{Sym}}(\lambda)$, such that for any $\lambda, n \in \mathbb{N}$, $m \in \{0,1\}^n$, equal-size $f_0, f_1 \in \mathcal{F}$ with domain $\{0,1\}^n$, and $b \in \{0,1\}$,

$$\mathsf{EK}, \mathsf{FSK}_{f_b} \approx_{\mathcal{A}, \mu_{\mathsf{Sym}}} \mathsf{EK}_b^\star, \mathsf{FSK}^\star \ ,$$

where $(\mathsf{EK}, \mathsf{FSK}_{f_b}) \leftarrow \mathsf{PFE.Setup}(1^\lambda, f_b)$ and $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$.

Indeed, EK and EK_b are identically distributed, whereas the only difference between FSK_{f_b} and $\mathsf{FSK}^\star$ is in how the underlying function is generated. In the first, we generate F such that

PFE

- $\mathsf{PFE.Setup}(1^\lambda, f)$:
  - Generate:
    - Symmetric encryption keys $(\mathsf{SK}_0, \mathsf{SK}_1) \leftarrow \{0,1\}^\lambda \times \{0,1\}^\lambda$.
    - Symmetric encryptions $(\mathsf{CT}_0, \mathsf{CT}_1) = \mathsf{Sym.Enc}(\mathsf{SK}_0, f) \times \mathsf{Sym.Enc}(\mathsf{SK}_1, f)$.
    - A circuit F defined for $(m, \mathsf{SK}, \beta) \in \{0,1\}^n \times \{0,1\}^\lambda \times \{0,1\}$ by
    $$F(m, \mathsf{SK}, \beta) = U(\mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_\beta), m) \ ,$$
    where $U(\cdot,\cdot)$ is the universal circuit.
    - Public key and functional key $(\mathsf{PK}, \mathsf{FSK}_F) \leftarrow \mathsf{FE.Setup}(1^\lambda, F)$.
    - Seed $\mathsf{K} \leftarrow \mathsf{Gen}_{\mathcal{PRF}}(1^\lambda)$ for a puncturable pseudo random function.
    - A random bit $\beta \leftarrow \{0,1\}$.
  - Output:
    - $\mathsf{EK} := (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_\beta, \beta)$.
    - $\mathsf{FSK}_f := \mathsf{FSK}_F$.
- $\mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$:
  - Generate everything as $\mathsf{PFE.Setup}(1^\lambda, f_0)$, except that instead of generating F as above, we generate $F^\star$ with $\mathsf{CT}_\beta = \mathsf{Sym.Enc}(\mathsf{SK}_\beta, f_0)$ and $\mathsf{CT}_{1-\beta} = \mathsf{Sym.Enc}(\mathsf{SK}_{1-\beta}, f_1)$, where again β is a random bit. That is, now one (random) ciphertext corresponds to f_0 and the other to f_1 (rather than both to the same f).
  - Output:
    - $\mathsf{EK}_0^\star := (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_\beta, \beta)$ and $\mathsf{EK}_1^\star := (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_{1-\beta}, 1-\beta)$.
    - $\mathsf{FSK}^\star := \mathsf{FSK}_{F^\star}$.
- $\mathsf{PFE.Enc}(\mathsf{EK}, m)$:
  - Parse $\mathsf{EK} = (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_\beta, \beta)$.
  - Output $\mathsf{FCT} = \mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}_\beta, \beta); \mathsf{PRF}_\mathsf{K}(m))$.
- $\mathsf{PFE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$:
  - Output $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$.
- $\mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, m)$:
  - Parse $\{\mathsf{EK}_b^\star = (\mathsf{PK}, \mathsf{K}, \mathsf{SK}_{\beta_b}, \beta_b)\}_{b\in\{0,1\}}$.
  - Compute $\mathsf{K}\{m\} = \mathsf{Punc}(\mathsf{K}, m)$.
  - Output $\{\mathsf{EK}_b^\star = (\mathsf{PK}, \mathsf{K}\{m\}, \mathsf{SK}_{\beta_b}, \beta_b)\}_{b\in\{0,1\}}$.

Figure 1: A puncturable functional encryption

both ciphertexts $\{\mathsf{CT}_\alpha = \mathsf{Sym.Enc}(\mathsf{SK}_\alpha, f_b)\}_{\alpha\in\{0,1\}}$ encrypt the same function f_b. In the second, $\mathsf{CT}_{1-\beta_b} = \mathsf{Sym.Enc}(\mathsf{SK}_{1-\beta_b}, f_{1-b})$ encrypts f_{1-b}. The key EK, respectively $\mathsf{EK}_b^\star$, only include the key SK_{β_b} and not the key $\mathsf{SK}_{1-\beta_b}$. Thus the indistinguishability of Sym applies.

Indistinguishability at punctured points: Fix any polynomial-size $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$. By pseudo-randomness at punctured points of $\mathcal{PRF}$ and selective security of FE. there exist a negligible $\mu_{\mathsf{PRF}}(\lambda), \mu_{\mathsf{FE}}(\lambda)$, such that for any $\lambda, n \in \mathbb{N}$, $m \in \{0,1\}^n$, $f_0, f_1 \in \mathcal{F}$ such that $f_0(m) = f_1(m)$,

and $b \in \{0,1\}$,

$$\begin{aligned}
&\{\mathsf{EK}^\star_\alpha\{m\}\}_{\alpha\in\{0,1\}}, \mathsf{FSK}^\star, \mathsf{PFE.Enc}(\mathsf{EK}^\star_b, m)) = \\
&\{\mathsf{PK}, \mathsf{K}\{m\}, \mathsf{SK}_{\beta_\alpha}, \beta_\alpha\}_{\alpha\in\{0,1\}}, \mathsf{FSK}^\star, \mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}_{\beta_b}, \beta_b); \mathsf{PRF}_\mathsf{K}(m)) \approx_{\mathcal{A},\mu_\mathsf{PRF}} \\
&\{\mathsf{PK}, \mathsf{K}\{m\}, \mathsf{SK}_{\beta_\alpha}, \beta_\alpha\}_{\alpha\in\{0,1\}}, \mathsf{FSK}^\star, \mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}_{\beta_b}, \beta_b); r) \approx_{\mathcal{A},\mu_\mathsf{FE}} \\
&\{\mathsf{PK}, \mathsf{K}\{m\}, \mathsf{SK}\beta_\alpha, \beta_\alpha\}_{\alpha\in\{0,1\}}, \mathsf{FSK}^\star, \mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}_0, 0); r) \ ,
\end{aligned}$$

where $(\mathsf{EK}^\star_0, \mathsf{EK}^\star_1, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, f_0, f_1)$, $(\mathsf{EK}^\star_0\{m\}, \mathsf{EK}^\star_1\{m\}) = \mathsf{PFE.Punc}(\mathsf{EK}^\star_0, \mathsf{EK}^\star_1, m)$, $\mathsf{K}\{m\} = \mathsf{Punc}(\mathsf{K}, m)$, and $r \leftarrow \{0,1\}^\lambda$.

The first indistinguishability follows from pseudorandomness at punctured points. The second indistinguishability follows from the selective security of FE and the fact that the function $F^\star$ underlying $\mathsf{FSK}^\star$ satisfies

$$F^\star(m, \mathsf{SK}_{\beta_b}, \beta_b) = U(\mathsf{Sym.Dec}(\mathsf{SK}_{\beta_b}, \mathsf{CT}_{\beta_b}), m) = U(f_b, m) = f_b(m) = f_0(m) = F^\star(m, \mathsf{SK}_0, 0) \ .$$

Overall, the distribution $\mathsf{EK}^\star_0\{m\}, \mathsf{EK}^\star_1\{m\}, \mathsf{FSK}^\star, \mathsf{PFE.Enc}(\mathsf{EK}^\star_b, m))$ is indistinguishable from a distribution that is independent of b, and thus the two distributions corresponding to $b \in \{0,1\}$ are indistinguishable.

For the last two properties, it follows readily that if $\mathsf{Sym}, \mathcal{PRF}, \mathsf{FE}$ are δ-secure for some concrete negligible δ then PFE is δ-secure.

Succinct encryption: Assume that FE is succinct. That is, there exist a polynomial Φ_FE and constant $0 < \varepsilon \leq 1$ such that the size of its encryption circuit is bounded by

$$s^{1-\varepsilon} \cdot \Phi_\mathsf{FE}(n, \lambda) \ ,$$

where n, s are the input-size and circuit-size of the function chosen during the setup phase.

We show that PFE is also succinct with parameters $(\varepsilon, \Phi_\mathsf{PFE})$ for a fixed polynomial Φ_PFE. First, observe that fake keys $\mathsf{EK}^\star$ and fake punctured keys $\mathsf{EK}^\star\{m\}$, for any message $m \in \{0,1\}^n$, are of the same size as real keys $\mathsf{EK}^\star$, and the corresponding encryption circuit is of the same size:

$$|\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}^\star, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}^\star\{m\}, \cdot)|$$

We now bound the size of $\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)$, which is of the form $\mathsf{FE.Enc}(\mathsf{PK}, (\cdot, \mathsf{SK}, \beta); \mathsf{PRF}_\mathsf{K}(\cdot))$.

Observe that

- $\mathsf{PRF}_\mathsf{K}(\cdot)$, for messages of size n, can be computed by a circuit of size $\Phi_\mathsf{PRF}(n, \lambda)$ for a fixed polynomial Φ_PRF.
- Given $r = \mathsf{PRF}_\mathsf{K}(m)$ as input, $\mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}, \beta); r)$ can be computed by a circuit of size $s'^{1-\varepsilon}\Phi_\mathsf{FE}(n', \lambda)$, where in PFE, messages m of size n translate to messages (m, SK, β) of size n' and circuits $f(m)$ of size s translate to circuits $F(m, \mathsf{SK}, \beta)$ of size s' in FE.

Observe that $n' = n + \lambda + 1$. It is left to bound s'. F chooses CT_β according to β and applies $U(\mathsf{Sym.Dec}(\mathsf{SK}_\beta, \mathsf{CT}_\beta), m)$. By Fact 2.7, each CT_β encrypting f is of size s and applying $\mathsf{Sym.Dec}$ requires size $s \cdot \Phi_\mathsf{Sym}(\lambda)$ for a fixed polynomial Φ_Sym. Applying a universal circuit for an input circuit of size s requires size $s \cdot \Phi_U(\log s) \leq s \cdot \Phi_U(\lambda)$ for a fixed polynomial Φ_U [Val76]. Thus overall,

$$s' \leq s \cdot \Phi_{\mathsf{Sym},U}(\lambda) \ ,$$

for a fixed polynomial $\Phi_{\mathsf{Sym},U}(\lambda)$ that aggregates the above.

In conclusion, we can bound the size of the encryption circuit $\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)$ by

$$\Phi_\mathsf{PRF}(n, \lambda) + s'^{1-\varepsilon}\Phi_\mathsf{FE}(n', \lambda) \leq s^{1-\varepsilon}\left(\Phi_\mathsf{PRF}(n, \lambda) + \Phi_{\mathsf{Sym},U}(\lambda) \cdot \Phi_\mathsf{FE}(n + \lambda + 1, \lambda)\right) \ ,$$

establishing our requirement with $\Phi_\mathsf{PFE}(n, \lambda) := \Phi_\mathsf{PRF}(n, \lambda) + \Phi_{\mathsf{Sym},U}(\lambda) \cdot \Phi_\mathsf{FE}(n + \lambda + 1, \lambda)$.

□

3.2 From Puncturable Functional Encryption to Indistinguishability Obfuscation

In this section, we show how to transform any puncturable functional encryption with succinct encryption, such as the one constructed in the previous section, into an indistinguishability obfuscator.

Ingredient. We rely on a puncturable functional encryption scheme

$$\mathsf{PFE} = (\mathsf{PFE.Setup}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec}, \mathsf{PFE.Setup}^\star, \mathsf{PFE.Punc}) \ .$$

Notation: Throughout, bit-strings will be boldfaced, whereas bits will not. In particular, for a string $\mathbf{x} \in \{0,1\}^n$, we denote by $\mathbf{x}_j$ its j-long prefix and by x_j the jth bit in the string.

The obfuscator. The obfuscator, formally given in Figure 2, is parameterized by a function $\tilde{\lambda}(n,\lambda) \geq \lambda$ (of the circuit input-length and the security parameter), which will define the security parameter with which PFE will be invoked. The security and the complexity of the obfuscator will depend on the choice of the function.

A high-level description. Given a circuit $C : \{0,1\}^n \to \{0,1\}^m$ and security parameter λ, the obfuscator $i\mathcal{O}_{\tilde{\lambda}}(C, 1^\lambda)$, first computes a new security parameter $\tilde{\lambda} = \tilde{\lambda}(n,\lambda)$, and invokes a recursive obfuscation procedure $r\mathcal{O}.\mathsf{Obf}(n, C, 1^{\tilde{\lambda}})$, formally described in Figure 3. A corresponding recursive evaluation procedure $r\mathcal{O}.\mathsf{Eval}$ is described right after in Figure 4.

The recursive obfuscation procedure $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ extends obfuscation for circuits with $i-1$ input-bits to obfuscation for circuits with i input-bits. To this end, it generates an obfuscation of an encryption circuit $\mathcal{E}_{i-1}$ and a corresponding functional key FSK_i, under the puncturable functional encryption scheme PFE. The encryption circuit $\mathcal{E}_{i-1}$ takes a prefix $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ and generates two encryptions — one for of each possible continuation $\mathbf{x}_{i-1}0$ and $\mathbf{x}_{i-1}1$. The corresponding functional key FSK_i allows to evaluate the circuit C_i on the encrypted input .

Unrolling this recursive process, we obtain "a tree of encryptions", where for each node corresponding to some input-prefix $\mathbf{x}_i$, there is a corresponding encryption $\mathsf{CT}_{\mathbf{x}_i}$ of that prefix, and the node's children can be computed using the corresponding functional key FSK_i. The leaves, correspond to encryptions of the full input $\mathbf{x} = \mathbf{x}_n$, and the last functional key FSK_n corresponds to the original circuit C. Overall, to evaluate the obfuscation at an input $\mathbf{x}$, one expands the tree along the path corresponding to $\mathbf{x}$, until obtaining $C(\mathbf{x})$ at the final level.

$$i\mathcal{O}_{\tilde{\lambda}}(C, 1^\lambda)$$

Input: A circuit $C : \{0,1\}^n \to \{0,1\}^m$ and security parameter λ.

1. Compute a new security parameter $\tilde{\lambda} = \tilde{\lambda}(n,\lambda)$.
2. Output $r\mathcal{O}.\mathsf{Obf}(n, C, 1^{\tilde{\lambda}})$.

Figure 2: The obfuscator.

Theorem 3.3. *Let λ be a security parameter, let n denote circuit-input size, and let $\tilde{\lambda}(n,\lambda)$ be a function. Then if* PFE *is $\delta(\tilde{\lambda})$-secure, for security parameter $\tilde{\lambda}$, $i\mathcal{O}_{\tilde{\lambda}}$, using security parameter λ, is a $2^{n^2}\delta(\tilde{\lambda})$-secure indistinguishability obfuscator for all circuits .*

The theorem immediately implies that subexponentially-secure puncturable functional encryption (which in turn, can be constructed from subexponentially-secure public-key functional encryption) implies an indistinguishability obfuscator.

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

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 an encryption key and functional key $(\mathsf{EK}_i, \mathsf{FSK}_i) \leftarrow \mathsf{PFE.Setup}(1^{\tilde{\lambda}}, C_i)$.
   - Generate the circuit $\mathcal{E}_{i-1}$ defined for any $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ by
   $$\mathcal{E}_{i-1}(\mathbf{x}_{i-1}) = \{\mathsf{PFE.Enc}(\mathsf{EK}_i, (\mathbf{x}_{i-1}, x_i))\}_{x_i \in \{0,1\}} \quad .$$
   - Pad $\mathcal{E}_{i-1}$ to size $\ell = \ell(|\mathcal{E}_{i-1}|)$ for some fixed linear function $\ell(s) = c \cdot s$ determined in the analysis. We denote by $\mathcal{E}^{\ell}_{i-1}$ the padded circuit.
   - Generate an obfuscation
   $$\widetilde{\mathcal{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathcal{E}^{\ell}_{i-1}, 1^{\tilde{\lambda}}) \ .$$
3. Output $(\widetilde{\mathcal{E}}_{i-1}, \mathsf{FSK}_i)$.

Figure 3: The recursive obfuscation procedure.

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

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

1. If $i = 1$, parse $\widetilde{\mathcal{E}}_1 = (\mathsf{CT}_0, \mathsf{CT}_1)$, and output $\mathsf{CT}_{\mathbf{x}_1}$.
2. Otherwise, compute $(\mathsf{CT}_{\mathbf{x}_{i-1}0}, \mathsf{CT}_{\mathbf{x}_{i-1}1}) = r\mathcal{O}.\mathsf{Eval}(i-1, \widetilde{\mathcal{E}}_{i-1}, \mathbf{x}_{i-1})$, where $\mathbf{x}_{i-1}$ are the first $i-1$ bits of $\mathbf{x}_i$.
3. Output $\mathsf{FE.Dec}(\mathsf{FSK}_i, \mathsf{CT}_{\mathbf{x}_i})$.

Figure 4: The recursive evaluation procedure.

Corollary 3.4. *If* PFE *is* $2^{-\tilde{\lambda}^{\alpha}}$*-secure, for security parameter* $\tilde{\lambda}$*, then* $i\mathcal{O}_{\tilde{\lambda}}$ *is* $\lambda^{-\omega(1)}$*-secure, for security parameter* λ*, and any function* $\tilde{\lambda}(n, \lambda) = \omega((n^2 + \log \lambda)^{\alpha^{-1}})$.

Remark 3.5 (Technical remark on the proof). One may think of our obfuscator as iteratively applying an input-extension procedure that takes an obfuscator for $i-1$ bits and produces an obfuscator for i bits. Each such step incurs a certain security loss in the distinguishing gap, which may of course depend on the size of the specific analyzed adversary. We formally analyze the obfuscator as a whole, rather than analyzing a single input-extension step. This allows us to capture the overall loss of an n-step extension process for any specific adversary (or rather any specific adversary size).

Proof of Theorem 3.3. We prove that the constructed scheme is an indistinguishability obfuscator.

Functionality. Correctness follows the intuition outlined above — an evaluation corresponds to gradually constructing an encryption of the input $\mathbf{x} = \mathbf{x}_n$, by using the corresponding function at every level of the tree, and eventually applying the last functional key corresponding to the obfuscated circuit C.

We formally prove correctness by induction on the input length $1 \leq i \leq n$. Specifically, we prove that for any circuit $C_i : \{0,1\}^i \to \{0,1\}^*$, letting $\widetilde{\mathcal{E}}_i = r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$, it holds that for any input $\mathbf{x}_i \in \{0,1\}^i$, $r\mathcal{O}.\mathsf{Eval}(i, \widetilde{\mathcal{E}}_i, \mathbf{x}_i) = C_i(\mathbf{x}_i)$. Correctness then follows by considering the full input $\mathbf{x} = \mathbf{x}_n$ and the obfuscated circuit $C_n = C$.

For the base case $i = 1$, it holds by definition that $\widetilde{\mathcal{E}}_1 = C_1(0), C_1(1)$ and for any $\mathbf{x}_1 \in \{0,1\}$, $r\mathcal{O}.\mathsf{Eval}(1, \widetilde{\mathcal{E}}_1, \mathbf{x}_1) = C_1(\mathbf{x}_1)$. We now assume that the required correctness holds for $i-1$ and show that it holds for i.

By definition

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

where $\widetilde{\mathcal{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathcal{E}^{\ell}_{i-1}, 1^{\tilde{\lambda}})$, $(\mathsf{EK}_i, \mathsf{FSK}_i) \leftarrow \mathsf{PFE.Setup}(1^{\tilde{\lambda}}, C_i)$, and $\mathcal{E}^{\ell}_{i-1}(\mathbf{x}_{i-1})$ is a circuit that computes two encryptions $\left\{\mathsf{CT}_{\mathbf{x}_{i-1}b} = \mathsf{PFE.Enc}(\mathsf{EK}_i, \mathbf{x}_{i-1}b)\right\}_{b\in\{0,1\}}$ (padded to size $\ell(|\mathcal{E}_{i-1}|)$).

By the induction hypothesis, we know that $r\mathcal{O}.\mathsf{Eval}(i-1, \widetilde{\mathcal{E}}_{i-1}, \mathbf{x}_{i-1}) = (\mathsf{CT}_{\mathbf{x}_{i-1}0}, \mathsf{CT}_{\mathbf{x}_{i-1}1})$; namely, the obfuscation $\widetilde{\mathcal{E}}_{i-1}$ preserves the functionality of the obfuscated circuit $\widetilde{\mathcal{E}}_i$. It then, follows by definition that

$$r\mathcal{O}.\mathsf{Eval}(i, \widetilde{\mathcal{E}}_i, \mathbf{x}_i) = \mathsf{PFE.Dec}(\mathsf{FSK}_i, \mathsf{CT}_{\mathbf{x}_i}) \ ,$$

which by the functionality of PFE is equal to $C_i(\mathbf{x}_i)$ as required.

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

We aim to bound

$$\begin{aligned}\delta_{i\mathcal{O}}(\lambda) := &\max_{C_0,C_1} \left| \Pr\left[\mathcal{D}(i\mathcal{O}_{\tilde{\lambda}}(C_0, 1^\lambda)) = 1\right] - \Pr\left[\mathcal{D}(i\mathcal{O}_{\tilde{\lambda}}(C_1, 1^\lambda)) = 1\right] \right| = \\ &\max_{C_0,C_1} \left| \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(n, C_0, 1^{\tilde{\lambda}})) = 1\right] - \Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(n, C_1, 1^{\tilde{\lambda}})) = 1\right] \right| \ ,\end{aligned}$$

where C_0, C_1 are any two circuits defined on $\{0,1\}^{n(\lambda)}$ of the same functionality and size $s(\lambda)$. For every $\lambda \in \mathbb{N}$, $1 \leq i \leq n(\lambda)$, and size-s circuit C, we may view

$$r\mathcal{O}.\mathsf{Obf}(n, C, 1^{\tilde{\lambda}}) = \left(r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}}), z\right)$$

as consisting of an obfuscation $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ for inputs of size i along with additional information z (capturing the functional keys $\mathsf{FSK}_{i+1}, \ldots, \mathsf{FSK}_n$). We shall denote by $\zeta_i = \zeta_i(\tilde{\lambda})$ the length of such a z.

We now define $\delta_n(\lambda) := \delta_{i\mathcal{O}}(\lambda)$ and for $i < n(\lambda)$,

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

where C_0 and C_1 are any two circuits defined on $\{0,1\}^i$ of the same size and functionality and $z \in \{0,1\}^{\zeta_i}$.

Proposition 3.1. *If* PFE *is γ-secure, there exists a function $\delta_{\mathsf{PFE}}(\tilde{\lambda}) = \gamma(\tilde{\lambda})^{\Omega(1)}$ such that for any $\lambda \in \mathbb{N}$, $n = n(\lambda)$, $\tilde{\lambda} = \tilde{\lambda}(n, \lambda)$, $i \in \{2, \ldots, n\}$:*

- $\delta_i(\lambda) \leq 2^{i+1} \cdot (\delta_{i-1}(\lambda) + \delta_{\mathsf{PFE}}(\tilde{\lambda}))$.
- $\delta_1(\lambda) = 0$.

Before proving the proposition, we show that it concludes the security analysis:

Claim 3.6. $\delta_n \leq O(2^{n^2} \cdot \delta_{\mathsf{PFE}})$.

Proof. To prove the claim, we show by induction on i that

$$\delta_i \leq \delta_{\mathsf{PFE}} \sum_{j=1}^{i-1} \prod_{k=0}^{j-1} 2^{i+1-k} \ .$$

By Proposition 3.1, $\delta_1 = 0$ and thus satisfies the above. Assuming the above holds for $i-1 \geq 1$, and using the proposition again:

$$\begin{aligned}
\delta_i \leq & 2^{i+1}(\delta_{i-1} + \delta_{\mathsf{PFE}}) \leq \\
& 2^{i+1}\delta_{\mathsf{PFE}} + 2^{i+1}\delta_{\mathsf{PFE}} \sum_{j=1}^{i-2} \prod_{k=0}^{j-1} 2^{i-k} = \\
& 2^{i+1}\delta_{\mathsf{PFE}} + \delta_{\mathsf{PFE}} \sum_{j=1}^{i-2} \prod_{k=0}^{j} 2^{i+1-k} = \\
& 2^{i+1}\delta_{\mathsf{PFE}} + \delta_{\mathsf{PFE}} \sum_{j=2}^{i-1} \prod_{k=0}^{j-1} 2^{i+1-k} = \\
& \delta_{\mathsf{PFE}} \sum_{j=1}^{i-1} \prod_{k=0}^{j-1} 2^{i+1-k} \ .
\end{aligned}$$

We deduce accordingly

$$\delta_n \leq \delta_{\mathsf{PFE}} \sum_{j=1}^{n-1} \prod_{k=0}^{j-1} 2^{n+1-k} \leq \delta_{\mathsf{PFE}}(n-1)2^{\sum_{j=3}^{n+1} j} \leq O(\delta_{\mathsf{PFE}} 2^{n^2}) \ .$$

□

We now turn to prove the proposition.

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 and fix any auxiliary input $z \in \{0,1\}^{\zeta_i}$. We will bound

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

For this purpose, we consider the following sequence hybrid experiments.

$\mathcal{H}_b$ for $b \in \{0,1\}$: This hybrid corresponds to the (real) distribution where:

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

where $\widetilde{\mathcal{E}} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathcal{E}^\ell[\mathsf{EK}], 1^{\tilde{\lambda}})$, $(\mathsf{EK}, \mathsf{FSK}) \leftarrow \mathsf{PFE.Setup}(1^{\tilde{\lambda}}, C_b)$, and $\mathcal{E}^\ell[\mathsf{EK}](\mathbf{x})$ is a circuit that for any $\mathbf{x} \in \{0,1\}^{i-1}$ computes two encryptions

$$\mathcal{E}[\mathsf{EK}](\mathbf{x}) = \{\mathsf{PFE.Enc}(\mathsf{EK}, \mathbf{x}\beta)\}_{\beta\in\{0,1\}} \quad ,$$

padded to size $\ell(|\mathcal{E}[\mathsf{EK}]|)$.

$\mathcal{H}_b^\star$ for $b \in \{0,1\}$: This hybrid is identical to $\mathcal{H}_b$, except that we sample fake keys $(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^{\tilde{\lambda}}, C_0, C_1)$ and use $\mathsf{EK}_b^\star, \mathsf{FSK}^\star$ instead of EK and FSK.

$\mathcal{G}_{\mathbf{y}}^\star$ for $\mathbf{y} \in [2^i+1]$: This hybrid is identical to $\mathcal{H}_0^\star$, except that $\widetilde{\mathcal{E}}$ is an obfuscation of a hybrid circuit $\mathcal{E}_{\mathbf{y}}[\mathsf{EK}_0^\star, \mathsf{EK}_1^\star]$ instead of the circuit $\mathcal{E}[\mathsf{EK}_0^\star]$. The hybrid circuit computes two encryptions, but uses $\mathsf{EK}_0^\star$ for all plaintexts $\mathbf{x}\beta \geq \mathbf{y}$ and $\mathsf{EK}_1^\star$ for all plaintexts $\mathbf{x}\beta < \mathbf{y}$, where we naturally interpret $\mathbf{x}\beta$ as an integer in $[2^i]$ according to lexicographic order.

That is, for any $\mathbf{x} \in \{0,1\}^{i-1}$, the circuit $\mathcal{E}_{\mathbf{y}}[\mathsf{EK}_0^\star, \mathsf{EK}_1^\star]$ is defined by

$$\mathcal{E}_{\mathbf{y}}[\mathsf{EK}_0^\star, \mathsf{EK}_1^\star](\mathbf{x}) = \left\{ \begin{array}{ll} \mathsf{PFE.Enc}(\mathsf{EK}_0^\star, \mathbf{x}\beta), & \text{if } \mathbf{x}\beta \geq \mathbf{y}; \\ \mathsf{PFE.Enc}(\mathsf{EK}_1^\star, \mathbf{x}\beta), & \text{if } \mathbf{x}\beta < \mathbf{y}. \end{array} \right\}_{\beta\in\{0,1\}} ,$$

and is padded to size $\ell = \ell(|\mathcal{E}[\mathsf{EK}]|)$.

$\mathcal{G}_{\mathbf{y},b}^\star$ for $\mathbf{y} \in [2^i+1]$, $b \in \{0,1\}$: This hybrid is identical to $\mathcal{G}_{\mathbf{y}}^\star$, except that $\widetilde{\mathcal{E}}$ is an obfuscation of a hybrid circuit $\mathcal{E}_{\mathbf{y},b}^\ell[\mathsf{EK}_0^\star\{\mathbf{y}\}, \mathsf{EK}_1^\star\{\mathbf{y}\}, \mathsf{CT}_{\mathbf{y},b}]$ instead of the circuit $\mathcal{E}_{\mathbf{y}}[\mathsf{EK}_0^\star, \mathsf{EK}_1^\star]$. Here the keys $(\mathsf{EK}_0^\star\{\mathbf{y}\}, \mathsf{EK}_1^\star\{\mathbf{y}\}) = \mathsf{PFE.Punc}(\mathsf{EK}_0^\star, \mathsf{EK}_1^\star, \mathbf{y})$ are punctured at $\mathbf{y}$, and the ciphertext $\mathsf{CT}_{\mathbf{y},b} = \mathsf{PFE.Enc}(\mathsf{EK}_b^\star, \mathbf{y})$ is hardwired and consists of an encryption of $\mathbf{y}$ under $\mathsf{EK}_b^\star$. The circuit behaves similarly to $\mathcal{E}_{\mathbf{y}}[\mathsf{EK}_0^\star, \mathsf{EK}_1^\star]$, only that for all $\mathbf{y}' \neq \mathbf{y}$ it uses the punctured keys, whereas for $\mathbf{y}$ it outputs the hardwired ciphertext $\mathsf{CT}_{\mathbf{y},b}$.

That is, for any $\mathbf{x} \in \{0,1\}^{i-1}$, the circuit $\mathcal{E}_{\mathbf{y},b}^\ell[\mathsf{EK}_0^\star\{\mathbf{y}\}, \mathsf{EK}_1^\star\{\mathbf{y}\}, \mathsf{CT}_{\mathbf{y},b}]$ is defined by

$$\mathcal{E}_{\mathbf{y},b}[\mathsf{EK}_0^\star\{\mathbf{y}\}, \mathsf{EK}_1^\star\{\mathbf{y}\}, \mathsf{CT}_{\mathbf{y}}](\mathbf{x}) = \left\{ \begin{array}{ll} \mathsf{PFE.Enc}(\mathsf{EK}_0^\star\{\mathbf{y}\}, \mathbf{x}\beta), & \text{if } \mathbf{x}\beta > \mathbf{y}; \\ \mathsf{CT}_{\mathbf{y},b}, & \text{if } \mathbf{x}\beta = \mathbf{y}; \\ \mathsf{PFE.Enc}(\mathsf{EK}_1^\star\{\mathbf{y}\}, \mathbf{x}\beta), & \text{if } \mathbf{x}\beta < \mathbf{y}. \end{array} \right\}_{\beta\in\{0,1\}} ,$$

and is padded to size $\ell = \ell(|\mathcal{E}[\mathsf{EK}]|)$.

Claim 3.7. *Assuming that* PFE *is* γ*-secure, there exists function* $\delta_{\mathsf{PFE}}(\tilde{\lambda}) = \gamma(\tilde{\lambda})^{\Omega(1)}$ *such that for any* $\lambda \in \mathbb{N}$, $n = n(\lambda)$, $\tilde{\lambda} = \tilde{\lambda}(\lambda, n)$, *any* $i \in [n]$, *and any two equal-size circuits* $C_0, C_1 : \{0,1\}^i \to \{0,1\}^*$:

1. $\mathcal{H}_0 \approx_{\mathcal{D},\delta_{\mathsf{PFE}}} \mathcal{H}_0^\star$.
2. $\mathcal{H}_0^\star = \mathcal{G}_1^\star$.
3. $\mathcal{G}_{\mathbf{y}}^\star \approx_{\mathcal{D},\delta_{i-1}} \mathcal{G}_{\mathbf{y},0}^\star$, *for all* $\mathbf{y} \in [2^i+1]$.
4. $\mathcal{G}_{\mathbf{y},0}^\star \approx_{\mathcal{D},\delta_{\mathsf{PFE}}} \mathcal{G}_{\mathbf{y},1}^\star$, *for all* $\mathbf{y} \in [2^i+1]$.
5. $\mathcal{G}_{\mathbf{y},1}^\star \approx_{\mathcal{D},\delta_{i-1}} \mathcal{G}_{\mathbf{y}+1}^\star$, *for all* $\mathbf{y} \in [2^i]$.
6. $\mathcal{G}_{2^i+1}^\star = \mathcal{H}_1^\star$.
7. $\mathcal{H}_1^\star \approx_{\mathcal{D},\delta_{\mathsf{PFE}}} \mathcal{H}_1$.

Before proving the claim, we show that it concludes the proof of Proposition 3.1:

$$\begin{aligned}
\delta_i &= \left|\Pr\left[\mathcal{D}(\mathcal{H}_0)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}_1)=1\right]\right| = \\
&\quad \left|\Pr\left[\mathcal{D}(\mathcal{H}^\star_0)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}^\star_1)=1\right]\right| + \sum_{b\in\{0,1\}}\left|\Pr\left[\mathcal{D}(\mathcal{H}_b)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}^\star_b)=1\right]\right| = \\
&\quad \left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_1)=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{2^i})=1\right]\right| + \sum_{b\in\{0,1\}}\left|\Pr\left[\mathcal{D}(\mathcal{H}_b)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}^\star_b)=1\right]\right| \leq \\
&\sum_{\mathbf{y}\in[2^i]}\left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y}})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y}+1})=1\right]\right| + \sum_{b\in\{0,1\}}\left|\Pr\left[\mathcal{D}(\mathcal{H}_b)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}^\star_b)=1\right]\right| \leq \\
&\sum_{\mathbf{y}\in[2^i]}\Big(\left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y}})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y},0})=1\right]\right| + \\
&\quad \left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y},0})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y},1})=1\right]\right| + \\
&\quad \left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y},1})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y}+1})=1\right]\right|\Big) + \sum_{b\in\{0,1\}}\left|\Pr\left[\mathcal{D}(\mathcal{H}_b)=1\right]-\Pr\left[\mathcal{D}(\mathcal{H}^\star_b)=1\right]\right| \leq \\
&\quad 2^i(2\delta_{i-1}+\delta_{\mathsf{PFE}})+2\delta_{\mathsf{PFE}} = 2^{i+1}\delta_{i-1}+(2^i+2)\delta_{\mathsf{PFE}} \leq 2^{i+1}\left(\delta_{i-1}+\delta_{\mathsf{PFE}}\right) \ .
\end{aligned}$$

Proof of Claim 3.7. The indistinguishability between the hybrids is established by applying the γ-security of the underlying puncturable functional encryption PFE and the bound δ_{i-1} on the distinguishing gap for circuits on $i-1$ input bits.

$\mathcal{H}_b \approx \mathcal{H}^\star_b$ for $b\in\{0,1\}$ (items 1,7): Here the difference between the hybrids is that in $\mathcal{H}_b$ we sample real keys $(\mathsf{EK},\mathsf{FSK}) \leftarrow \mathsf{PFE.Setup}(1^{\tilde{\lambda}}, C_b)$ whereas in $\mathcal{H}^\star_b$ we sample fake keys $(\mathsf{EK}^\star_0,\mathsf{EK}^\star_1,\mathsf{FSK}^\star) \leftarrow \mathsf{PFE.Setup}^\star(1^{\tilde{\lambda}}, C_0, C_1)$ and use $\mathsf{EK}^\star_b$. By the key indistinguishability of PFE:

$$\mathsf{EK},\mathsf{FSK} \approx_\gamma \mathsf{EK}^\star_b,\mathsf{FSK}^\star \ .$$

Thus,

$$\mathcal{H}_b \approx_{\mathcal{D},\gamma^\alpha} \mathcal{H}^\star_b \ ,$$

for some constant $\alpha \leq 1$ that depends only on $\mathcal{D}$.

$\mathcal{H}^\star_0 = \mathcal{G}^\star_1$, $\mathcal{G}^\star_{2^i+1} = \mathcal{H}^\star_1$ (items 2,6): The equality between the hybrids follows by their definition. Indeed, in $\mathcal{H}^\star_0$ we always use $\mathsf{EK}^\star_0$ as in $\mathcal{G}^\star_1$ and in $\mathcal{H}^\star_1$ we always use $\mathsf{EK}^\star_1$ as in $\mathcal{G}^\star_{2^i+1}$.

$\mathcal{G}^\star_{\mathbf{y},b} \approx \mathcal{G}^\star_{\mathbf{y}+b}$ for $b\in\{0,1\}$ (items 3,5): Here the difference between $\mathcal{G}^\star_{\mathbf{y}+b}$ and $\mathcal{G}^\star_{\mathbf{y},b}$ is in the obfuscated circuit $\widetilde{\mathcal{E}}$. In the first, $\widetilde{\mathcal{E}} = r\mathcal{O}.\mathsf{Obf}(i-1,\mathcal{E}^\ell_{\mathbf{y}+b}[\mathsf{EK}^\star_0,\mathsf{EK}^\star_1],1^{\tilde{\lambda}})$, whereas in the second, $\widetilde{\mathcal{E}} = r\mathcal{O}.\mathsf{Obf}(i-1,\mathcal{E}^\ell_{\mathbf{y}+b,b}[\mathsf{EK}^\star_0,\mathsf{EK}^\star_1,\mathsf{CT}_{\mathbf{y}+b,b}],1^{\tilde{\lambda}})$.

The two circuits compute the exact same function, but in two different ways:

- The first circuit uses the keys $(\mathsf{EK}^\star_0,\mathsf{EK}^\star_1)$ to encrypt all plaintexts $\mathbf{x}\beta \neq \mathbf{y}+b$, whereas the second circuit uses the punctured keys $(\mathsf{EK}^\star_0\{\mathbf{y}+b\},\mathsf{EK}^\star_1\{\mathbf{y}+b\})$. By punctured-key correctness of PFE, the two result in the same ciphertexts.
- For the plaintext $\mathbf{y}+b$, the first circuit computes $\mathsf{CT}_{\mathbf{y}+b,b} = \mathsf{PFE.Enc}(\mathsf{EK}_b,\mathbf{y}+b)$ on its own using the key $\mathsf{EK}^\star_b$, whereas the second circuit uses a hardwired $\mathsf{CT}_{\mathbf{y}+b,b}$.

The two circuits are padded to the same size $\ell = \ell(|\mathcal{E}[\mathsf{EK}]|)$. We can thus bound

$$\begin{aligned}
&\left|\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y}+b})=1\right]-\Pr\left[\mathcal{D}(\mathcal{G}^\star_{\mathbf{y},b})=1\right]\right| = \\
&\left|\Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i-1,\mathcal{E}_{\mathbf{y}+b},1^{\tilde{\lambda}}),\mathsf{FSK}^\star,z)=1\right]-\Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i-1,\mathcal{E}_{\mathbf{y},b},1^{\tilde{\lambda}}),\mathsf{FSK}^\star,z)=1\right]\right| \leq \\
&\max_{C'_0,C'_1,z'}\left|\Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i-1,C'_0,1^{\tilde{\lambda}}),z')=1\right]-\Pr\left[\mathcal{D}(r\mathcal{O}.\mathsf{Obf}(i-1,C'_1,1^{\tilde{\lambda}}),z')=1\right]\right| = \delta_{i-1} \ ,
\end{aligned}$$

where above C_0 and C_1 are any two circuits defined on $\{0,1\}^{i-1}$ of the same functionality and size ℓ and we view $(z, \mathsf{FSK}^\star)$ as $z' \in \{0,1\}^{\zeta_{i-1}}$.

It follows that

$$\mathcal{G}^\star_{\mathbf{y}+b} \approx_{\mathcal{D},\delta_{i-1}} \mathcal{G}^\star_{\mathbf{y},b} \ .$$

$\mathcal{G}^\star_{\mathbf{y},0} \approx \mathcal{G}^\star_{\mathbf{y},1}$ (item 4): Here the difference between the hybrids is in the hardwired ciphertext $\overline{\mathsf{CT}_{\mathbf{y},b} = \mathsf{PFE.Enc}(\mathsf{EK}_b, \mathbf{y})}$, where $b = 0$ in the first hybrid and $b = 1$ in the second. By indistinguishability at punctured points of PFE:

$$\mathsf{EK}^\star_0\{\mathbf{y}\}, \mathsf{EK}^\star_1\{\mathbf{y}\}, \mathsf{FSK}^\star, \mathsf{CT}_{\mathbf{y},0} \approx_\delta \mathsf{EK}^\star_0\{\mathbf{y}\}, \mathsf{EK}^\star_1\{\mathbf{y}\}, \mathsf{FSK}^\star, \mathsf{CT}_{\mathbf{y},1} \ .$$

Thus,

$$\mathcal{G}^\star_{\mathbf{y},0} \approx_{\gamma^{\alpha'}} \mathcal{G}^\star_{\mathbf{y},1} \ .$$

for some constant $\alpha' \leq 1$ that depends only on $\mathcal{D}$.

The function δ_{PFE}. The function $\delta_{\mathsf{PFE}}(\tilde{\lambda}) = \gamma(\tilde{\lambda})^{\min\{\alpha,\alpha'\}}$ thus satisfies the requirement of the Claim 3.7. This concludes the proof of the claim. □

The padding parameter. The padding parameter $\ell(|\mathcal{E}|)$ is chosen to account for the maximal-size circuit among the circuits $\mathcal{E}, \mathcal{E}_{\mathbf{y}}, \mathcal{E}_{\mathbf{y},b}$ considered in the above hybrids. Observe that the size of the circuits $\mathcal{E}_{\mathbf{y}}, \mathcal{E}_{\mathbf{y},b}$ is indeed linear in the size of the (real or fake) encryption circuits

$$|\mathsf{PFE.Enc}(\mathsf{EK}^\star_b, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}^\star_b\{\mathbf{y}\}, \cdot)| = |\mathsf{PFE.Enc}(\mathsf{EK}, \cdot) = |\mathcal{E}| \ .$$

This concludes the proof of Proposition 3.1 and the proof of security. □

Efficiency. We now analyze the efficiency of the obfuscator relying on succinct encryption of the underlying PFE. Assume that PFE is succinct. That is, there exists a polynomial Φ_{PFE} and a constant $0 < \varepsilon \leq 1$ such that the size of the corresponding encryption circuit is bounded by

$$s^{1-\varepsilon} \cdot \Phi_{\mathsf{PFE}}(n, \lambda) \ ,$$

where n, s are the input-size and circuit-size of the functions chosen during the (possibly fake) setup phase.

We show that for any polynomial $\tilde{\lambda}(n, \lambda) = \mathrm{poly}(n, \lambda)$, there exists a fixed polynomial $\Phi_{i\mathcal{O}}$ such that $i\mathcal{O}_{\tilde{\lambda}}(C, 1^\lambda)$ runs in time $\Phi_{i\mathcal{O}}(|C|, \lambda)$. First, observe that the recursive $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ only invokes the efficient PFE algorithms, and thus there exists a polynomial Φ_r such that for all $\lambda, n \in \mathbb{N}, i \in [n]$, the running time of $r\mathcal{O}.\mathsf{Obf}(i, C_i, 1^{\tilde{\lambda}})$ is bounded by $\Phi_r(|C_i|, \tilde{\lambda}|)$.

We prove the following:

Claim 3.8. *Let $\ell(s) = c \cdot s$ be the padding function and let $\Phi_{\mathsf{PFE}} = \Phi_{\mathsf{PFE}}(n, \tilde{\lambda})$. Then, for all $i \in [n]$,*

$$|C_i| \leq |C| \cdot (c\Phi_{\mathsf{PFE}})^{1/\varepsilon} \ .$$

Before proving the claim, we show that it concludes the efficiency analysis. Indeed, it implies that the total running time of the obfuscator is bounded by a fixed polynomial

$$\Phi_{i\mathcal{O}}(|C|, \lambda) \leq n \cdot \max_{i \in [n]} \Phi_r(|C_i|, \tilde{\lambda}(n, \lambda)|) \leq |C| \cdot \Phi_r(|C| \cdot ((c\Phi_{\mathsf{PFE}})^{1/\varepsilon}, \tilde{\lambda}(|C|, \lambda)) \ .$$

Proof of Claim 3.8. We prove by induction on i that

$$|C_{n-i}| \leq |C| \cdot (c\Phi_{\mathsf{PFE}})^{\sum_{j=0}^{i-1}(1-\varepsilon)^j} \quad .$$

This concludes the proof since $\sum_{j=0}^{\infty}(1-\varepsilon)^j = \varepsilon^{-1}$.

For $i = 0$, $C_n = C$ and the bound holds.

Assume the bounds hold for $i > 0$. By the induction hypothesis, the succinctness of PFE, and the definition of $r\mathcal{O}.\mathsf{Obf}$, it holds that for any $i > 0$,

$$\begin{aligned}|C_{n-i}| = |\mathcal{E}_i^{\ell}| = \ell(|\mathcal{E}_i|) \leq c \cdot |C_{n-i+1}|^{1-\varepsilon}\Phi_{\mathsf{PFE}} \leq \\ |C|^{1-\varepsilon}(c\Phi_{\mathsf{PFE}})^{(1-\varepsilon)\sum_{j=0}^{i-2}(1-\varepsilon)^j} c\Phi_{\mathsf{PFE}} \leq \\ |C| \cdot (c\Phi_{\mathsf{PFE}})^{\sum_{j=0}^{i-1}(1-\varepsilon)^j} \quad .\end{aligned}$$

□

This concludes the proof of Theorem 3.3. □

3.3 IO with Linear Overhead

In this section, we observe that a variant of our construction, combined with known results from the literature, implies that any IO scheme can be turned into an IO scheme where the obfuscation only grows linearly in the size of the circuit. Specifically, Concretely, under the Learning with Errors (LWE) Assumption, we show that any IO can be transformed into one where an obfuscation of a size-s, depth-d circuit, with input length n, is of size $2s + \mathrm{poly}(n, d, \lambda)$.

This is achieved by relying on a single application of the recursive obfuscator (instead of n applications) and instantiating our puncturable functional encryption with a public-key functional encryption scheme with succinct keys as the one constructed by Boneh et al. [BGG+14]:

Proposition 3.2 (FE with succinct keys [BGG+14])**.** *Assuming subexponential LWE, there exists a subexponentially-secure, single-key, public-key, functional encryption scheme, where there exists a polynomial Φ_{FE} such that for any depth-d circuit $C : \{0,1\}^n \to \{0,1\}^m$, letting $(\mathsf{PK}, \mathsf{FSK}_C) \leftarrow \mathsf{FE.Setup}(1^\lambda, C)$, it holds that:*

- $\mathsf{FSK}_C = (C, \mathsf{fsk})$.
- *The key* fsk *and the encryption circuit* $\mathsf{FE.Enc}(\mathsf{PK}, \cdot)$ *are of size at most* $\Phi_{\mathsf{FE}}(n, m, d, \lambda)$.

We prove:

Corollary 3.9. *Assuming subexponential LWE and subexponentially-secure IO, there exists IO where there exists a polynomial Φ, such that given any size-s, depth-d circuit $C : \{0,1\}^n \to \{0,1\}^m$, a corresponding obfuscation is of size $2s + \Phi(n, m, d, \lambda)$.*

Proof. First, we observe that by plugging-in the FE scheme given by Proposition 3.2 into the construction from Section 3.1, we obtain a puncturable FE that inherits the succinctness of the underlying FE scheme up to a mild loss.

Claim 3.10. *Assuming subexponential LWE, there exists a subexponentially-secure puncturable functional encryption scheme, where there exists a polynomial Φ_{PFE} such that for any depth-d circuit $C : \{0,1\}^n \to \{0,1\}^m$, letting $(\mathsf{EK}, \mathsf{FSK}_C) \leftarrow \mathsf{PFE.Setup}(1^\lambda, C)$, it holds that:*

- FSK_C *is of size at most* $2|C| + \Phi_{\mathsf{PFE}}(n, m, d, \lambda)$.

- *The encryption circuit* $\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)$ *is of size at most* $\Phi_{\mathsf{PFE}}(n, m, d, \lambda)$.

Proof. Let FE be the scheme given by Proposition 3.2 and let $\mathsf{PFE}^{\mathsf{FE}}$ be the puncturable functional encryption scheme from Section 3.1 when instantiated with FE. In $\mathsf{PFE}^{\mathsf{FE}}$, a functional key FSK for a circuit C consists of a functional key FSK_F generated under FE for a circuit $F = F[\mathsf{CT}_0, \mathsf{CT}_1]$ that is parameterized by two symmetric-key encryptions $\mathsf{CT}_0, \mathsf{CT}_1$ of C. The circuit F can be efficiently constructed given only $\mathsf{CT}_0, \mathsf{CT}_1$. Also, recall that in the given FE scheme, FSK has the form (F, fsk) where fsk is of size $\Phi(n, m, d, \lambda)$.

We now consider a slight variant of $\mathsf{PFE}^{\mathsf{FE}}$, which we shall denote by

$$\widehat{\mathsf{PFE}}^{\mathsf{FE}} = (\mathsf{PFE.Setup}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec}, \mathsf{PFE.Setup}^\star, \mathsf{PFE.Punc}) \ ,$$

where the functional key $\widehat{\mathsf{FSK}}$ for C consists of $(\mathsf{CT}_0, \mathsf{CT}_1, \mathsf{fsk})$ and the functional decryption process $\mathsf{PFE.Dec}(\widehat{\mathsf{FSK}}, \mathsf{CT})$ first parses $\widehat{\mathsf{FSK}} = (\mathsf{CT}_0, \mathsf{CT}_1, \mathsf{fsk})$, then computes the secret key $\mathsf{FSK} = (F[\mathsf{CT}_0, \mathsf{CT}_1], \mathsf{fsk})$, and finally applies $\mathsf{FE.Dec}(\mathsf{FSK}, \mathsf{CT})$.

Correctness and security. First, observe that the scheme $\widehat{\mathsf{PFE}}^{\mathsf{FE}}$ satisfies the same functionality and security properties of $\mathsf{PFE}^{\mathsf{FE}}$ (proven in Section 3.1). Indeed, the two schemes only defer in the format of functional keys and a functional key FSK under $\mathsf{PFE}^{\mathsf{FE}}$ can be efficiently simulated from a functional key $\widehat{\mathsf{FSK}}$ under $\widehat{\mathsf{PFE}}^{\mathsf{FE}}$.

Function-key succinctness. We now analyze the succinctness of $\widehat{\mathsf{PFE}}^{\mathsf{FE}}$. By the above construction, the size of a functional key for a circuit C is bounded by

$$|\widehat{\mathsf{FSK}}| = 2|\mathsf{CT}_0| + |\mathsf{fsk}| = 2|C| + \Phi_{\mathsf{FE}}(n, m, d, \lambda) \ ,$$

where we use the fact that in (one-time) symmetric-key encryption schemes, the ciphertext-size equals the message-size (Fact 2.7).

Encryption succinctness. The difference from the succinctness analysis in Section 3.1 is that there the size of the encryption circuit of FE grows (sublinearly) with circuit-size, whereas here it does not grow with the circuit size, but does grow (polynomially) with the output length and depth.

The encryption circuit in $\widehat{\mathsf{PFE}}^{\mathsf{FE}}$ has the form

$$\mathsf{PFE.Enc}(\mathsf{EK}, \cdot) = \mathsf{FE.Enc}(\mathsf{PK}, (\cdot, \mathsf{SK}, \beta); \mathsf{PRF}_{\mathsf{K}}(\cdot)) \ .$$

Observe that

- $\mathsf{PRF}_{\mathsf{K}}(\cdot)$, for messages of size n, can be computed by a circuit of size $\Phi_{\mathsf{PRF}}(n, \lambda)$ for a fixed polynomial Φ_{PRF}.
- Given $r = \mathsf{PRF}_{\mathsf{K}}(m)$ as input, $\mathsf{FE.Enc}(\mathsf{PK}, (m, \mathsf{SK}, \beta); r)$ can be computed by a circuit of size $\Phi_{\mathsf{FE}}(n', m', d', \lambda)$, where in PFE, messages m of size n translate to messages (m, SK, β) of size n' and circuits $f : \{0,1\}^n \to \{0,1\}^m$ of size s and depth d translate to circuits $F : \{0,1\}^{n'} \to \{0,1\}^{m'}$ of size s' and depth d' in FE.

Observe that $n' = n + \lambda + 1$ and $m' = m$. It is left to bound s' (here, it suffices that $s' = s^{O(1)}$) and d'. F chooses CT_β according to β and applies $U(\mathsf{Sym.Dec}(\mathsf{SK}_\beta, \mathsf{CT}_\beta), m)$. By Fact 2.7, each CT_β encrypting f is of size s and applying $\mathsf{Sym.Dec}$ requires size $s \cdot \Phi_{\mathsf{Sym}}(\lambda)$ and depth $\Phi_{\mathsf{Sym}}(\lambda)$ for a fixed polynomial Φ_{Sym}. Applying a depth-universal circuit for an input circuit of size s

and depth d requires size $s^3 \cdot \Phi_U(\log s) \leq s^3 \cdot \Phi_U(\lambda)$ and depth $c_U \cdot d$ for a fixed polynomial Φ_U and constant c_U [CH85]. Thus overall, there exists a polynomial Φ such that

$$s' \leq \Phi(s, \lambda), \qquad d' \leq \Phi(d, \lambda) \ ,$$

for a fixed polynomial Φ that aggregates the above.

In conclusion, we can bound the size of the encryption circuit $\mathsf{PFE.Enc}(\mathsf{EK}, \cdot)$ by

$$\Phi_{\mathsf{PRF}}(n, \lambda) + \Phi_{\mathsf{FE}}(n', m', d', \lambda) \leq \Phi_{\mathsf{PRF}}(n, \lambda) + \Phi_{\mathsf{FE}}(n + \lambda + 1, m, \Phi(d, \lambda), \lambda) \ ,$$

establishing our requirement with $\Phi_{\mathsf{PFE}}(n, m, d, \lambda) := \Phi_{\mathsf{PRF}}(n, \lambda) + \Phi_{\mathsf{FE}}(n + \lambda + 1, m, \Phi(d, \lambda), \lambda)$. □

We next show that any $2^{\tilde{\lambda}^\alpha}$-secure indistinguishability obfuscator $i\mathcal{O}$ can be combined with a $2^{\tilde{\lambda}^\alpha}$-secure puncturable functional encryption $\widehat{\mathsf{PFE}}$ as the one given by Claim 3.10 (where in both $\tilde{\lambda}$ is the security parameter) to obtain a new indistinguishabiliy obfuscator $i\mathcal{O}'$ where the size of an obfuscation of a size s depth d circuit $C : \{0,1\}^n \to \{0,1\}^m$ is of size $2s + \Phi(n, m, d, \lambda)$ for a fixed polynomial Φ.

The construction. We consider a single iteration of the recursive transformation from Section 3.2. That is, we define

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

where $r\mathcal{O}.\mathsf{Obf}(n, \cdot, 1^{\tilde{\lambda}})$ is defined as in Section 3.1 and is instantiated with $\widehat{\mathsf{PFE}}$ (from Claim 3.10), and redefine

$$r\mathcal{O}.\mathsf{Obf}(n-1, \cdot, 1^{\tilde{\lambda}}) = i\mathcal{O}(\cdot, 1^{\tilde{\lambda}}) \ .$$

In addition, we define $\tilde{\lambda}(\lambda, n) = \omega(n + \log \lambda)^{\alpha^{-1}}$.

The evaluation procedure is changed accordingly. That is, given an obfuscation $\widetilde{C}$, we define

$$i\mathcal{O}'.\mathsf{Eval}(\widetilde{C}, \mathbf{x}) = r\mathcal{O}.\mathsf{Eval}(n, \widetilde{C}, \mathbf{x}) \ ,$$

where $r\mathcal{O}.\mathsf{Eval}(n, \widetilde{C}, \mathbf{x})$ is defined as in Section 3.1 and is instantiated with $\widehat{\mathsf{PFE}}$ (from Claim 3.10), and redefine

$$r\mathcal{O}.\mathsf{Eval}(n-1, \widetilde{E}, \mathbf{x}_{n-1}) = i\mathcal{O}.\mathsf{Eval}(\widetilde{E}, \mathbf{x}_{n-1}) \ .$$

The correctness of the scheme follows readily, we focus on succinctness and security.

Efficiency and succinctness. In the new scheme, an obfuscation

$$i\mathcal{O}'(C, 1^\lambda) = (\widetilde{\mathcal{E}}, \widehat{\mathsf{FSK}})$$

consists of a functional key $\widehat{\mathsf{FSK}}$ and an obfuscation $\widetilde{\mathcal{E}} \leftarrow i\mathcal{O}(\mathcal{E}^\ell, 1^{\tilde{\lambda}})$ of the circuit $\mathcal{E}(\cdot) = \mathsf{PFE.Enc}(\mathsf{EK}, \cdot)$ padded to size $\ell(|\mathcal{E}|)$ where $(\mathsf{EK}, \widehat{\mathsf{FSK}}) \leftarrow \mathsf{PFE.Setup}^\star(1^\lambda, C)$ and $\ell(s) = c \cdot s$, for an absolute constant c.

By the succinctness of PFE and the efficiency of $i\mathcal{O}$:

$$\begin{aligned} |i\mathcal{O}'(C, 1^\lambda)| = |\widehat{\mathsf{FSK}}| + |i\mathcal{O}(\mathcal{E}^\ell, 1^{\tilde{\lambda}})| \leq \\ 2|C| + \Phi_{\mathsf{PFE}}(n, m, d, \tilde{\lambda}) + \Phi_{i\mathcal{O}}(|\mathcal{E}^\ell|) \leq \\ 2|C| + \Phi_{\mathsf{PFE}}(n, m, d, \tilde{\lambda}) + \Phi_{i\mathcal{O}}(c \cdot \Phi_{\mathsf{PFE}}(n, m, d, \tilde{\lambda})) \ , \end{aligned}$$

where $\Phi_{i\mathcal{O}}$ is a fixed polynomial that bounds the running time of $i\mathcal{O}$.

This establishes the succinctness requirement given by Corollary 3.4, with

$$\Phi(n, m, d, \lambda) := \Phi_{\mathsf{PFE}}(n, m, d, \tilde{\lambda}(n, \lambda)) + \Phi_{i\mathcal{O}}(c \cdot \Phi_{\mathsf{PFE}}(n, m, d, \tilde{\lambda}(n, \lambda))) \ .$$

The fact that the obfuscator runs in polynomial time follows readily from the fact that all underlying algorithms run in polynomial time and are applied at most once.

Security. The security of the scheme follows from the security proof of the recursive scheme in Section 3.2. Specifically, for any polynomial-size distinguisher $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$, we define $\delta_n(\lambda)$ and $\delta_{n-1}(\lambda)$ as in Section 3.2. (The first is the maximal distinguishing advantage of $\mathcal{D}$ against the constructed obfuscator $i\mathcal{O}'$ and the second is against $r\mathcal{O}.\mathsf{Obf}(n-1, \cdot, 1^{\tilde{\lambda}})$, which in our case is the underlying obfuscator $i\mathcal{O}(\cdot, 1^{\tilde{\lambda}})$).

By Proposition 3.1, the subexponential-security of PFE and $i\mathcal{O}$, and our choice of $\tilde{\lambda}(n, \lambda)$,

$$\begin{aligned} \delta_n(\lambda) &\leq 2^{n+1} \cdot (\delta_{n-1}(\lambda) + \delta_{\mathsf{PFE}}(\tilde{\lambda})) \leq \\ &2^{n+1} \cdot \left(2^{\Omega(-\tilde{\lambda}^\alpha)} + 2^{\Omega(-\tilde{\lambda}^\alpha)}\right) \leq 2^{-\omega(\log \lambda)} \ . \end{aligned}$$

□

4 A Bootstrapping Theorem

In this section, we show how to transform any *multi-key* functional encryption scheme with certain weak succinctness into a succinct single-key functional encryption scheme (which in particular is suitable for our FE to IO transformation from Section 3).

The notion of multi-key functional encryption schemes that we consider is a natural generalization of the single-key notion (Section 2.2), where security is guaranteed even in the presence of multiple functional keys. In the literature this notion is commonly referred to as *collusion resistance*. We in fact consider a definition that is somewhat weaker than the typical definition in the literature where all functions are known at setup time. (Note that since we aim to construct succinct single-key functional encryption from multi-key functional encryption, considering such a weaker notion only strengthens the result.)

In terms of succinctness, the common requirement for such schemes is that the complexity of encryption may grow with the circuit-size of functions, but not with the overall number of keys generated. Here, we will even allow sublinear dependence on the number of functions (we call this requirement *weak size-succinctness*.) We show that any such scheme can be transformed to a succinct single-key scheme.

The transformation can be applied to functional encryption schemes from the literature, such as the one by Garg, Gentry, Halevi and Zhandry [GGHZ16]. This gives rise to an IO construction based on a subexponential variant of the assumptions in [GGHZ16] on multilinear graded encodings.

We further show that the same transformation can be applied to transform any succinct single/multi-key scheme where encryption complexity scales (polynomially/exponentially) with *circuit depth* into a succinct scheme with no dependence on the depth (beyond the sublinear dependence on circuit size).

We start with the relevant definition and then proceed to the transformation.

4.1 Multi-Key FE with Succinct Encryption

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

- $\mathsf{MFE.Setup}(1^\lambda, f_1, \ldots, f_\ell)$: takes as input a security parameter λ in unary and ℓ functions $f_1, \ldots, f_\ell \in \mathcal{F}$ and outputs a public key PK and functional keys $\mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}$.
- $\mathsf{MFE.Enc}(\mathsf{PK}, m)$: takes as input a public key PK and 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 $\mathsf{MFE.Enc}(\mathsf{PK}, m; r)$.
- $\mathsf{MFE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: 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.

Definition 4.1 (Multi-key, selectively-secure, public-key FE with succinct encryption)**.** *A tuple of PPT algorithms* $\mathsf{MFE} = (\mathsf{MFE.Setup}, \mathsf{MFE.Enc}, \mathsf{MFE.Dec})$ *is a multi-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* $\lambda, n, \ell \in \mathbb{N}$*, message* $m \in \{0,1\}^n$*, functions* $\boldsymbol{f} = (f_1, \ldots, f_\ell) \in \mathcal{F}^\ell$*, with domain* $\{0,1\}^n$*, and every* $i \in [\ell]$*,*

$$\Pr\left[f_i(m) \leftarrow \mathsf{MFE.Dec}(\mathsf{FSK}_{f_i}, \mathsf{CT}) \;\middle|\; \begin{array}{c} (\mathsf{PK}, \mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}) \leftarrow \mathsf{MFE.Setup}(1^\lambda, \boldsymbol{f}) \\ \mathsf{CT} \leftarrow \mathsf{MFE.Enc}(\mathsf{PK}, m) \end{array} \right] = 1 \ .$$

2. **Selective security:** *for any polynomial-size adversary* $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$*, there exists a negligible function* $\mu(\lambda)$ *such that for any* $\lambda, n, \ell \in \mathbb{N}$*, any* $m_0, m_1 \in \{0,1\}^n$*, and functions* $\boldsymbol{f} = (f_1, \ldots, f_\ell) \in \mathcal{F}^\ell$ *such that* $\boldsymbol{f}(m_0) = \boldsymbol{f}(m_1)$*,*

$$\mathsf{PK}, \mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}, \mathsf{MFE.Enc}(\mathsf{PK}, m_0) \approx_{\mathcal{A},\mu} \mathsf{PK}, \mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}, \mathsf{MFE.Enc}(\mathsf{PK}, m_1) \ ,$$

   where $(\mathsf{PK}, \mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}) \leftarrow \mathsf{MFE.Setup}(1^\lambda, \boldsymbol{f})$*.*

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

3. **Succinct encryption circuit:** *there exists a polynomial* Φ *and a constant* $0 < \varepsilon \leq 1$*, such that for any input-size* n*, circuit-size* s*, depth* d *and* s*-size,* d*-depth functions* $\boldsymbol{f} = (f_1, \ldots, f_\ell) : \{0,1\}^n \to \{0,1\}^*$*,*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq (\ell \cdot s)^{1-\varepsilon} \cdot \Phi(n, \lambda) \ ,$$

   where $(\mathsf{PK}, \mathsf{FSK}_{f_1}, \ldots, \mathsf{FSK}_{f_\ell}) \leftarrow \mathsf{MFE.Setup}(1^\lambda, \boldsymbol{f})$*.*

   - *Encryption is* **fully succinct** *if* $\varepsilon = 1$*.*
   - *Encryption is* **weakly depth-succinct** *if*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq (\ell \cdot s)^{1-\varepsilon} \cdot \Phi(n, d, \lambda) \ .$$

   - *Encryption is* **very weakly depth-succinct** *if*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq (\ell \cdot s)^{1-\varepsilon} \cdot \Phi(n, 2^d, \lambda) \ .$$

   - *Encryption is* **weakly size-succinct** *if*

$$|\mathsf{MFE.Enc}(\mathsf{PK}, \cdot)| \leq \ell^{1-\varepsilon} \cdot \Phi(s, \lambda) \ .$$

We first observe that for all notions of succinctness, but weak size-succinctness, single-key and multi-key FE are equivalent (essentially tautological).

Claim 4.2. *The notions of (δ-secure) single-key FE (Definition 2.1) and multi-key FE (Definition 2.1) are equivalent with respect to succinctness/weak-depth-succinctness/very-weak-depth-succinctness.*

Proof. First, note that any multi-key FE is a single-key FE where $\ell = 1$, with respect to the above succinctness notions. To see that any single-key FE FE implies a multi-key FE MFE, consider the MFE scheme where any tuple of size-s, depth-d functions $f_1, \ldots, f_\ell : \{0,1\}^n \to \{0,1\}^m$ is parsed as a size-$s \cdot \ell$, depth-d single function $\boldsymbol{f} : \{0,1\}^n \to \{0,1\}^{m \times \ell}$.

We then define:

- $\mathsf{MFE.Setup}(1^\lambda, f_1, \ldots, f_\ell) = \mathsf{PK}, (\mathsf{FSK}_{\boldsymbol{f}}, 1), \ldots, (\mathsf{FSK}_{\boldsymbol{f}}, \ell)$, where $\mathsf{PK}, \mathsf{FSK}_{\boldsymbol{f}} \leftarrow \mathsf{FE.Setup}(1^\lambda, \boldsymbol{f})$.
- $\mathsf{MFE.Enc}(\mathsf{PK}, m) = \mathsf{FE.Enc}(\mathsf{PK}, m)$.
- $\mathsf{MFE.Dec}((\mathsf{FSK}, i), \mathsf{CT}) = m_i$, where $m_1 \ldots m_\ell = \mathsf{FE.Dec}(\mathsf{FSK}, \mathsf{CT})$.

The functionality, security, and succinctness/weak-depth-succinctness/very-weak-depth-succinctness of MFE follow readily from those of FE. □

In contrast to the above, *weak size-succinctness* is only meaningful for $\ell \gg 1$. For $\ell = 1$, it corresponds to a single-key scheme with no succinctness at all. Indeed, the notion of full collusion-resistance common in the literature (for instance, in [GGHZ16]) only implies weak size-succinctness.

From weak succinctness to succinctness. We now prove a bootstrapping theorem that shows that any multi-key FE scheme with (one of several forms of) weak succinctness implies succinct single-key FE (and also succinct multi-key FE by Claim 4.2).

The transformation is similar in spirit to previous randomized-encoding-based bootstrapping schemes from the literature [GVW12, ABSV15]. Roughly speaking, we rely on the fact that any function $f(x)$ represented by an arbitrary circuit has a randomized encoding $\hat{f}(x; r)$ that can be *decomposed* into multiple functions $\left\{\hat{f}_i(x; r)\right\}$ of lower complexity, both in terms of circuit-size and circuit-depth. Such an encoding can be efficiently decoded to $f(x)$ and reveals nothing on x but the result $f(x)$. This intuitively means that in order to generate a functional key for any function f it is sufficient to generate functional keys for the encoding functions $\left\{\hat{f}_i\right\}$, while including the randomness required for encoding as part of the encrypted message. In particular, if the complexity of encryption depends on the complexity of functions, we have now managed to reduce it.

The actual construction is somewhat more complicated in order to facilitate the security proof. Here also we rely on a common proof technique based on *the trapdoor paradigm* [FS89, CIJ+13, ABSV15] (sometimes termed *the Trojan method*).

4.2 The Transformation

We turn to describe the actual transformation.

Ingredients. We rely on the following primitives:

- A multi-key, selectively-secure, public-key functional encryption scheme MFE for all circuits.
- A one-time symmetric encryption scheme Sym.

- A decomposable randomized encoding scheme RE.

The constructed scheme FE consists of algorithms

$$(\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$$

described in Figure 5.

FE

- $\mathsf{FE.Setup}(1^\lambda, f)$:
  - Compute a decomposed encoding circuit $\hat{f}_1, \ldots, \hat{f}_\ell \leftarrow \mathsf{RE.Enc}(f, 1^\lambda)$.
  - Sample a symmetric encryption key $\mathsf{SK} \leftarrow \{0,1\}^\lambda$.
  - Sample an encryption $\mathsf{CT} \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, 0^\ell)$.
  - Construct circuits $\hat{F}_1, \ldots, \hat{F}_\ell : \{0,1\}^n \times \{0,1\}^\lambda \times \{0,1\}^\lambda \times \{0,1\} \to \{0,1\}$ as follows:

$$\hat{F}_i(m, r, \mathsf{SK}, b) = \begin{cases} \hat{f}_i(m; r), & \text{if } b = 0; \\ \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i, i) & \text{if } b = 1. \end{cases}$$

  - Sample keys $(\mathsf{PK}, \mathsf{FSK}_{\hat{F}_1}, \ldots, \mathsf{FSK}_{\hat{F}_\ell}) \leftarrow \mathsf{MFE.Setup}(1^\lambda, \hat{F}_1, \ldots, \hat{F}_\ell)$.
  - Output:
    - PK.
    - $\mathsf{FSK}_f = \mathsf{FSK}_{\hat{F}_1}, \ldots, \mathsf{FSK}_{\hat{F}_\ell}$.
- $\mathsf{FE.Enc}(\mathsf{PK}, m)$:
  - Sample encoding randomness $r \leftarrow \{0,1\}^\lambda$.
  - Output $\mathsf{FCT} \leftarrow \mathsf{MFE.Enc}(\mathsf{PK}, (m, r, \perp, 0))$.
- $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT})$:
  - Parse $\mathsf{FSK}_f = \mathsf{FSK}_{\hat{F}_1}, \ldots, \mathsf{FSK}_{\hat{F}_\ell}$.
  - Compute an encoding $\hat{y} \leftarrow \mathsf{MFE.Dec}(\mathsf{FSK}_{\hat{F}_1}, \mathsf{FCT}), \ldots, \mathsf{FE.Dec}(\mathsf{FSK}_{\hat{F}_\ell}, \mathsf{FCT})$.
  - Output the decoded value $\mathsf{RE.Dec}(\hat{y})$.

Figure 5: A single-key functional encryption with succinct encryption

Theorem 4.3. FE *is a single-key functional encryption for all circuits. If* MFE *is weakly (depth/size) succinct, or* MFE *is very weakly depth-succinct and both* RE *and* Sym *are shallow, then* FE *is succinct. If* MFE, RE, *and* Sym *are all δ-secure so is* FE.

The above theorem, in particular, implies that any single key FE that is weakly depth-succinct implies a succinct FE, and assuming also pseudorandom functions in NC^1, so does any very weakly depth-succinct FE.

Corollary 4.4 (of Claim 4.2 and Facts 2.7,2.9)**.** *Any (δ-secure) single key functional encryption that is weakly depth-succinct can be transformed into a (δ-secure) succinct scheme. Assuming pseudorandom functions in NC^1, very weak depth-succinctness is sufficient.*

Proof. We prove that the constructed scheme satisfies the properties of single-key FE with succinct encryption.

Correctness: Fix a security parameter $\lambda \in \mathbb{N}$, message $m \in \{0,1\}^n$, and $f \in \mathcal{F}$ with domain $\{0,1\}^n$. By the functionality of MFE and correctness of RE,

$$\begin{aligned}\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{FCT}) = \ & \mathsf{RE.Dec}\left(\left\{\mathsf{MFE.Dec}(\mathsf{FSK}_{\hat{F}_i}, \mathsf{FCT})\right\}_{i\in[\ell]}\right) = \\ & \mathsf{RE.Dec}\left(\left\{\hat{F}_i(m, r, \bot, 0)\right\}_{i\in\ell}\right) = \\ & \mathsf{RE.Dec}\left(\left\{\hat{f}_i(m; r)\right\}_{i\in[\ell]}\right) = f(m) \ ,\end{aligned}$$

where $(\mathsf{PK}, \mathsf{FSK}_f) \leftarrow \mathsf{FE.Setup}(1^\lambda, f)$, $\mathsf{FSK}_f = \left\{\mathsf{FSK}_{\hat{F}_i}\right\}_{i\in[\ell]}$, $\hat{F}_i$ is the function underlying $\mathsf{FSK}_{\hat{F}_i}$, $\mathsf{FCT} = \mathsf{FE.Enc}(\mathsf{EK}, m)$, and $\left\{\hat{f}_i\right\} \leftarrow \mathsf{RE.Enc}(f)$.

Selective security: Fix any polynomial-size $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we prove that there exists negligible $\mu_{\mathsf{FE}}(\lambda)$ such that for any security parameter λ, function $f : \{0,1\}^n \to \{0,1\}^m$, and two messages $m_0, m_1 \in \{0,1\}^n$ such that $f(m_0) = f(m_1)$,

$$\mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_0) \approx_{\mathcal{A},\mu} \mathsf{FSK}_f, \mathsf{FE.Enc}(\mathsf{PK}, m_1) \ ,$$

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

For this purpose, we consider the following sequence hybrid experiments.

$\mathcal{H}_b^0$ for $b \in \{0,1\}$: This hybrid corresponds to the (real) distribution:

$$\begin{aligned}&(\mathsf{PK}, \mathsf{FSK}_f) = (\mathsf{PK}, \mathsf{FSK}_{\hat{F}_1}, \ldots, \mathsf{FSK}_{\hat{F}_\ell}) \leftarrow \mathsf{MFE.Setup}(1^\lambda, \hat{F}_1, \ldots, \hat{F}_\ell) \ , \\ &\mathsf{FCT} \leftarrow \mathsf{MFE.Enc}(\mathsf{PK}, (m_b, r, \bot, 0)) \ ,\end{aligned}$$

where $\hat{F}_i = \hat{F}_i[\mathsf{CT}_i]$ is a circuit that according to a choice input bit β, either computes the ith encoding $\hat{f}_i(m, r)$ or decrypts CT_i. In this hybrid, $\beta = 0$ and the circuit performs the first operation. The ciphertext $\mathsf{CT} = \mathsf{CT}_1 \ldots, \mathsf{CT}_\ell \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, 0^\ell)$ encrypts zeros.

$\mathcal{H}_b^1$ for $b \in \{0,1\}$: This hybrid is identical to $\mathcal{H}_b^0$ except that $\mathsf{CT} \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, \hat{f}_1(m_b; r), \ldots, \hat{f}_\ell(m_b; r))$ is an encryption of the encoding $\hat{f}(m; r) = \hat{f}_1(m; r), \ldots, f_\ell(m; r)$ instead of 0^ℓ.

$\mathcal{H}_b^2$ for $b \in \{0,1\}$: This hybrid is identical to $\mathcal{H}_b^2$ except that $\mathsf{FCT} \leftarrow \mathsf{MFE.Enc}(\mathsf{PK}, (\bot, \bot, \mathsf{SK}, 1))$ encrypts $(\bot, \bot, \mathsf{SK}, 1)$ instead of $(m_b, r, \bot, 0)$.

$\mathcal{H}^3$: This hybrid is identical to $\mathcal{H}_b^3$ except that $\mathsf{CT} \leftarrow \mathsf{Sym.Enc}(\mathsf{SK}, \hat{y}_1, \ldots, \hat{y}_\ell)$ is an encryption of a simulated encoding $y_1, \ldots, \hat{y}_\ell \leftarrow \mathsf{RE.Sim}(f, f(m_0))$ instead of $\hat{f}(m_b; r)$.

Claim 4.5. *There exist negligible functions $\mu_{\mathsf{Sym}}(\lambda), \mu_{\mathsf{MFE}}(\lambda), \mu_{\mathsf{RE}}(\lambda)$ such that:*

$$\mathcal{H}_b^0 \approx_{\mathcal{A},\mu_{\mathsf{Sym}}} \mathcal{H}_b^1 \approx_{\mathcal{A},\mu_{\mathsf{MFE}}} \mathcal{H}_b^2 \approx_{\mathcal{A},\mu_{\mathsf{RE}}} \mathcal{H}^3 \ .$$

Furthermore, if $\mathsf{Sym}, \mathsf{MFE}, \mathsf{RE}$ *are δ-secure, the functions μ above can be replaced with $\delta^{\Omega(1)}$.*

Before proving the claim, note that it concludes the security proof. Indeed, setting $\mu_{\mathsf{FE}} = 2(\mu_{\mathsf{Sym}} + \mu_{\mathsf{MFE}} + \mu_{\mathsf{RE}})$, it implies indistinguishability of the two real experiments: $\mathcal{H}_0^0 \approx_{\mathcal{A},\mu_{\mathsf{FE}}} \mathcal{H}_0^1$.

Proof of Claim 4.5. We prove indistinguishability between each two subsequent hybrids.

$\mathcal{H}_b^0 \approx \mathcal{H}_b^1$: Here the difference is in the symmetric encryption $\mathsf{CT} = \mathsf{CT}_1, \ldots, \mathsf{CT}_\ell$ underlying the functions $\hat{F}_1[\mathsf{CT}_1], \ldots, \hat{F}_\ell[\mathsf{CT}_\ell]$. In $\mathcal{H}_b^0$, we encrypt 0^ℓ and in $\mathcal{H}_b^1$, we encrypt $\hat{f}(m_b, r)$.

Thus, by the indistinguishability of Sym, there exists μ_{Sym} such that:

$$\mathcal{H}_b^0 \approx_{\mathcal{A},\mu_{\mathsf{Sym}}} \mathcal{H}_b^1 \ .$$

$\underline{\mathcal{H}_b^1 \approx \mathcal{H}_b^2}$: Here the difference is in the functional ciphertext FCT. In $\mathcal{H}_b^1$, we encrypt $(m_b, r, \perp, 0)$ and in $\mathcal{H}_b^2$, we encrypt $(\perp, \perp, \mathsf{SK}, r)$. For every function $\hat{F}_i$,

$$\hat{F}_i(m_b, r, \perp, 0) = \hat{f}_i(m_b; r) = \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i, i) = \hat{F}_i(\perp, \perp, \mathsf{SK}, 1) \ .$$

Thus, by the selective security of MFE, there exists μ_{MFE} such that:

$$\mathcal{H}_b^1 \approx_{\mathcal{A},\mu_{\mathsf{MFE}}} \mathcal{H}_b^2 \ .$$

$\underline{\mathcal{H}_b^2 \approx \mathcal{H}_b^3}$: Here the difference is again in the symmetric encryption $\mathsf{CT} = \mathsf{CT}_1, \ldots, \mathsf{CT}_\ell$ underlying the functions $\hat{F}_1[\mathsf{CT}_1], \ldots, \hat{F}_\ell[\mathsf{CT}_\ell]$. In $\mathcal{H}_b^2$, we encrypt the encoding $\hat{f}(m_b, r)$ and in $\mathcal{H}^3$ we encrypt the simulated encoding $\hat{y} \leftarrow \mathsf{RE.Sim}(f, y)$, where $y = f(m_0) = f(m_b)$.

Thus, by the privacy of RE, there exists μ_{RE} such that:

$$\mathcal{H}_b^2 \approx_{\mathcal{A},\mu_{\mathsf{RE}}} \mathcal{H}^3 \ .$$

In all of the above, it follows readily that if $\mathsf{Sym}, \mathsf{RE}, \mathsf{MFE}$ are δ-secure for some concrete negligible δ then FE is δ-secure.

This concludes the proof of the claim and of the security of the constructed FE. □

Succinct encryption: We first show that if MFE is weakly size-succinct, then FE is succinct. This also implies that FE is succinct if MFE is weakly depth-succinct, as weak depth-succinctness implies weak size-succinctness. We then show that FE is succinct also if MFE is very weakly depth-succinct and RE and Sym are shallow.

Assume that MFE is weakly size-succinct. That is, there exist a polynomial Φ_{MFE} and a constant $0 < \varepsilon \leq 1$ such that the size of the encryption circuit corresponding to ℓ functions is bounded by

$$\ell^{1-\varepsilon} \cdot \Phi_{\mathsf{MFE}}(\hat{s}, \lambda) \ ,$$

where $\hat{s}$ is the circuit-size of each of the ℓ functions chosen during the setup phase. We show that FE is also succinct with parameters $(\varepsilon, \Phi_{\mathsf{FE}})$ for a fixed polynomial Φ_{FE}.

Fix a size-s function f. We analyze the size and depth of the corresponding functions $\hat{F}_1, \ldots, \hat{F}_\ell$. Recall that each function $\hat{F}_i = \hat{F}_i[\mathsf{CT}_i]$ either decrypts a bit, or computes one encoding bit. Specifically, it has the form

$$\hat{F}_i(m, r, \mathsf{SK}, b) = b \otimes \mathsf{Sym.Dec}(\mathsf{SK}, \mathsf{CT}_i, i) \oplus (1 - b) \otimes \hat{f}_i(m; r) \ ,$$

where $\otimes, \oplus$ denote multiplication and addition modolu 2.

Thus,

$$\hat{s} := |\hat{F}_i| \leq O(|\mathsf{Sym.Dec}(\cdot, \mathsf{CT}_i, i)| + |\hat{f}_i(\cdot;\cdot)|) \leq \Phi_{\mathsf{Sym,RE}}(n, \lambda) \ ,$$

where $\Phi_{\mathsf{Sym,RE}}$ is a fixed polynomial that depends only on the schemes Sym and RE. Furthermore, by the decomposability of RE, it is the case that

$$\ell \leq s \cdot \Phi_{\mathsf{RE}}(n, \lambda) \ ,$$

where Φ_{RE} is a fixed polynomial that depends only on RE.

Applying the weak size-succinctness of MFE, it follows that in FE, the size of an encryption circuit corresponding to a size-s function f is bounded by

$$(s \cdot \Phi_{\mathsf{RE}}(n,\lambda))^{1-\varepsilon}\, \Phi_{\mathsf{MFE}}(\Phi_{\mathsf{Sym,RE}}(n,\lambda)\,\lambda) \ .$$

This implies succinctness of FE with parameters (ε, Φ) when setting

$$\Phi(n,\lambda) = \Phi_{\mathsf{RE}}(n,\lambda)\Phi_{\mathsf{MFE}}(\Phi_{\mathsf{Sym,RE}}(n,\lambda),\lambda) \ .$$

Assume now that MFE is very weakly depth-succinct. That is, there exists a polynomial Φ_{MFE} and constant $0 < \varepsilon \leq 1$ such that the size of the encryption circuit corresponding to ℓ functions is bounded by

$$(\ell\hat{s})^{1-\varepsilon} \cdot \Phi_{\mathsf{MFE}}(n, 2^{\hat{d}}, \lambda \ ,$$

where $\hat{s}$ is the circuit-size and $\hat{d}$ is the circuit-depth of each of the ℓ functions chosen during the setup phase. Assume also that both Sym and RE are shallow.

Recalling again the definition of the functions $\hat{F}_1, \ldots, \hat{F}_\ell$ note that the depth of each $\hat{F}_i$ is at most

$$c \cdot \max\{d_{\mathsf{Sym}}, d_{\mathsf{RE}}\}) \leq 2c \cdot \log\lambda \ ,$$

where d_{Sym} and d_{RE} are the depth of the decryption circuit in Sym and of each $\hat{f}_i$ in RE, c is an absolute constant, and we rely on the fact that both schemes are shallow.

Applying the very weak depth-succinctness of MFE, it follows that in FE, the size of an encryption circuit corresponding to a size-s function f is bounded by

$$(s \cdot \Phi_{\mathsf{RE}}(n,\lambda) \cdot \Phi_{\mathsf{Sym,RE}}(n,\lambda))^{1-\varepsilon}\, \Phi_{\mathsf{MFE}}(n, 2^{2c\log\lambda}, \lambda) \ .$$

This implies succinctness of FE with parameters (ε, Φ) when setting

$$\Phi(n,\lambda) = \Phi_{\mathsf{RE}} \cdot \Phi_{\mathsf{Sym,RE}}(n,\lambda) \cdot \Phi_{\mathsf{MFE}}(n, \lambda^{2c}, \lambda) \ .$$

□

Acknowledgements

We thank the reviewers of JACM for their insightful comments and valuable suggestions. We thank Daniel Wichs for pointing out an error in a previous version of the paper.

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 Canetti and Garay [CG13], pages 500–518.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 166–175, 2004.

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

[AIKW13] Benny Applebaum, Yuval Ishai, Eyal Kushilevitz, and Brent Waters. Encoding functions with constant online rate or how to compress garbled circuits keys. In Canetti and Garay [CG13], pages 166–184.

[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.

[AJS17] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation for turing machines: Constant overhead and amortization. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 252–279, 2017.

[AS17] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 152–181, 2017.

[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.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography.*, pages 79–158. 2017.

[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.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469. Springer, 1997.

[CG13] 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*. Springer, 2013.

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

[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.

[CIJ+13] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In Canetti and Garay [CG13], pages 519–535.

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

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, 2010.

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

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 578–602, 2014.

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

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

[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+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 555–564, 2013.

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

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

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

[GPSZ17] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the sub-exponential barrier in obfustopia. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, pages 156–181, 2017.

[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. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 419–442, 2016.

[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.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HMS07] Dennis Hofheinz, John Malone-Lee, and Martijn Stam. Obfuscation for cryptographic purposes. In Salil P. Vadhan, editor, *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings*, volume 4392 of *Lecture Notes in Computer Science*, pages 214–232. Springer, 2007.

[HRSV11] Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. *J. Cryptology*, 24(4):694–719, 2011.

[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.

[KNT17] Fuyuki Kitagawa, Ryo Nishimaki, and Keisuke Tanaka. Indistinguishability obfuscation for all circuits from secret-key functional encryption. *IACR Cryptology ePrint Archive*, 2017:361, 2017.

[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.

[Lin17] Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 prgs. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 599–629, 2017.

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, pages 443–468, 2016.

[LPS04] Ben Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 20–39. Springer, 2004.

[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.

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local prgs. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, pages 630–660, 2017.

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from ddh-like assumptions on constant-degree graded encodings. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 11–20, 2016.

[LZ17] Qipeng Liu and Mark Zhandry. Decomposable obfuscation: A framework for building applications of obfuscation from polynomial hardness. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, pages 138–169, 2017.

[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.

[Wee05] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 523–532. ACM, 2005.

[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.