Indistinguishability Obfuscation from Functional Encryption

Nir Bitansky* Vinod Vaikuntanathan†

February 26, 2015

Abstract

Indistinguishability obfuscation is a tremendous notion, powerful enough to give rise to almost any known cryptographic object. We construct indistinguishability obfuscation from any public-key functional encryption scheme with succinct ciphertexts and sub-exponential security.

*MIT. E-mail: nirbitan@csail.mit.edu.
†MIT. E-mail: vinodv@mit.edu.

1 Introduction

Program obfuscation, aiming to turn programs into "unintelligible" ones while preserving functionality, has been a holy grail in cryptography for over a decade. Rather unfortunately, the most natural and intuitively appealing notion of obfuscation, namely *virtual-black-box* (VBB) obfuscation [BGI+12], was shown to have strong limitations [BGI+12, GK05, BCC+14]. Furthermore, except for very restricted function classes, no candidate construction with any form of meaningful security was known.

This dramatically changed with recent breakthrough results. First, Garg, Gentry, Halevi, Raykova, Sahai and Waters [GGH+13] demonstrated a candidate obfuscation algorithm for all circuits, and conjectured that it satisfies an apparently weak notion of *indistinguishability obfuscation* (IO) [BGI+12, GR07], requiring only that the obfuscations of any two circuits of the same size and functionality are 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, GLSW14], all known IO constructions [GGH+13, PST14, BR14, BGK+14, GLSW14, AB15, Zim15] are still based on the hardness of little-studied problems on multi-linear maps [GGH12]. Thus, an outstanding question in the foundations of cryptography is:

Can we base indistinguishability obfuscation on strong cryptographic foundations?

1.1 This Work

In this work, we make progress in the above direction, showing how to construct indistinguishability obfuscation from an apparently weaker primitive: *public-key functional encryption*. In a functional encryption scheme [BCOP04, SW05, BSW12, O'N10], the owner of a master secret key MSK can produce functional keys FSK_f for functions f. Given such a functional key FSK_f and an encryption of an input x, one can compute $f(x)$, but nothing more about x itself.

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

Our main result is the following:

Theorem 1.1 (informal)**.** *Assuming the existence of a sub-exponentially secure public-key functional encryption scheme for all circuits, where the ciphertext size is polynomial in the input-size and sub-linear in the circuit-size, there exists indistinguishability obfuscation for all circuits.*

Furthermore, in the above theorem, it suffices to start from a scheme that supports only a single-key and satisfies a mild selective-security indistinguishability-based guarantee. The underlying assumption can in fact be further relaxed to require only functional encryption for NC^1, assuming (sub-exponential) puncturable pseudo-random functions in NC^1, or fully homomorphic encryption with decryption in NC^1.

Interpretation. As far as we know, functional encryption schemes satisfying the ciphertext compactness required in Theorem 1.1 are only known based on (sub-exponential) IO [GGH+13, Wat14], and thus our result establishes the equivalence of these two primitives (up to some sub-exponential loss). The question

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

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. Indeed, in the past few years, we have seen a remarkable progress in the area of functional encryption [SS10, GVW12, GVW13, GGHZ14]. The state of the art scheme based on a standard assumption is that of Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [GKP+12] relying on the sub-exponential learning with errors assumption. The construction achieves ciphertext size that only grows polynomially with the circuit output size and depth; thus, for circuits with say a single output bit, ciphertexts may indeed be sub-linear in circuit size, but this will not be the case for circuits with long outputs. Interestingly, the latter construction achieves a strong simulation-based security guarantee, under which sub-linear growth in the output size is actually impossible [AGVW13, GKP+12]. Reducing the dependence on the output (under an indistinguishability-based notion) has been a tantalizing problem. Now this question becomes of central importance in the quest to achieve indistinguishability obfuscation.

In a recent result, Gorbunov, Vaikuntanathan and Wee [GVW15] showed how to construct predicate encryption schemes for all circuits (with a-priori bounded depth) from the sub-exponential learning with errors (LWE) assumption. In their scheme, the ciphertext size is polynomial in the input length and the depth of the circuit, and otherwise independent of the circuit size and output size. A predicate encryption scheme can be interpreted as a functional encryption scheme with a "weak attribute hiding" property (see [KSW13, AFV11, GVW15] for more details). Strengthening this to "full attribute hiding" will give us a functional encryption scheme that satisfies the requirements of Theorem 1.1, and is yet another frontier in achieving indistinguishability obfuscation from LWE.

1.2 Main Ideas

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

As observed in [GKP+12], and generalized in [BS15], in *private-key* functional encryption schemes, it is always possible to harness the existing message-hiding to also guarantee function-hiding. This can be interpreted as a relaxed form of interactive obfuscation termed in [GKP+12] as *token based obfuscation*. Here the function-hiding functional-key FSK_f is seen as an obfuscation of f. In order to evaluate the obfuscation on an input x, the evaluator first needs to request a corresponding token, which is just an encryption of x. The major drawback of course is that encryption is a private-key operation, meaning that tokens cannot be generated publicly and require interaction with the secret-key owner.

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

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

1. *IO from much less IO.* We first show that the above intuition can be fulfilled, not only with ideal obfuscation, but also in the context of indistinguishability obfuscation. Concretely, starting from functional encryption, we obfuscate, under the IO notion, any function f assuming IO only for a restricted class of smaller circuits that simply generate encrypted inputs.

2. *Recurse: IO from slightly less IO.* In the second step, with the goal of obfuscating the latter input encryption circuit, we show how to recursively reduce IO for circuits that encrypt n input bits to functional encryption and IO for circuits that encrypt only a smaller number of $n-1$ bits. At the base of this recursion, we only need to obfuscate circuits with a single input bit, which can be done trivially by writing only their respective outputs.

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

Step 1: IO from much less IO. A natural first attempt to achieve our goal is to mimic the ideal solution. Namely, starting from a (private-key) function-hiding functional encryption scheme, to obfuscate any f, generate the functional key FSK_f and add an obfuscation $i\mathcal{O}(\mathsf{Enc}(\mathsf{MPK},\cdot))$ of the corresponding encryption circuit. Here encryption is derandomized in the standard way by applying a pseudo-random function (PRF) to the inputs. While this solution would have worked with an ideal notion of obfuscation (e.g. auxiliary-input VBB), it is not clear how to prove its security based solely on IO. In fact, using similar ideas to those in the impossibility result of Barak et al. [BGI+12], one can show that we cannot hope to rely *any* private-key (function-hiding) scheme, since there exists such schemes where access to an encryption circuit may lead to a devastating attack.

Our solution will, in fact, rely on public-key functional encryption. Here function-hiding is not be guaranteed; rather, we shall enforce it explicitly in our construction using similar techniques to those used in the private-key setting [BS15] going back to the classic two-key paradigm [FS89, NY90].

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

Proving that the above construction is secure can be decoupled into two main ideas that go back to previous works. The first comes from the work of Brakerski and Segev [BS15]. There the adversary, whose goal is to distinguish between a functional key corresponding to f_0 to one corresponding to a functionally-equivalent f_1, does not ever obtain a circuit that computes the above encryptions. Rather it only views the outputs of this circuit. Let us, in fact, think about a simple case where the distinguisher only obtains a single encryption $\mathsf{Enc}^*(x) := \mathsf{Enc}(\mathsf{MPK},(x,\mathsf{SK}_0))$ of some pre-selected input x. In this setting, we can employ a straight forward hybrid argument to show that the functional keys $(\mathsf{FSK}_{f_0^*}, \mathsf{FSK}_{f_1^*})$ corresponding to f_0 and f_1 are indistinguishable. Indeed, relying on the symmetric-key guarantee we can change CT_1 to encrypt f_1, and then relying on the FE guarantee we change $\mathsf{Enc}^*(x)$ to encrypt SK_1 instead of SK_0, indeed we know that $f_0(x) = f_1(x)$. Then, we can symmetrically switch the other cipher to encrypt f_1 and switch the keys again.

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

Concretely, it would suffice to show that we can change the circuit $\mathsf{Enc}^*(x)$ to freely switch between encrypting SK_0 to encrypting SK_1 *for all inputs simultaneously*. Here comes into play another idea that has been used in several recent works and formalized by Canetti, Lin, Tessaro, and Vaikuntanthan as

probabilistic IO. They show that given two public samplers $C_0(x;r), C_1(x;r)$ such that for any input x $C_0(x)$ and $C_1(x)$ are computationally indistinguishable, the circuits can be derandomized using a *puncturable PRF* and obfuscated so that their IO obfuscations are indistinguishable. In our setting, we simply apply this argument to the circuits $C_b(x) := \mathsf{Enc}^*(x, \mathsf{SK}_b)$, and make sure to derandomize it with a puncturable PRF. One restriction inherited from this argument is that it only works assuming that the underlying IO and puncturable PRF are both sub-exponentially secure. Also, for the argument to hold indistinguishability is required even given the public circuits, which is the reason for our reliance on public-key functional encryption. We do not know whether the above solution (or any solution) can be based on general private-key functional encryption. (It does seem that a certain notion of a puncturable private-key scheme would suffice, but at this point do not know how to achieve this notion, without relying on public-key schemes.)

Step 2: IO from slightly less IO. We have reduced the complexity of the circuit to be obfuscated from that of f to that of Enc^*, but how do we obfuscate Enc^*? Here using a similar approach to that above, we show how to reduce the obfuscation of Enc^* that deals with n bit inputs, to an obfuscation of Enc^*_{n-1} that only deals with $n-1$ input bits. Concretely, we now think of a function f^*_n that expects an input $x \in \{0,1\}^{n-1}$ and outputs two encryptions $\mathsf{Enc}^*(x0), \mathsf{Enc}^*(x1)$. Accordingly, we publish $\mathsf{FSK}_{f^*_n}$ and of Enc^*_{n-1}. This process is than performed recursively, at each level sampling a new instance of the functional encryption scheme as well of the symmetric key encryption, until the last step where Enc^*_1 simply consists of two hardwired encryption.

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

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

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

Organization

In Section 2, we give the definitions of the cryptographic primitives used throughout the work, including functional encryption. In Section 3, we describe the transformation from public-key functional encryption to indistinguishability obfuscation and analyze it.

2 Definitions

The cryptographic definitions in the paper follow the convention of modeling security against non-uniform adversaries. An efficient adversary $\mathcal{A}$ is modeled as a sequence of circuits $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$, such

that each circuit $\mathcal{A}_\lambda$ is of polynomial size $\lambda^{O(1)}$ with $\lambda^{O(1)}$ input and output bits. We often omit the subscript λ when it is clear from the context.

2.1 Public-Key Functional Encryption

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

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

- $\mathsf{FE.Setup}(1^\lambda)$: Takes as input a security parameter λ in unary and outputs a (master) public key and a secret key $(\mathsf{MPK}, \mathsf{MSK})$.
- $\mathsf{FE.Gen}(\mathsf{MSK}, f)$: Takes as input a secret key MSK, a function $f \in \mathcal{F}$ and outputs a functional key FSK_f.
- $\mathsf{FE.Enc}(\mathsf{MPK}, m)$: Takes as input a public key MPK, 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 $\mathsf{FE.Enc}(\mathsf{MPK}, m; r)$.
- $\mathsf{FE.Dec}(\mathsf{FSK}_f, \mathsf{CT})$: Takes as input a functional key FSK_f, a ciphertext CT and outputs $\widehat{m}$.

We next the define the required correctness and security properties.

Definition 2.1 (Selectively-secure public-key FE)**.** *A tuple of PPT algorithms* $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Gen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ *is a selectively-secure public-key functional encryption scheme, for function class* $\mathcal{F}$*, and message space* $\{0,1\}^*$*, if it satisfies:*

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

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

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

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

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

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

   (b) *The challenger executes* $\mathsf{FE.Setup}(1^\lambda)$ *to obtain* $(\mathsf{MPK}, \mathsf{MSK})$*. It then executes* $\mathsf{FE.Enc}(\mathsf{MPK}, m_b)$ *to obtain* CT*. The challenger sends* $(\mathsf{MPK}, \mathsf{CT})$ *to the adversary.*

   (c) *The adversary submits function queries to the challenger. For any submitted function query* $f \in \mathcal{F}$ *defined over* $\{0,1\}^n$*, if* $f(m_0) = f(m_1)$*, the challenger generates and sends* $\mathsf{FSK}_f \leftarrow \mathsf{FE.Gen}(\mathsf{MSK}, f)$*. In any other case, the challenger aborts.*

   (d) *The output of the experiment is the output of* $\mathcal{A}$*.*

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

Relaxation: single key schemes with sub-linear circuit dependence. In this paper, we can further relax the above definition in two ways:

1. *Single-key release:* in the above definition, the adversary may make any polynomial number of functional queries $\{f_i\}$. It will suffice that security holds with respect to adversaries that only perform a single query f.

2. *Sub-linear output dependence:* in the above definition, the running time of the encryption algorithm (and ciphertext size) is polynomial in the plaintext size n and the security parameter λ. We can, in fact, allow encryption time (and ciphertext size) to grow sub-linearly in the circuit size of functions. We shall say that encryption is $(1-\varepsilon)$-compact if encryption time is bounded by $s^{1-\varepsilon} \cdot \mathrm{poly}(n, \lambda)$ where $s = \max_{f \in \mathcal{F}_n} |f|$, $\mathcal{F}_n \subseteq \mathcal{F}$ is the subset of functions defined on $\{0,1\}^n$, and poly is any fixed polynomial.

2.2 Indistinguishability Obfuscation

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

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

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

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

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

2.3 Puncturable Pseudorandom Functions

We consider a simple case of the puncturable pseudo-random functions (PRFs) where any PRF may be punctured at a single point. The definition is formulated as in [SW14], and is satisfied by the GGM [GGM86] PRF [BW13, KPTZ13, BGI14].

Definition 2.3 (Puncturable PRFs)**.** *Let n, k be polynomially bounded length functions. An efficiently computable family of functions*

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

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

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

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

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

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

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

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

2.4 Symmetric Encryption

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

Definition 2.4 (One-Time Symmetric Encryption)**.** *A pair of PPT algorithms* $(\mathsf{Sym.Enc}, \mathsf{Sym.Dec})$ *is a one-time symmetric encryption scheme for message space* $\{0,1\}^*$ *if it satisfies:*

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

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

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

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

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

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

3 The Transformation

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

Ingredients. We rely on the following primitives:

- A $2^{-\tilde{\lambda}^\varepsilon}$-secure public-key functional encryption scheme FE for P/poly, for a single key (possibly with sub-linear ciphertext dependence on circuit size).
- A $2^{-\tilde{\lambda}^\varepsilon}$-secure one-time symmetric encryption scheme Sym,
- A $2^{-\tilde{\lambda}^\varepsilon}$-secure puncturable pseudo-random function family $\mathcal{PRF}$.

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

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

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

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

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

Figure 1: The recursive obfuscation procedure.

Theorem 3.1. *$i\mathcal{O}$ is an indistinguishability obfuscator for P/poly.*

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

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

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

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

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

Figure 2: The recursive evaluation procedure.

Efficiency. For simplicity, let us first assume that the encryption time of the underlying FE scheme is completely independent of the circuit or output size of functions. In Appendix A, we show how to extend the analysis to the case of sub-linear dependence on the circuit size.

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

3.1 Security Analysis

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

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

where C_0 and C_1 are any two circuits defined on $\{0,1\}^{n(\lambda)}$ of the same functionality and size $s(\lambda)$.
For every every $\lambda \in \mathbb{N}$, define $\delta_{n(\lambda)} := \delta_{i\mathcal{O}}(\lambda)$ and for $1 \leq i < n(\lambda)$, define

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

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

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

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

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

Proof of Proposition 3.1. First, to see that $\delta_1 = 0$, note that for any C defined on $\{0,1\}$,

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

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

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

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

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

Recall that

$$r\mathcal{O}.\mathsf{Obf}(i, C_b, 1^{\tilde{\lambda}}) = \widetilde{\mathsf{E}}_{i-1} = r\mathcal{O}.\mathsf{Obf}(i-1, \mathsf{E}^0_{i-1}, 1^{\tilde{\lambda}}), \mathsf{FSK}_i \ ,$$

where E^0_{i-1} is a circuit that has $(\mathsf{MPK}_i, \mathsf{SK}^0_i, \mathsf{K}_i)$ hardwired, and which on input $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$, computes two encryptions

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

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

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

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

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

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

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

We now show equations two and four; concretely, we focus on the second equation, and the forth is proven using a similar argument. Recall again that the difference between $\mathcal{H}_0^{0,1}$ and $\mathcal{H}_1^{0,1}$ is in the obfuscated $\widetilde{\mathsf{E}}_{i-1}$. In the first, the circuit E_{i-1}^0, which always puts SK_i^0 in the plaintext, is obfuscated, and in the second E_{i-1}^1, which always puts SK_i^1 in the plaintext, is obfuscated. The key to the indistinguishability behind the hybrids is that the output of the two circuits on any point $\mathbf{x}_{i-1} \in \{0,1\}^{i-1}$ is indistinguishable even given the two circuits themselves as long as the randomness used to generate the output is not revealed. Indeed, because the circuits encrypted in $\mathsf{CT}_i^0, \mathsf{CT}_0^1$ compute the same function, FSK_i does not allow distinguishing between the two cases and we can invoke the FE guarantee. Canetti, Lin, Tessaro, and Vaikuntanathan [CLTV15] show that sub-exponential IO in conjunction with sub-exponential puncturable PRFs are sufficient in this setting, which they formalize by *probabilistic IO* notion. For the sake of completeness, we next give the full argument.

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

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

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

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

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

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

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

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

  hardwired as well as a punctured key $\mathsf{K}_i\{\mathbf{x}\}$. The circuit is padded to size $\ell(\tilde{\lambda})$.

  Since $\mathsf{E}_{\mathbf{x}-1}$ and $\mathsf{E}'_{\mathbf{x}-1}$ compute the same function:

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

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

  $$\left\{\mathsf{FEnc}(\mathsf{MPK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); \mathsf{PRF}_{\mathsf{K}_i}(\mathbf{x}, x_i)) \;\middle|\; x_i \in \{0,1\}\right\}$$

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

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

  By pseudo-randomness at punctured points

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

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

$$\left\{\mathsf{FEnc}(\mathsf{MPK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^0, 0); r) \;\middle|\; x_i \in \{0,1\}\right\}$$

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

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

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

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

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

$$\left\{\mathsf{FEnc}(\mathsf{MPK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^1, 1); r) \;\middle|\; x_i \in \{0,1\}\right\}$$

with

$$\left\{\mathsf{FEnc}(\mathsf{MPK}_i, ((\mathbf{x}, x_i), \mathsf{SK}_i^1, 1); \mathsf{PRF}_{\mathsf{K}_i}(\mathbf{x}, x_i)) \;\middle|\; x_i \in \{0,1\}\right\} \quad .$$

By pseudo-randomness at punctured points

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

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

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

Overall,

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

as required, which completes the proof of the proposition.

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

Remark 3.2 (Functional encryption for NC^1 rather than for all circuits). We observe that in our construction one can start from a functional encryption scheme only for NC^1 assuming symmetric-key encryption with decryption in NC^1. The resulting IO will accordingly hold only for NC^1 and can then be bootstrapped using known techniques based on either fully-homomorphic encryption with decryption in NC^1 [GGH+13] or based on sub-exponential punctutrable PRFs in NC^1 [App14, CLTV15, BGL+15].

References

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

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2011.

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

[App14] Benny Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, pages 162–172, 2014.

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

[BCOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 506–522, 2004.

[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 Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 221–238. Springer, 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.

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

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

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

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

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

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

[GGH12] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices and applications. *IACR Cryptology ePrint Archive*, 2012:610, 2012.

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

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

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

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

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

[GLSW14] Craig Gentry, Allison B. Lewko, Amit Sahai, and Brent Waters. Indistinguishability obfuscation from the multilinear subgroup elimination assumption. *IACR Cryptology ePrint Archive*, 2014:309, 2014.

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

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

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

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

[KMN+14] Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 374–383. IEEE Computer Society, 2014.

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

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

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

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

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

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

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

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

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

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

A Efficiency Analysis given Sub-linear Dependence on Circuit Size

In Section 3, we have analyzed the efficiency of our obfuscator, assuming that the running time of the functional encryption algorithm is bounded by some fixed polynomial $\mathrm{poly}(n, \lambda)$ in the total input size n and the security parameter λ, independently of either the circuit or output size of functions.

We now show that efficiency is still guaranteed if we allow encryption time (and ciphertext size) to grow sub-linearly with the circuit size of functions. Namely, encryption time is bounded by

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

where $s = \max_{f \in \mathcal{F}_n} |f|$, and $\varepsilon < 1$ is some constant, and poly is any fixed polynomial. Concretely, we will first show that this holds for some ε related to the underlying cryptographic primitives. We will then observe that this holds for any ε assuming efficient enough pseudo-random generators.

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

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

where poly is some fixed polynomial and c is some constant that depends on the efficiency of symmetric decryption. We can now bound the size of each circuit E_i^0 as follows

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

where the first inequality follows by the bound on encryption time, and the second follows from the previous bound on f_{i+1}. Also,

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

where C is the obfuscated circuit. It follows that,

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

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

$$\prod_{j=0}^{k} p^{(c(1-\varepsilon))^j} = 2^{\log p \sum_{j=0}^{k} (c(1-\varepsilon))^j} \leq p^{\frac{1}{1-c(1-\varepsilon)}} \ .$$

We conclude that

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

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

Remark A.1. A slight inaccuracy is that in Section 3, we have assumed that encryption always uses $\tilde{\lambda}$ bits of randomness (recall that λ is polynomimally related to (n,λ)). Thus when E^0_i computes the pseudo-random function it only incurs fixed blowup $\mathrm{poly}(n,\lambda)$. Now, when encryption time depends on $s^{1-\varepsilon}$, more randomness might be required. We can always resolve this by applying a pseudo-random generator, but we should account for it running time. This can also be reflected in the constant c.

Working with any $\varepsilon < 1$. Looking more closely in the complexity of f_i we observe that we can assume that $c = 1 - o(1)$, assuming pseudo-random generators that run in qausi-linear time in their outputs. Indeed, the circuit f_i first decrypts one of the (symmetric) encryptions of E^0_i and then applies a universal circuit to the decrypted E^0_i and the input x. These operations are quasi-linear in $|\mathsf{E}^0_i|$ assuming that encryption and encryption is done by padding with the result of a PRG that can be computed in quasi-linear time in its output. (The efficiency of the PRG is needed again when taking into account Remark A.1.)