



From Cryptomania to Obfustopia Through Secret-Key Functional Encryption

Nir Bitansky*

Tel Aviv University, Tel Aviv-Yafo, Israel
nirbitan@tau.ac.il

Ryo Nishimaki[†]

NTT Secure Platform Laboratories, Tokyo, Japan
ryo.nishimaki.zk@hco.ntt.co.jp

Alain Passelègue[‡]

ENS de Lyon, Lyon, France
alapasse@gmail.com

Daniel Wichs[§]

Northeastern University, Boston, USA
NTT Research, Palo Alto, USA
wichs@ccs.neu.edu

Communicated by Manoj Prabhakaran

Received 16 November 2017 / Revised 13 September 2019

Online publication 30 September 2019

Abstract. Functional encryption lies at the frontiers of the current research in cryptography; some variants have been shown sufficiently powerful to yield indistinguishability obfuscation (IO), while other variants have been constructed from standard assumptions such as LWE. Indeed, most variants have been classified as belonging to either the former

*Member of the Check Point Institute of Information Security. Supported by the Alon Young Faculty Fellowship, by Len Blavatnik and the Blavatnik Family foundation, and an ISF Grant 18/484. This work was done in part while the author was at MIT and was supported by the DARPA and ARO under Contract No. W911NF-15-C-0236. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA and ARO.

[†]This work was done in part while the author was visiting Northeastern University.

[‡]This work was done in part while the author was visiting Northeastern University. Research supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF Grants 1619348, 1228984, 1136174, and 1065276, BSF Grant 2012378, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

[§]Supported in part by NSF Grants CNS-1347350, CNS-1314722, CNS-1413964.

or the latter category. However, one mystery that has remained is the case of *secret-key functional encryption* with an unbounded number of keys and ciphertexts. On the one hand, this primitive is not known to imply anything outside of minicrypt, the land of secret-key cryptography, but, on the other hand, we do not know how to construct it without the heavy hammers in obfustopia. In this work, we show that (subexponentially secure) secret-key functional encryption is powerful enough to construct indistinguishability obfuscation if we additionally assume the existence of (subexponentially secure) plain public-key encryption. In other words, secret-key functional encryption provides a bridge from cryptomania to obfustopia. On the technical side, our result relies on two main components. As our first contribution, we show how to use secret-key functional encryption to get “exponentially efficient indistinguishability obfuscation” (XIO), a notion recently introduced by Lin et al. (PKC’16) as a relaxation of IO. Lin et al. show how to use XIO and the LWE assumption to build IO. As our second contribution, we improve on this result by replacing its reliance on the LWE assumption with any plain public-key encryption scheme. Lastly, we ask whether secret-key functional encryption can be used to construct public-key encryption itself and therefore take us all the way from minicrypt to obfustopia. A result of Asharov and Segev (FOCS’15) shows that this is not the case under black-box constructions, even for exponentially secure functional encryption. We show, through a non-black-box construction, that subexponentially secure-key functional encryption indeed leads to public-key encryption. The resulting public-key encryption scheme, however, is at most quasi-polynomially secure, which is insufficient to take us to obfustopia.

Keywords. Indistinguishability obfuscation, Functional encryption, Public-key encryption.

1. Introduction

The concept of *functional encryption* [21,55] extends that of traditional encryption by allowing the distribution of *functional decryption keys* that reveal specified functions of encrypted messages, but nothing beyond. This concept is one of the main frontiers in cryptography today. It offers tremendous flexibility in controlling and computing on encrypted data, is strongly connected to the holy grail of program obfuscation [3,22,52], and for many problems, may give superior solutions to obfuscation-based ones [34,35]. Accordingly, recent years have seen outstanding progress in the study of functional encryption, both in constructing functional encryption schemes and in exploring different notions, their power, and the relationship among them (see, for instance, [1,2,4,5,9,12,16,18–20,26–29,32,33,37,38,47,48,58,60] and many more).

One striking question that has yet to be solved is *the gap between public-key and secret-key functional encryption schemes. In particular, does any secret-key scheme imply a public-key one?*

The answer to this question is nuanced and seems to depend on certain features of functional encryption schemes, such as the number of functional decryption keys and number of ciphertexts that can be released. For functional encryption schemes that only allow the release of an *a priori bounded* number of functional keys (often referred to as *bounded collusion*), we know that the above gap is essentially the same as the gap between plain (rather than functional) secret-key encryption and public-key encryption, and should thus be as hard to bridge. Specifically, in the secret-key setting, such schemes supporting an unbounded number of ciphertexts can be constructed assuming low-depth pseudorandom

generators (or just one-way functions in the single-key case) [37,58]. These secret-key constructions are then converted to public-key ones, relying on (plain) public-key encryption (and this is done quite directly by replacing invocations of a secret-key encryption scheme with invocations of a public-key one). The same state of affairs holds when reversing the roles and considering a bounded number of ciphertexts and an unbounded number of keys [37,58]. In other words, in the terminology of Impagliazzo’s complexity worlds [40], if the number of keys or ciphertexts is a priori bounded, then symmetric-key functional encryption lies in *minicrypt*, the world of one-way functions, and public-key functional encryption lies in *cryptomania*, the world of public-key encryption.

For functional encryption schemes supporting an unbounded (polynomial) number of keys and unbounded number of ciphertexts, which will be the default notion throughout the rest of the paper, the question is far less understood (we will often refer to such schemes as *multi-key* and *multi-ciphertext*, respectively). In the public-key setting, such functional encryption schemes with subexponential security are known to imply indistinguishability obfuscation [3,4,22]. In contrast, Bitansky and Vaikuntanathan [22] show that their construction of indistinguishability obfuscation using functional encryption may be insecure when instantiated with a secret-key functional encryption scheme. In fact, secret-key functional encryption schemes (even exponentially secure ones) are not known to imply any cryptographic primitive beyond those that follow from one-way functions. As far as we know, the two notions of functional encryption may correspond to opposite extremes of the complexity spectrum: on one side, public-key schemes correspond to *obfustopia*, the world where indistinguishability obfuscation exists, and on the other side secret-key schemes may lie in *minicrypt* where there is even no (plain) public-key encryption.

One piece of evidence that may support such a view of the world is given by Asharov and Segev [6] who show that there do not exist *fully black-box* constructions of *plain* public-key encryption from secret-key functional encryption, even if the latter is exponentially secure. Still, while we may hope that secret-key functional encryption schemes could be constructed from significantly weaker assumptions than needed for public-key schemes, so far no such construction has been exhibited—all known constructions live in *obfustopia*.

1.1. Our Contributions

In this work, we shed new light on the question of secret-key vs public-key functional encryption (in the multi-key, multi-ciphertext setting). Our main result bridges the two notions based on (plain) public-key encryption.

Theorem 1.1. (Informal) *Assuming secret-key functional encryption and plain public-key encryption that are both subexponentially secure, there exists indistinguishability obfuscation, and in particular, also public-key functional encryption.*

In the terminology of Impagliazzo’s complexity worlds: *secret-key functional encryption would turn cryptomania, the land of public-key encryption, into obfustopia*. This puts in new perspective the question of constructing such secret-key schemes from standard assumptions—any such construction would lead to indistinguishability obfuscation from standard assumptions.

The above result still does not settle the question of whether secret-key functional encryption on its own implies (plain) public-key encryption. Here, we show that assuming subexponentially secure secret-key functional encryption and (almost) exponentially secure one-way functions, there exists (polynomially secure) public-key encryption.

Theorem 1.2. (Informal) *Assuming subexponentially secure secret-key functional encryption and $2^{n/\log \log n}$ -secure one-way functions, there exists (polynomially secure) public-key encryption.*

The resulting public-key encryption is not strong enough to take us to obfustopia. Concretely, the constructed scheme is not subexponentially secure as required by our first theorem—it can be *quasi-polynomially* broken. Nevertheless, the result does show that the black-box barrier shown by Asharov and Segev [6], which applies even if the underlying secret-key functional encryption scheme and one-way functions are *exponentially secure*, can be circumvented. Indeed, our construction uses the functional encryption scheme in a non-black-box way (see further details in A Technical Overview section below).

1.2. A Technical Overview

We now provide an overview of the main steps and ideas leading to our results.

Key observation: from SKFE to (strong) exponentially efficient IO Our first observation is that secret-key functional encryption (or SKFE in short) implies a weak form of indistinguishability obfuscators termed by Lin et al. [51] exponentially efficient indistinguishability obfuscation (XIO). Like IO, this notion preserves the functionality of obfuscated circuits and guarantees that obfuscations of circuits of the same size and functionality are indistinguishable. However, in terms of efficiency the XIO notion only requires that an obfuscation \tilde{C} of a circuit $C : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is just mildly smaller than its truth table, namely $|\tilde{C}| \leq 2^{\gamma n} \cdot \text{poly}(|C|)$, for some compression factor $\gamma < 1$, and a fixed polynomial poly , rather than the usual requirement that *the time to obfuscate*, and in particular the size of \tilde{C} , are polynomial in $|C|$. We show that SKFE implies a slightly stronger notion than XIO where the time to obfuscate C is bounded by $2^{\gamma n} \cdot \text{poly}(|C|)$. We call this notion strong exponentially efficient indistinguishability obfuscation (SXIO). (We note that, for either XIO or SXIO, we shall typically be interested in circuits over some polynomial-size domain, which could be much larger than the circuit itself, e.g., $\{0, 1\}^n$ where $n = 100 \log |C|$.)

Proposition 1.1 (Informal)

1. *For any constant $\gamma < 1$, there exists a transformation from SKFE to SXIO with compression factor γ (and polynomial security loss).*
2. *For some subconstant $\gamma = o(1)$, there exists a transformation from subexponentially secure SKFE to polynomially secure SXIO with compression factor γ .*

We add more technical details regarding the proof of the above SXIO proposition later on. Both of our theorems stated above rely on the constructed SXIO as a main tool. We next explain, still at a high-level, how the first theorem is obtained. We then dive

into further technical details about the proof of this theorem as well as the proof of the second theorem.

From SXIO to IO through public-key encryption Subexponentially secure SXIO (or even XIO) schemes with a constant compression factor (as in Proposition 1.1) are already shown to be quite strong in [51]—assuming subexponential hardness of learning with errors (LWE) [56], they imply IO.

Corollary 1.1 (of Proposition 1.1 and [51]) *Assuming SKFE and LWE, both subexponentially secure, there exists IO.*

We go beyond the above corollary, showing that LWE can be replaced with a generic assumption—the existence of (plain) public-key encryption schemes. The transformation of [51] from LWE and XIO to IO essentially relies on LWE to obtain a specific type of public-key functional encryption (PKFE) with certain succinctness properties. We show how to construct such PKFE from public-key encryption and SXIO. More details follow.

Concretely, the notion considered is of PKFE schemes that support a *single decryption key*. Furthermore, the time complexity of encryption is bounded by roughly $s^\beta \cdot d^{O(1)}$, where s and d are the size and depth of the circuit computing the function, and $\beta < 1$ is some compression factor. We call such schemes *weakly succinct PKFE schemes*. A weakly succinct PKFE for *boolean* functions (i.e., functions with a single output bit) is constructed by Goldwasser et al. [33] from (subexponentially hard) LWE; in fact, the Goldwasser et al. construction has no dependence at all on the circuit size s (namely, $\beta = 0$).

Lin et al. [51] then show a transformation, relying on XIO, that extends the class of functions also to functions with a long output, rather than just boolean ones (their transformation is stated for the case that $\beta = 0$ assuming any constant XIO compression factor $\gamma < 1$, but can be extended to also work for any sufficiently small constant compression factor β for the PKFE). Such weakly succinct PKFE schemes can then be plugged in to the transformations of [3, 22, 52] to obtain full-fledged IO.¹

We follow a similar blueprint. We first construct weakly succinct PKFE for functions with a single output bit based on SXIO and PKE, rather than LWE (much of the technical effort in this work lies in this construction). We then bootstrap the construction to deal with multi-bit functions using (a slightly augmented version of) the transformation from [51].

Proposition 1.2. (Informal) *For any $\beta = \Omega(1)$, assuming PKE and SXIO with a small enough constant compression factor γ , there exists a single-key weakly succinct PKFE scheme with compression factor β (for functions with long output). We summarize how to construct IO from SKFE and PKE in Fig 1.*

¹The above is a slightly oversimplified account of [51]. They also rely on LWE to deduce the existence of puncturable PRFs in NC^1 and show their transformation starting from weakly succinct PKFE for functions in NC^1 . We avoid the reliance on puncturable PRFs in NC^1 by constructing weakly succinct PKFE for functions with no depth restriction, at the expense of allowing the complexity of encryption to scale polynomially in the depth. This is still sufficient for [22, Section 3.2].

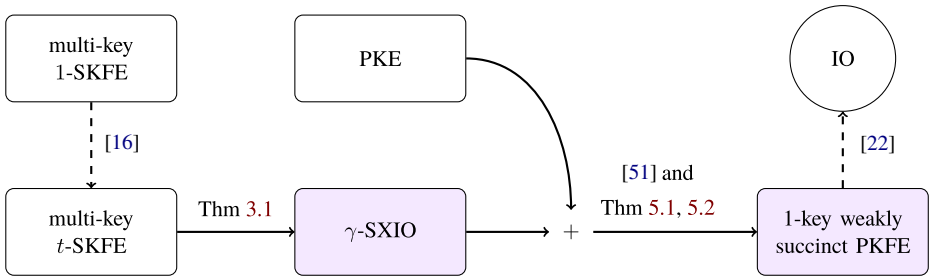


Fig. 1. An illustration of our result on IO. Dashed lines denote known results. White backgrounds denote our ingredients or goal. Primitives in rounded rectangles are subexponentially secure. t -SKFE denotes t -input SKFE. γ -SXIO denotes SXIO with compression factor γ , which is an arbitrary constant less than 1.

A Closer Look into the Techniques

We now provide further details regarding the proofs of Propositions 1.1 and 1.2 as well as the proof of Theorem 1.2.

SKFE to SXIO: the basic idea To convey the basic idea behind the transformation, we first describe a construction of SXIO with compression $\gamma = 1/2$. We then explain how to extend it to obtain the more general form of Proposition 1.1.

Recall that in an SKFE scheme, first a master secret key MSK is generated, and can then be used to:

- encrypt (any number of) plaintext messages,
- derive (any number of) functional keys.

The constructed obfuscator $sxiO$ is given a circuit C defined on domain $\{0, 1\}^n$, where we shall assume for simplicity that the input length is even (this is not essential), and works as follows:

- For every $x \in \{0, 1\}^{n/2}$, computes a ciphertext CT_x encrypting the circuit $C_x(\cdot)$ that given input $y \in \{0, 1\}^{n/2}$, returns $C(x, y)$.
- For every $y \in \{0, 1\}^{n/2}$, derives a functional decryption key SK_y for the function $U_y(\cdot)$ that given as input a circuit D of size at most $\max_x |C_x|$, returns $D(y)$.
- Outputs $\tilde{C} = \left(\{CT_x\}_{x \in \{0, 1\}^{n/2}}, \{SK_y\}_{y \in \{0, 1\}^{n/2}} \right)$ as the obfuscation.

To evaluate \tilde{C} on input $(x, y) \in \{0, 1\}^n$, simply decrypt

$$\text{Dec}(SK_y, CT_x) = U_y(C_x) = C_x(y) = C(x, y) .$$

Indeed, the required compression factor $\gamma = 1/2$ is achieved. Generating each ciphertext is proportional to the size of the message $|C_x| = \tilde{O}(|C|)$ and some fixed polynomial in the security parameter λ . Similarly, the time to generate each functional key is proportional to the size of the circuit $|U_y| = \tilde{O}(|C|)$ and some fixed polynomial in the security parameter λ . Thus overall, the time to generate \tilde{C} is bounded by $2^{n/2} \cdot \text{poly}(|C|, \lambda)$.

The indistinguishability guarantee follows easily from that of the underlying SKFE. Indeed, SKFE guarantees that for any two sequences $m = \{m_i\}$ and $m' = \{m'_i\}$ of messages to be encrypted and any sequence of functions $\{f_i\}$ for which keys are derived,

encryptions of the m are indistinguishable from encryptions of the m' , provided that the messages are not “separated by the functions,” i.e., $f_j(m_i) = f_j(m'_i)$ for every (i, j) . In particular, any two circuits C and C' that have equal size and functionality will correspond to such two sequences of messages $\{C_x\}_{x \in \{0,1\}^{n/2}}$ and $\{C'_x\}_{x \in \{0,1\}^{n/2}}$, whereas $\{U_y\}_{y \in \{0,1\}^n}$ are indeed functions such that $U_y(C_x) = C(x, y) = C'(x, y) = U_y(C'_x)$ for all (x, y) . (The above argument works even given a very weak selective security definition where all messages and functions are chosen by the attacker ahead of time.) As said, the above transformation achieves compression factor $\gamma = 1/2$. While such compression is sufficient, for example, to obtain IO based on LWE, it will not suffice for our two Theorems 1.1 and 1.2 (for the first we will need γ to be a smaller constant, and for the second we will need it to even be slightly subconstant). To prove Proposition 1.1 in its more general form, we rely on a result by Brakerski et al. [16] that shows how to convert any SKFE into a t -input SKFE. A t -input scheme allows to encrypt a tuple of messages (m_1, \dots, m_t) each independently and derive keys for t -input functions $f(m_1, \dots, m_t)$. In their transformation, starting from a multi-key SKFE results in a multi-key t -input SKFE.

The general transformation then follows naturally. Rather than arranging the input space in a 2-dimensional cube $\{0, 1\}^{n/2} \times \{0, 1\}^{n/2}$ as we did before with a 1-input scheme, given a t -input scheme we can arrange it in a $(t + 1)$ -dimensional cube $\{0, 1\}^{n/(t+1)} \times \dots \times \{0, 1\}^{n/(t+1)}$, and we will accordingly get compression $\gamma = 1/(t+1)$. The only caveat is that the BKS transformation incurs a security loss and blowup in the size of the scheme that can grow doubly exponentially in t . As long as t is constant, the security loss and blowup are fixed polynomials. The transformation can also be invoked for slightly super-constant t (double logarithmic) assuming subexponential security of the underlying 1-input SKFE (giving rise to the second part of Proposition 1.1).

We remark that previously Goldwasser et al. [27] showed that t -input SKFE for polynomial t directly gives full-fledged IO. We demonstrate that even when t is small (even constant), t -input SKFE implies a meaningful obfuscation notion such as SXIO. **From SXIO and PKE to weakly succinct PKFE: main ideas** We now describe the main ideas behind our construction of a single-key weakly succinct PKFE. We shall focus on the main step of obtaining such a scheme for functions with a single output bit.²

Our starting point is the single-key PKFE scheme of Sahai and Seyalioglu [58] based on Yao’s garbled circuit method [61]. Their scheme basically works as follows (we assume basic familiarity with the garbled circuit method):

- The master public key MPK consists of L pairs of public keys $\left\{ \text{PK}_i^0, \text{PK}_i^1 \right\}_{i \in L}$ for a (plain) public-key encryption scheme.

²Extending this to functions with multi-bit output is then done, based on SXIO, using a transformation of [51]. Concretely, given an m -bit output function $f(x)$ we consider a new single bit function $g_f(x, i)$ that returns the i th bit of $f(x)$. The function key is then derived for the boolean function g_f . The new encryption algorithm, for message x , produces an SXIO obfuscation of a circuit that given $i \in [m]$ uses the old encryption scheme to encrypt (m, i) , deriving randomness using a puncturable PRF. The security of the construction is proven as in [51] based on a probabilistic IO argument [25]. (Mild) efficiency of the encryption then follows from the mild efficiency of the SXIO and PKFE with related (constant) compression factors.

- A functional decryption key \mathbf{SK}_f for a function (circuit) f of size L consists of the secret decryption keys $\{\mathbf{SK}_i^{f_i}\}_{i \in L}$ corresponding to the above public keys, according to the bits of f 's description.
- To encrypt a message m , the encryptor generates a garbled circuit \widehat{U}_m for the universal circuit U_m that given f , returns $f(m)$. It then encrypts the corresponding input labels $\{k_i^0, k_i^1\}_{i \in L}$ under the corresponding public keys.
- The decryptor in possession of \mathbf{SK}_f can then decrypt to obtain the labels $\{k_i^{f_i}\}_{i \in L}$ and decode the garbled circuit to obtain $U_m(f) = f(m)$.

Selective security of this scheme (where the function f and all messages are chosen ahead of time) follows from the semantic security of PKE and the garbled circuit guarantee which says that $\widehat{U}_m, \{k_i^{f_i}\}_{i \in L}$ can be simulated from $f(m)$.

The scheme is indeed *not* succinct in any way. The complexity of encryption and even the size of the ciphertext grows with the complexity of f . Nevertheless, it does seem that the encryption process has a much more succinct representation. In particular, computing a garbled circuit is a *decomposable* process—each garbled gate in \widehat{U}_m depends on a single gate in the original circuit U_m and a small amount of randomness (for computing the labels corresponding to its wires). Furthermore, the universal circuit U_m itself is also decomposable—there exists a small (say, $\text{poly}(|m|, \log L)$ -sized) circuit that given i can output the i -th gate in U_m along with its neighbors. The derivation of randomness itself can also be made decomposable using a pseudorandom function. All in all, there exists a small ($\text{poly}(|m|, \log L, \lambda)$ -size, for security parameter λ), *decomposition circuit* $U_{m,K}^{\text{de}}$ associated with a key $K \in \{0, 1\}^\lambda$ for a pseudorandom function that can produce the i th garbled gate/input-label given input i .

Yet, the second part of the encryption process, where the input labels $\{k_i^0, k_i^1\}_{i \in L}$ are encrypted under the corresponding public keys $\{\mathbf{PK}_i^0, \mathbf{PK}_i^1\}_{i \in L}$, may not be decomposable at all. Indeed, in general, it is not clear how to even compress the representation of these $2L$ public keys. In this high-level exposition, let us make the simplifying assumption that we have at our disposal a *succinct identity-based-encryption (IBE) scheme*. Such a scheme has a single public-key \mathbf{PK} that allows to encrypt a message to an identity $\text{id} \in \mathcal{ID}$ taken from an identity space \mathcal{ID} . Those in possession of a corresponding secret key \mathbf{SK}_{id} can decrypt and others learn nothing. Succinctness means that the complexity of encryption may only grow mildly in the size of the identity space. Concretely, by a factor of $|\mathcal{ID}|^\gamma$ for some small constant $\gamma < 1$. In the body, we show that such a scheme can be constructed from (plain) public-key encryption and SXIO. (The construction relies on standard “puncturing techniques” and is pretty natural, see Sect. 5.1 for more details.)

Equipped with such an IBE scheme, we can now augment the Sahai–Selyalioglu scheme to make sure that the entire encryption procedure is decomposable. Concretely, we will consider the identity space $\mathcal{ID} = [L] \times \{0, 1\}$, augment the public key to only include the IBE's public key \mathbf{PK} , and provide the decryptor with the identity keys $\{\mathbf{SK}_{(i, f_i)}\}_{i \in L}$. Encrypting the input labels $\{k_i^0, k_i^1\}_{i \in L}$ will now be done by simply encrypting to the corresponding identities $\{(i, 0), (i, 1)\}_{i \in L}$. This part of the encryption can now also be described by a small (say $L^\gamma \cdot \text{poly}(\lambda, \log L)$ -size) decomposition circuit $E_{K, K', \mathbf{PK}}^{\text{de}}$ that has the PRF key K to derive input labels, the IBE public key \mathbf{PK} , and

another PRF key K' to derive randomness for encryption. Given an identity (i, b) , it generates the corresponding encrypted input label.

At this point, a natural direction is to have the encryptor send a *compressed* version of the Sahai–Selyalioglu encryption, by first using SXIO to shield the two decomposition circuits $E_{K, K', PK}^{\text{de}}, U_{m, K}^{\text{de}}$ and then sending the two obfuscations. Indeed, decryption can be done just as before by first reconstructing the expanded garbled circuit and input labels and then proceeding as before. Also, in terms of encryption complexity, provided that the IBE compression factor γ is a small enough constant, the entire encryption time will scale only sublinearly in the function's size $|f| = L$ (i.e., with L^β for some constant $\beta < 1$).

The only question is of course security. It is not too hard to see that if the decomposition circuits $E_{K, K', PK}^{\text{de}}, U_{m, K}^{\text{de}}$ are given as black boxes, then security is guaranteed just as before. The challenge is to prove security relying only on the indistinguishability guarantee of SXIO. A somewhat similar challenge is encountered in the work of Bitansky et al. [8, 13] when constructing *succinct randomized encodings*. In their setting, they obfuscate (using standard IO rather than SXIO) a decomposition circuit $C_{x, K}^{\text{de}}$ (analogous to our $U_{m, K}^{\text{de}}$) that computes the garbled gates of some succinctly represented long computation.

As already demonstrated in [8, 13], proving the security of such a construction is rather delicate. As in the standard setting of garbled circuits, the goal is to gradually transition through a sequence of hybrids, from a real garbled circuit (that depends on the actual computation) to a simulated garbled circuit that depends just on the result of the computation. However, unlike the standard setting of garbled circuits, here each of these hybrids should be generated by a *hybrid obfuscated decomposition circuit* and the attacker should not be able to tell the hybrid circuits apart.

A feature of the hybrid strategy which is crucial in this context is the amount of information that hybrid decomposition circuits need to maintain about the actual computation. Indeed, as the amount of this information grows so will the size of these decomposition circuits as will the size of the decomposition circuits in the actual construction (that will have to be equally padded to preserve indistinguishability). Accordingly, this feature affects the succinctness of the entire construction.

Bitansky et al. [8, 13] show a hybrid strategy where the amount of information scales with the *space* of the computation (or *circuit width*). Whereas in their context this is meaningful (as the aim is to save comparing to the *time* of the computation), in our context this is clearly insufficient. Indeed, in our case the space of the computation given by the universal circuit U_m and the function f can be as large as f 's description. Instead, we invoke a different hybrid strategy by Hemenway et al. [39] that scales only with the *circuit depth*. Indeed, this is the cause for the polynomial dependence on depth in our single-key PKFE construction. Below, we further elaborate on the Hemenway et al. hybrid strategy and how it is imported into our setting.

Decomposable Garbling and Pebbling The work of Hemenway et al. [39] provided a useful abstraction for proving the security of Yao's garbled circuits via a sequence of hybrid games. The goal is to transition from a “real” garbled circuit, where each garbled gate is in “RealGate” mode consisting of four ciphertexts encrypting the two labels k_c^0, k_c^1 of the output wire c under the labels of the input wires, to a “simulated” garbled

circuit where each garbled gate is in **SimGate** mode consisting of four ciphertexts that all encrypt the same dummy label k_c^0 . As an intermediate step, we can also create a garbled gate in **CompDepSimGate** mode consisting of four ciphertexts encrypting the same label $k_c^{v(c)}$ where $v(c)$ is the value going over wire c during the computation $C(x)$ and therefore depends on the actual computation.

The transition from a real garbled circuit to a simulated garbled circuit proceeds via a sequence of hybrids where in each subsequent hybrid we can change one gate at a time from **RealGate** to **CompDepSimGate** (and vice versa) if all of its predecessors are in **CompDepSimGate** mode or it is an input gate, or change a gate from **CompDepSimGate** mode to **SimGate** mode (and vice versa) if all of its successors are in **CompDepSimGate** or **SimGate** modes. The goal of Hemenway et al. was to give a strategy using the least number of gates in **CompDepSimGate** mode as possible.³ They abstracted this problem as a pebbling game and show that for circuits of depth d there exists a sequence of $2^{O(d)}$ hybrids with at most $O(d)$ gates in **CompDepSimGate** mode in any single hybrid.

In our case, we can give a decomposable circuit for each such hybrid game consisting of gates in **RealGate**, **SimGate**, **CompDepSimGate** modes. In particular, the decomposable circuit takes as input a gate index and outputs the garbled gate in the correct mode. We only need to remember which gate is in which mode, and for all gates in **CompDepSimGate** mode we need to remember the bit $v(c)$ going over the wire c during the computation $C(x)$. It turns out that the configuration of which mode each gate is in can be represented succinctly, and therefore the number of bits we need to remember is roughly proportional to the number of gates in **CompDepSimGate** mode in any given hybrid. Therefore, for circuits of depth d , the decomposable circuit is of size $O(d)$ and the number of hybrid steps is $2^{O(d)}$.

To ensure that the obfuscations of decomposable circuits corresponding to neighboring hybrids are indistinguishable, we also need to rely on standard puncturing techniques. In particular, the gates are garbled using a punctured PRF and we show that in any transition between neighboring hybrids we can even give the adversary the PRF key punctured only on the surrounding of the gate whose mode is changed.

On the Security Loss We note that our transformation from SXIO and PKE to weakly succinct PKFE only incurs a polynomial security loss when considering logarithmic depth circuits. Subexponential security of SXIO and PKE in Theorem 1.1 is required in order to use the known transformations from PKFE to IO, which require subexponential security [3, 22].

From SKFE to PKE: the basic idea We end our technical exposition by explaining the basic idea behind the construction of public-key encryption (PKE) from SKFE. The construction is rather natural. Using subexponentially secure SKFE and the second part of Proposition 1.1, we can obtain a $\text{poly}(\lambda)$ -secure SXIO with a subconstant compression factor $\gamma = o(1)$; concretely, it can be, for example, $O(1/\log \log \lambda)$. We can now think about this obfuscator as a plain (efficient) indistinguishability obfuscator for circuits with input length at most $\log \lambda \cdot \log \log \lambda$.

³Their aim was proving adaptive security, which is completely orthogonal to our aim. However, for entirely different reasons, the above goal is useful in both their work and ours.

Then, we take a construction of public-key encryption from IO and one-way functions where the input size of obfuscated circuits can be scaled down at the expense of strengthening the one-way functions. For instance, following the basic *witness encryption paradigm* in [31], the public key can be a pseudorandom string $\text{PK} = \text{PRG}(s)$ for a $2^{n/\log \log n}$ -secure length-doubling pseudorandom generator with seed length $n = \log \lambda \cdot \log \log \lambda$. Here, the obfuscator is only invoked for a circuit with inputs in $\{0, 1\}^n$. An encryption of m is simply an obfuscation of a circuit that has PK hardwired, and releases m only given a seed s such that $\text{PK} = \text{PRG}(s)$. Security follows essentially as in [31]. Note that in this construction, we cannot expect more than 2^n security, which is quasi-polynomial in the security parameter λ .

How does the construction circumvent the Asharov-Segev barrier? As noted earlier, Asharov and Segev [6] show that even exponentially secure SKFE cannot lead to public-key encryption through a fully black-box construction (see their paper for details about the exact model). The reason that our construction does not fall under their criteria lies in the transformation from SKFE to SXIO with subconstant compression, and concretely in the Brakerski–Komargodski–Segev [16] transformation from SKFE to t -input SKFE that makes non-black-box use in the algorithms of the underlying SKFE scheme.

1.3. Subsequent Works

We now mention several follow-up works, including ones that have relied on our work for new application and ones that have extended our work.

- New Transformations** Several works gave different improved transformations from SKFE to IO. Komargodski and Segev [46] improved the known transformation [16] from single-input SKFE to multi-input SKFE. Via the improved transformation, they can improve the direct IO construction we described above, which avoids the reliance on public-key encryption. Starting from quasi-polynomially SKFE, they obtain IO for circuits of sub-polynomial size and inputs of polylogarithmic. Kitagawa et al. [44] present a simpler transformation from SKFE and public-key encryption to PKFE (and IO) than the one presented here. Then, in a follow-up work, the same set of authors [42, 43] removed the reliance on public-key encryption altogether. That is, they transform any subexponentially secure multi-key SKFE to IO for all circuits (without a restriction on their size or input length). In another follow-up work, the same set of authors [41, 43] replace multi-key SKFE with single-key weakly succinct SKFE in the transformation above by showing a transformation from any single-key weakly succinct SKFE to multi-key (succinct) SKFE.
- IO from SKFE** A progression of works [7, 48, 49, 53, 54] has shown how to reduce the degree of multi-linear maps required for constructing IO. In these constructions, the core ingredient is a construction of FE for low-degree polynomials, which is then bootstrapped to full-fledged FE and IO. Relying on our work, it is sufficient to construct such schemes in the secret-key setting, which allows reducing the required degree to *three* in the state of the art (assuming also appropriate pseudorandom generators) [49].
- Non-Trivial null XIO from Standard Assumptions** Brakerski et al. [15] show that our key technique can actually be applied to existing constructions of attribute-

based, or predicate, encryption. This yields non-trivial notions of exponential witness encryption (XWE) and null XIO for NC^1 , based on standard assumptions like learning with errors.

- **FE with Ideal Multi-Linear Maps** Bitansky, Lin, and Paneth [17] prove that any construction of FE schemes that are succinct enough that uses constant degree multi-linear maps in a black-box way, can be transformed into a scheme based on bilinear maps. For this, they use XIO as a main component and crucially rely on the fact that our XIO construction from FE uses the underlying FE as a black box.

Organization In Sect. 2, we provide preliminaries and basic definitions used throughout the paper. In Sect. 3, we introduce the definition of SXIO and present our construction based on SKFE schemes. In Sect. 4, we review Yao’s garbled circuits and introduce a notion of decomposable garbling. In Sect. 5, we present our construction of IO from PKE and SXIO. In Sect. 6, we present a polynomially secure PKE scheme from SKFE schemes.

2. Preliminaries

We review basic concepts and definitions used throughout the paper.

2.1. Standard Computational Concepts

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

- We say that a (uniform) Turing machine is PPT if it is probabilistic and runs in polynomial time.
- A polynomial-size (or just polysize) circuit family \mathcal{C} is a sequence of circuits $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$, such that each circuit C_λ is of polynomial size $\lambda^{O(1)}$ and has $\lambda^{O(1)}$ input and output bits.
- We follow the standard habit of modeling any efficient adversary strategy as a family of polynomial-size circuits. For an adversary \mathcal{A} corresponding to a family of polysize circuits $\{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$, we often omit the subscript λ , when it is clear from the context.
- We say that a function $f : \mathbb{N} \rightarrow \mathbb{R}$ is negligible if for all constants $c > 0$, there exists $N \in \mathbb{N}$ such that for all $n > N$, $f(n) < n^{-c}$.
- If $\mathcal{X}^{(b)} = \{X_\lambda^{(b)}\}_{\lambda \in \mathbb{N}}$ for $b \in \{0, 1\}$ are two ensembles of random variables indexed by $\lambda \in \mathbb{N}$, we say that $\mathcal{X}^{(0)}$ and $\mathcal{X}^{(1)}$ are computationally indistinguishable if for all polysize distinguishers \mathcal{D} , there exists a negligible function ν such that for all λ ,

$$\Delta = \left| \Pr[\mathcal{D}(X_\lambda^{(0)}) = 1] - \Pr[\mathcal{D}(X_\lambda^{(1)}) = 1] \right| \leq \nu(\lambda).$$

- We write $\mathcal{X}^{(0)} \approx_\delta \mathcal{X}^{(1)}$ to denote that the advantage Δ is bounded by δ .

2.2. Functional Encryption

In this section, we define the different notions of functional encryption (FE) considered in this work.

Definition 2.1. (*Multi-input secret-key functional encryption*) Let $t(\lambda)$ be a function, $\overline{\mathcal{M}} = \{\overline{\mathcal{M}}_\lambda = \mathcal{M}_\lambda^{(1)} \times \cdots \times \mathcal{M}_\lambda^{(t(\lambda))}\}_{\lambda \in \mathbb{N}}$ be a product message domain, $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$ a range, and $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ a class of t -input functions $f : \overline{\mathcal{M}}_\lambda \rightarrow \mathcal{Y}_\lambda$. A t -input secret-key functional encryption (t -SKFE) scheme for $\mathcal{M}, \mathcal{Y}, \mathcal{F}$ is a tuple of algorithms $\text{SKFE}_t = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ where:

- $\text{Setup}(1^\lambda)$ takes as input the security parameter and outputs a master secret key MSK .
- $\text{KeyGen}(\text{MSK}, f)$ takes as input the master secret MSK and a function $f \in \mathcal{F}$. It outputs a secret key SK_f for f .
- $\text{Enc}(\text{MSK}, m, i)$ takes as input the master secret key MSK , a message $m \in \mathcal{M}_\lambda^{(i)}$, and an index $i \in [t(\lambda)]$, and outputs a ciphertext CT_i .
- $\text{Dec}(\text{SK}_f, \text{CT}_1, \dots, \text{CT}_t)$ takes as input the secret key SK_f for a function $f \in \mathcal{F}$ and ciphertexts $\text{CT}_1, \dots, \text{CT}_t$, and outputs some $y \in \mathcal{Y}$, or \perp .

We also require the following property: **Correctness:** For all tuples $\mathbf{m} = (m_1, \dots, m_t) \in \overline{\mathcal{M}}_\lambda$ and any function $f \in \mathcal{F}_\lambda$, we have that

$$\Pr \left[\begin{array}{l} \text{MSK} \leftarrow \text{Setup}(1^\lambda), \\ \text{Dec}(\text{SK}_f, \text{CT}_1, \dots, \text{CT}_t) = f(\mathbf{m}) : \text{SK}_f \leftarrow \text{KeyGen}(\text{MSK}, f), \\ \forall i \text{ CT}_i \leftarrow \text{Enc}(\text{MSK}, m, i) \end{array} \right] = 1$$

Definition 2.2. (*Selectively secure multi-key t -SKFE*) We say that a tuple of algorithms $\text{SKFE}_t = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ is a *selectively secure* t -input secret-key functional encryption scheme for $\overline{\mathcal{M}}, \mathcal{Y}, \mathcal{F}$, if it satisfies the following requirement, formalized by the experiment $\text{Expt}_{\mathcal{A}}^{\text{SKFE}_t}(1^\lambda, b)$ between an adversary \mathcal{A} and a challenger:

1. The adversary submits challenge message tuples $\left\{ (m_{i,1}^0, m_{i,1}^1, i) \right\}_{i \in [t]}, \dots, \left\{ (m_{i,q}^0, m_{i,q}^1, i) \right\}_{i \in [t]}$ for all $i \in [t]$ to the challenger where q is an arbitrary polynomial in λ .
2. The challenger runs $\text{MSK} \leftarrow \text{Setup}(1^\lambda)$
3. The challenger generates ciphertexts $\text{CT}_{i,j} \leftarrow \text{Enc}(\text{MSK}, m_{i,j}^b, i)$ for all $i \in [t]$ and $j \in [q]$, and gives $\{\text{CT}_{i,j}\}_{i \in [t], j \in [q]}$ to \mathcal{A} .
4. \mathcal{A} is allowed to make q function queries, where it sends a function $f_j \in \mathcal{F}$ to the challenger for $j \in [q]$ and q is an arbitrary polynomial in λ . The challenger responds with $\text{SK}_{f_j} \leftarrow \text{KeyGen}(\text{MSK}, f_j)$.
5. \mathcal{A} outputs a guess b' for b .
6. The output of the experiment is b' if the adversary's queries are valid:

$$f_j(m_{1,j_1}^0, \dots, m_{t,j_t}^0) = f_j(m_{1,j_1}^1, \dots, m_{t,j_t}^1) \text{ for all } j_1, \dots, j_t, j \in [q].$$

Otherwise, the output of the experiment is set to be \perp .

We say that the functional encryption scheme is selectively secure if, for all polysize adversaries \mathcal{A} , there exists a negligible function $\mu(\lambda)$, such that

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

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

We recall the following theorem by Brakerski, Komargodski, and Segev, which states that one can build selectively secure t -SKFE from any selectively secure 1-SKFE. The transformation induces a significant blowup and security loss in the number of inputs t . This loss is polynomial as long as t is constant, but in general grows doubly exponentially in t .

Theorem 2.1. ([16])

1. For $t = O(1)$, if there exists δ -selectively secure single-input SKFE for P/poly , then there exists δ -selectively secure t -input SKFE for P/poly .
2. There exists a constant $\varepsilon < 1$, such that for $t(\lambda) = \varepsilon \cdot \log \log(\lambda)$, $\tilde{\lambda} = 2^{(\log \lambda)^\varepsilon}$, $\delta(\tilde{\lambda}) = 2^{-\tilde{\lambda}^\varepsilon}$, if there exists δ -selectively secure single-input SKFE for P/poly , then there exists polynomially secure selectively secure t -input SKFE for functions of size at most $2^{O((\log \lambda)^\varepsilon)}$. (Here, $\tilde{\lambda}$ is the single-input SKFE security parameter and λ is the t -input SKFE security parameter.)

Remark 2.1. (Dependence on circuit size in [16]) The [16] transformation incurs a $(s \cdot \tilde{\lambda})^{2^{O(t)}}$ blowup in parameters, where s is the size of maximal circuit size of supported functions and $\tilde{\lambda}$ is the security parameter used in the underlying single-input SKFE. In the main setting of parameters considered there, $t = O(1)$, the security parameter λ of the t -SKFE scheme can be identified with $\tilde{\lambda}$ and s can be any polynomial in this security parameter. (Accordingly, the dependence on s is implicit there, and the blowup they address is $\lambda^{2^{O(t)}}$.)

For the second part of the theorem, to avoid superpolynomial blowup in λ , the security parameter $\tilde{\lambda}$ for the underlying SKFE and the maximal circuit size s should be set to $2^{O((\log \lambda)^\varepsilon)}$.

Definition 2.3. (Public-key functional encryption) Let $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ be a message domain, $\mathcal{Y} = \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$ a range, and $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ a class of functions $f : \mathcal{M} \rightarrow \mathcal{Y}$. A public-key functional encryption (PKFE) scheme for $\mathcal{M}, \mathcal{Y}, \mathcal{F}$ is a tuple of algorithms $\text{PKFE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ where:

- $\text{Setup}(1^\lambda)$ takes as input the security parameter and outputs a master secret key MSK and master public key MPK.
- $\text{KeyGen}(\text{MSK}, f)$ takes as input the master secret MSK and a function $f \in \mathcal{F}$. It outputs a secret key SK_f for f .

- $\text{Enc}(\text{MPK}, m)$ takes as input the master public key MPK and a message $m \in \mathcal{M}$, and outputs a ciphertext c .
- $\text{Dec}(\text{SK}_f, c)$ takes as input the secret key SK_f for a function $f \in \mathcal{F}$ and a ciphertext c , and outputs some $y \in \mathcal{Y}$, or \perp .

We also require the following property: **Correctness:** For any message $m \in \mathcal{M}$ and function $f \in \mathcal{F}$, we have that

$$\Pr \left[\begin{array}{l} (\text{MSK}, \text{MPK}) \leftarrow \text{Setup}(1^\lambda), \\ \text{Dec}(\text{SK}_f, c) = f(m) : \text{SK}_f \leftarrow \text{KeyGen}(\text{MSK}, f), \\ c \leftarrow \text{Enc}(\text{MPK}, m) \end{array} \right] = 1$$

Definition 2.4. (*Selectively secure single-key PKFE*) We say that a tuple of algorithm $\text{PKFE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ is a *selectively secure single-key* public-key functional encryption scheme for $\mathcal{M}, \mathcal{Y}, \mathcal{F}$, if it satisfies the following requirement, formalized by the experiment $\text{Expt}_{\mathcal{A}}^{\text{PKFE}}(1^\lambda, b)$ between an adversary \mathcal{A} and a challenger:

1. \mathcal{A} submits the message pair $m_0^*, m_1^* \in \mathcal{M}$ and a function f to the challenger.
2. The challenger runs $(\text{MSK}, \text{MPK}) \leftarrow \text{Setup}(1^\lambda)$, generates ciphertext $\text{CT}^* \leftarrow \text{Enc}(\text{MPK}, m_b^*)$ and a secret key $\text{SK}_f \leftarrow \text{KeyGen}(\text{MSK}, f)$. The challenger gives $(\text{MPK}, \text{CT}^*, \text{sk}_f)$ to \mathcal{A} .
3. \mathcal{A} outputs a guess b' for b .
4. The output of the experiment is b' if $f(m_0^*) = f(m_1^*)$ and \perp otherwise.

We say that the public-key functional encryption scheme is *selectively secure* if, for all PPT adversaries \mathcal{A} , there exists a negligible function $\mu(\lambda)$, such that

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

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

We now further define a notion of succinctness for functional encryption schemes as above. To do this, we define functional encryption for collections of classes $\mathcal{F} = \{\mathcal{F} = \{\mathcal{F}_\lambda\}\}$, such as P/poly or NC^1 .

Definition 2.5. (*Functional Encryption for a Collection of Functions* [17]) In a functional encryption scheme $\text{FE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for a collection of function classes $\mathcal{F} = \{\mathcal{F} = \{\mathcal{F}_\lambda\}\}$, the setup algorithm $\text{Setup}(1^\lambda, n, s)$ takes the security parameter 1^λ and two parameters $(n(\lambda), s(\lambda))$ representing bounds on input length and circuit size.

For any class $\mathcal{F} = \{\mathcal{F}_\lambda\} \in \mathcal{F}$ with maximum input length $n(\lambda)$ and maximum circuit size $s(\lambda)$, $\text{FE}_{\mathcal{F}} = (\text{Setup}(\cdot, n(\cdot), s(\cdot)), \text{KeyGen}, \text{Enc}, \text{Dec})$ satisfies the correctness and security requirement defined above for general functional encryption schemes.

Definition 2.6. (*Weakly Succinct functional encryption* [17,22]) A functional encryption scheme for a collection of function classes $\mathcal{F} = \{\mathcal{F} = \{\mathcal{F}_\lambda\}\}$ is *weakly succinct* if

there exists a constant $\gamma < 1$, and a fixed polynomial $\text{poly}(\cdot)$, depending only on \mathcal{F} (but not on any specific $\mathcal{F} \in \mathcal{F}$), such that for any class $\mathcal{F} = \{\mathcal{F}_\lambda\} \in \mathcal{F}$ with

- maximum input length $n(\lambda)$,
- maximum circuit size $s(\lambda)$, and
- maximum circuit depth $d(\lambda)$,

in $\text{FE}_{\mathcal{F}}$, the size of the encryption circuit is bounded by $s^\gamma \cdot \text{poly}(n, \lambda, d)$. We call γ the compression factor.

The following result from [22, Sections 3.1 and 3.2] states that one can construct an indistinguishability obfuscator from any single-key weakly succinct public-key functional encryption scheme.

Theorem 2.2. ([22]) *If there exists a subexponentially secure single-key weakly succinct PKFE scheme for P/poly , then there exists an indistinguishability obfuscator for P/poly .*

2.3. Indistinguishability Obfuscation

We recall the notion of indistinguishability obfuscation (IO) [10].

Definition 2.7. (*Indistinguishability obfuscator (IO)*) A PPT machine iO is an *indistinguishability obfuscator* for a circuit class $\{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if the following conditions are satisfied:

- **Functionality** for all security parameters $\lambda \in \mathbb{N}$, for all $C \in \mathcal{C}_\lambda$, for all inputs x , we have that

$$\Pr[C'(x) = C(x) : C' \leftarrow \text{iO}(C)] = 1 .$$

- **Indistinguishability** for any polysize distinguisher \mathcal{D} , there exists a negligible function $\mu(\cdot)$ such that the following holds: for all security parameters $\lambda \in \mathbb{N}$, for all pairs of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ of the same size and such that $C_0(x) = C_1(x)$ for all inputs x , then

$$|\Pr[\mathcal{D}(\text{iO}(C_0)) = 1] - \Pr[\mathcal{D}(\text{iO}(C_1)) = 1]| \leq \mu(\lambda) .$$

We further say that iO 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. Puncturable Pseudorandom Functions

Puncturable PRFs, defined by Sahai and Waters [59], are PRFs with a key-puncturing procedure that produces keys that allow evaluation of the PRF on all inputs, except for a designated polynomial-size set.

Definition 2.8. (*Puncturable pseudorandom function*) For sets \mathcal{D}, \mathcal{R} , a *puncturable pseudorandom function* (PPRF) consists of a tuple of algorithms $\mathcal{PPRF} = (\text{PRF.Gen}, \text{PRF.Ev}, \text{PRF.Punc})$ that satisfy the following two conditions.

Functionality preserving under puncturing: For all polynomial-size set $S \subseteq \mathcal{D}$ and for all $x \in \mathcal{D} \setminus S$, it holds that

$$\begin{aligned} \Pr[\text{PRF.Ev}_K(x) = \text{PRF.Ev}_{K\{S\}}(x) : K \leftarrow \text{PRF.Gen}(1^\lambda), \\ K\{S\} \leftarrow \text{PRF.Punc}(K, S)] = 1. \end{aligned}$$

Pseudorandom at punctured points: For all polynomial size set $S \subseteq \mathcal{D}$ with $S = \{x_1, \dots, x_{k(\lambda)}\}$ and any polysize distinguisher \mathcal{A} , there exists a negligible function μ , such that:

$$\begin{aligned} |\Pr[\mathcal{A}(\text{PRF.Ev}_{K\{S\}}, \{\text{PRF.Ev}_K(x_i)\}_{i \in [k]}) = 1] \\ - \Pr[\mathcal{A}(\text{PRF.Ev}_{K\{S\}}, U^k) = 1]| \leq \mu(\lambda) \end{aligned}$$

where $K \leftarrow \text{PRF.Gen}(1^\lambda)$, $K\{S\} \leftarrow \text{PRF.Punc}(K, S)$ and U denotes the uniform distribution over \mathcal{R} . We further say that \mathcal{PPRF} is δ -secure, for some concrete negligible function $\delta(\cdot)$, if for all polysize distinguishers the above indistinguishability gap $\mu(\lambda)$ is smaller than $\delta(\lambda)^{\Omega(1)}$.

The GGM tree-based construction of PRFs [30] from one-way functions is easily seen to yield puncturable PRFs where the size of the punctured key grows polynomially with the size of the set S being punctured, as recently observed by [11, 23, 45]. Thus, we have:

Theorem 2.3. ([11, 23, 30, 45]) *If one-way functions exist, then for all efficiently computable functions $n(\lambda)$ and $m(\lambda)$, there exists a puncturable pseudorandom function that maps $n(\lambda)$ bits to $m(\lambda)$ bits (i.e., $\mathcal{D} = \{0, 1\}^{n(\lambda)}$ and $\mathcal{R} = \{0, 1\}^{m(\lambda)}$).*

2.5. Public-Key Encryption

We recall the notion of plain public-key encryption (PKE).

Definition 2.9. (*Plain public-key encryption*) Let \mathcal{M} be some message space. A *public-key encryption* (PKE) scheme for \mathcal{M} is a tuple of algorithms $(\text{KeyGen}, \text{Enc}, \text{Dec})$ where:

- $\text{KeyGen}(1^\lambda)$ takes as input the security parameter and outputs a public key PK and a secret key SK .
- $\text{Enc}(\text{PK}, m)$ takes as input the public key PK and a message $m \in \mathcal{M}$ and outputs a ciphertext CT .
- $\text{Dec}(\text{SK}, \text{CT})$ takes as input the secret key SK and a ciphertext CT , and outputs some $m \in \mathcal{M}$, or \perp .

We also require the following property:

Correctness For any message $m \in \mathcal{M}$, we have that

$$\Pr \left[\text{Dec}(\text{SK}, c) = m : \begin{array}{l} (\text{SK}, \text{PK}) \leftarrow \text{KeyGen}(1^\lambda), \\ \text{CT} \leftarrow \text{Enc}(\text{PK}, m) \end{array} \right] = 1$$

We also recall the standard notion of security.

Definition 2.10. (*Secure public-key encryption*) A tuple of algorithms $PKE = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is a *secure* PKE for \mathcal{M} if it satisfies the following requirement, formalized by the experiment $\text{Expt}_{\mathcal{A}}^{\text{PKE}}(1^\lambda, b)$ between an adversary \mathcal{A} and a challenger:

1. The challenger runs $(\text{SK}, \text{PK}) \leftarrow \text{KeyGen}(1^\lambda)$, and gives PK to \mathcal{A} .
2. At some point, \mathcal{A} sends two messages m_0^*, m_1^* as the challenge messages to the challenger.
3. The challenger generates ciphertext $\text{CT}^* \leftarrow \text{Enc}(\text{PK}, m_b^*)$ and sends CT^* to \mathcal{A} .
4. \mathcal{A} outputs a guess b' for b . The experiment outputs 1 if $b' = b$, 0 otherwise.

We say the PKE scheme is secure if, for all PPT adversaries \mathcal{A} , there exists a negligible function $\mu(\lambda)$, it holds:

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

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

2.6. Succinct Identity-Based Encryption

We define identity-based encryption (IBE) [57] with a succinctness properties.

Definition 2.11. (*Succinct IBE with γ -compression*) Let \mathcal{M} be some message space and \mathcal{ID} be an identity space. A *succinct IBE scheme with γ -compression* for $\mathcal{M}, \mathcal{ID}$ is a tuple of algorithms $(\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ where:

- $\text{Setup}(1^\lambda)$ takes as input the security parameter and outputs a master secret key MSK and a master public key MPK .
- $\text{KeyGen}(\text{MSK}, \text{id})$ takes as input the master secret MSK and an identity $\text{id} \in \mathcal{ID}$. It outputs a secret key SK_{id} for id .
- $\text{Enc}(\text{MPK}, \text{id}, m)$ takes as input the public-parameter MPK , an identity $\text{id} \in \mathcal{ID}$, and a message $m \in \mathcal{M}$, and outputs a ciphertext c .
- $\text{Dec}(\text{SK}_{\text{id}}, c)$ takes as input the secret key SK_{id} for an identity $\text{id} \in \mathcal{ID}$ and a ciphertext c , and outputs some $m \in \mathcal{M}$, or \perp .

We require the following properties:

Correctness For any message $m \in \mathcal{M}$ and identity $\text{id} \in \mathcal{ID}$, we have that

$$\Pr \left[\text{Dec}(\text{SK}_{\text{id}}, c) = m : \begin{array}{l} (\text{MSK}, \text{MPK}) \leftarrow \text{Setup}(1^\lambda), \\ \text{SK}_{\text{id}} \leftarrow \text{KeyGen}(\text{MSK}, \text{id}), \\ c \leftarrow \text{Enc}(\text{MPK}, \text{id}, m) \end{array} \right] = 1$$

Succinctness: For any security parameter $\lambda \in \mathbb{N}$, identity space \mathcal{ID} , the size of the encryption circuit Enc , for messages of size ℓ , is at most $|\mathcal{ID}|^\gamma \cdot \text{poly}(\lambda, \ell)$.

In this work, we shall consider the following selective security.

Definition 2.12. (*Selectively secure IBE*) A tuple of algorithms $\text{IBE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ is a *selectively secure* IBE scheme for $\mathcal{M}, \mathcal{ID}$ if it satisfies the following requirement, formalized by the experiment $\text{Expt}_{\mathcal{A}}^{\text{IBE}}(1^\lambda, b)$ between an adversary \mathcal{A} and a challenger:

1. \mathcal{A} submits the challenge identity $\text{id}^* \in \mathcal{ID}$ and the challenge messages (m_0^*, m_1^*) to the challenger.
2. The challenger runs $(\text{MSK}, \text{MPK}) \leftarrow \text{Setup}(1^\lambda)$, generates ciphertext $\text{CT}^* \leftarrow \text{Enc}(\text{MPK}, m_b^*)$, and gives $(\text{MPK}, \text{CT}^*)$ to \mathcal{A} .
3. \mathcal{A} is allowed to query (polynomially many) identities $\text{id} \in \mathcal{ID}$ such that $\text{id} \neq \text{id}^*$. The challenger gives $\text{SK}_{\text{id}} \leftarrow \text{KeyGen}(1^\lambda, \text{MSK}, \text{id})$ to the adversary.
4. \mathcal{A} outputs a guess b' for b . The experiment outputs 1 if $b' = b$, 0 otherwise.

We say the IBE scheme is selectively secure if, for all PPT adversaries \mathcal{A} , there exists a negligible function $\mu(\lambda)$, it holds

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

We further say that IBE is δ -selectively 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. Strong Exponentially Efficient Indistinguishability Obfuscation

Lin et al. [51] propose a variant of IO that has a weak (yet non-trivial) efficiency, which they call exponentially efficient IO (XIO). All that this notion requires in terms of efficiency is that the size of an obfuscated circuit is sublinear in the size of the corresponding truth table. They also refer to a stronger notion that requires that also the time to obfuscate a given circuit is sublinear in the size of the truth table. This notion, which we call *strong* exponentially efficient IO (SXIO), serves as one of the main abstractions in our work.

Definition 3.1. (*Strong exponentially efficient indistinguishability obfuscation (SXIO)* [51]) For a constant $\gamma < 1$, a machine sxiO is a γ -compressing *strong exponentially efficient indistinguishability obfuscator* (SXIO) for a collection of circuit classes $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if it satisfies the functionality and indistinguishability in Definition 2.7 and the following efficiency requirements:

Non-trivial time efficiency there exists a constant $\gamma < 1$ and a fixed polynomial $\text{poly}(\cdot)$, depending on the collection \mathcal{C} (but not on any specific class $\mathcal{C} \in \mathcal{C}$), such that for any class $\mathcal{C} \in \mathcal{C}$, security parameter $\lambda \in \mathbb{N}$, and circuit $C \in \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ with input length n , the running time of sxiO on input $(1^\lambda, C)$ is at most $2^{n^\gamma} \cdot \text{poly}(\lambda, |C|)$.

3.1. SXIO from Single-Input SKFE

In this section, we show that we can construct SXIO from any selectively secure t -input SKFE scheme. We recall that such a t -SKFE scheme can be constructed from any selectively secure 1-SKFE scheme, as stated in Theorem 2.1.

Theorem 3.1. *For any function $t(\lambda)$, if there exists δ -selectively secure t -SKFE for \mathbf{P}/poly , then there exists $\frac{1}{t+1}$ -compressing δ -secure SXIO for \mathbf{P}/poly .*

The idea of the construction of SXIO from SKFE is explained in Introduction.

We immediately obtain the following corollary from Theorems 2.1 and 3.1.

- Corollary 3.1.**
1. *If there exists δ -selectively secure single-input SKFE for \mathbf{P}/poly , then there exists γ -compressing δ -secure SXIO for \mathbf{P}/poly where $\gamma < 1$ is an arbitrary constant.*
 2. *Let $\varepsilon < 1$ be a constant and $\tilde{\lambda} = 2^{(\log \lambda)^\varepsilon}$. If there exists $2^{-\tilde{\lambda}^{\Omega(1)}}$ -selectively secure single-input SKFE for \mathbf{P}/poly , then there exists polynomially secure SXIO with compression factor $\gamma(\lambda) = O(1/\log \log \lambda)$ for circuits of size at most $2^{O((\log \lambda)^\varepsilon)}$. (Here, $\tilde{\lambda}$ is the single-input SKFE security parameter and λ is the SXIO security parameter.)*

3.2. The Construction of SXIO

We start by describing the SXIO construction and next argue its security. In what follows, given a circuit C , we identify its input space with $[N] = \{1, \dots, N\}$ (so in particular, $N = 2^n$ if C takes n -bit strings as input). Let $\text{SKFE}_t = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ be a selectively secure t -input secret-key functional encryption scheme.

Construction We construct an SXIO scheme $\text{sxi}\mathcal{O}$ as follows.

$\text{sxi}\mathcal{O}(1^\lambda, C)$: For every $j \in [N^{1/(t+1)}]$:

- let U_j be the t -input universal circuit that given $j_1, \dots, j_{t-1} \in [N^{1/(t+1)}]$ and a t -input circuit D , returns $D(j_1, \dots, j_{t-1}, j)$.
- let C_j be the t -input circuit that given $j_1, \dots, j_t \in [N^{1/(t+1)}]$ returns $C(j_1, \dots, j_t, j)$.

1. Generate $\text{MSK} \leftarrow \text{Setup}(1^\lambda)$.
2. Generate $\text{CT}_{t,j} \leftarrow \text{Enc}(\text{MSK}, C_j, t)$ for $j \in [N^{1/(t+1)}]$.
3. Generate $\text{CT}_{i,j} \leftarrow \text{Enc}(\text{MSK}, j, i)$ for $i \in [t-1]$ and $j \in [N^{1/(t+1)}]$.
4. Generate $\text{SK}_{U_j} \leftarrow \text{KeyGen}(\text{MSK}, U_j)$ for $j \in [N^{1/(t+1)}]$
5. $\text{sxi}\mathcal{O}(C) = (\{\text{CT}_{i,j}\}_{i \in [t], j \in [N^{1/(t+1)}]}, \{\text{SK}_{U_j}\}_{j \in [N^{1/(t+1)}]})$

$\text{Eval}(\text{sxi}\mathcal{O}, x)$: To evaluate the obfuscated circuit, convert $x \in [N]$ into $(j_1, \dots, j_t, j_{t+1}) \in [N^{1/(t+1)}]^{(t+1)}$ and output $\text{Dec}(\text{SK}_{U_{j_{t+1}}}, \text{CT}_{1,j_1}, \dots, \text{CT}_{t,j_t})$.

Proof of Theorem 3.1. We first note that $\text{sxi}\mathcal{O}$ indeed satisfies the non-trivial time efficiency requirement. The obfuscated circuit consists of $t \cdot N^{1/(t+1)}$ ciphertexts, each computable in time $\text{poly}(|C|, \lambda)$, and $N^{1/(t+1)}$ secret keys of SKFE_t , each computable in

time $\text{poly}(|C|, \lambda)$. Thus, for $t \leq \lambda$, the overall running time of the obfuscator is bounded by $N^{1/(t+1)} \cdot \text{poly}(|C|, \lambda)$, as required.

Security We show that if there exists a distinguisher \mathcal{D} against SXiO , then there exists an adversary \mathcal{A} that breaks the security of the underlying scheme SKFE_t . (It even breaks a weaker security notion than the one from Definition 2.2, where function queries are also fixed in advance.)

The reduction is straightforward. If \mathcal{D} distinguishes SXiO obfuscations of circuits C, C' (of same size and functionality), then \mathcal{A} can invoke \mathcal{D} to distinguish

$$\left\{ (j_1, \dots, j_{t-1}, C_{j_t}) : j_i \in [N^{1/(t+1)}] \right\} \text{ and } \left\{ (j_1, \dots, j_{t-1}, C'_{j_t}) : j_i \in [N^{1/(t+1)}] \right\},$$

given secret keys for all functions $\{U_{j_{t+1}} : j_{t+1} \in [N^{1/(t+1)}]\}$. Since the two circuits are functionally equivalent, all these queries are valid queries, since $U_{j_{t+1}}(j_1, \dots, j_{t-1}, C_{j_t}) = C(j_1, \dots, j_t, j_{t+1}) = C(j_1, \dots, j_t, j_{t+1}) = U_{j_{t+1}}(j_1, \dots, j_{t-1}, C'_{j_t})$.

This completes the proof of Theorem 3.1. □

Remark 3.1. (SXIO from succinct single-key SKFE) To get t -input SKFE as required above from 1-input SKFE, via the [16] transformation, the original SKFE indeed has to support an unbounded polynomial number of functional keys. We note that a similar SXIO construction is possible from a 1-input SKFE that supports a functional key for a single function f , but is *succinct* in the sense that encryption only grows mildly with the complexity of f , namely with $|f|^\beta$ for some constant $\beta < 1$.

In more detail, assume a (1-input) single-key SKFE with succinctness as above, where the time to derive a key for a function f is bounded by $|f|^c \cdot \text{poly}(\lambda)$ for some constant $c \geq 1$. The SXIO will consist of a single key for the function f that given as input C_j , as defined above, returns $C_j(1), \dots, C_j(N^{\frac{1}{c+1-\beta}})$, and encryptions of $C_1, \dots, C_{N^{c-\beta/c+1-\beta}}$. Accordingly, we still get SXIO with compression factor $\gamma = 1 - \frac{1-\beta}{c+1-\beta}$. This does not lead to arbitrary constant compression (in contrast with the theorem above), since $\frac{1}{2} \leq \gamma < 1$. Yet, it already suffices to obtain IO, when combined with LWE (as in Corollary 1.1).

4. Yao's Garbled Circuits are Decomposable

In this section, we define the notion of decomposable garbled circuits and prove that the classical Yao's garbled circuit construction satisfies our definition of decomposability (in some parameter regime). We use a decomposable garbling scheme as a building block to construct a PKFE scheme in Sect. 5.2.

4.1. Decomposable Garbling

Circuit garbling schemes [14, 61] typically consist of algorithms $(\text{Gar.CirEn}, \text{Gar.InpEn}, \text{Gar.De})$. $\text{Gar.CirEn}(C, K)$ is a circuit garbling algorithm that given a circuit C and secret key K produces a garbled circuit \widehat{C} . $\text{Gar.InpEn}(x, K)$ is an input garbling algorithm that takes an input x and the same secret key K , and produces a garbled input \widehat{x} . $\text{Gar.De}(\widehat{C}, \widehat{x})$ is a decoder that given the garbled circuit and input decodes the result y .

In this work, we shall particularly be interested in garbling *decomposable circuits*. A decomposable circuit C can be represented by a smaller circuit C_{de} that can generate each of the gates in the circuit C (along with pointers to their neighbors). When garbling such circuits, we shall require that the garbling process will also be decomposable and will admit certain *decomposable security* properties. We next formally define the notion of decomposable circuits and decomposable garbling schemes.

Definition 4.1. (*Decomposable Circuit*) Let $C : \{0, 1\}^n \rightarrow \{0, 1\}$ be a boolean circuit with L binary gates and W wires. Each gate $g \in [L]$ has an associated tuple (f, w_a, w_b, w_c) where $f : \{0, 1\}^2 \rightarrow \{0, 1\}$ is the binary function computed by the gate, $w_a, w_b \in [W]$ are the incoming wires, and $w_c \in [W]$ is the outgoing wire. A wire w_c can be the outgoing wire of at most a single gate, but can be used as an incoming wire to several different gates and therefore this models a circuit with fan-in 2 and unbounded fan-out. We define the *predecessor gates* of g to be the gates whose outgoing wires are w_a, w_b (at most 2 of them). We define the *successor gates* of g to be the gates that have w_c as an incoming wire. The gates are topologically ordered and labeled by $1, \dots, L$ so that if j is a successor of i , then $i < j$. A wire w is an *input wire* if it is not the outgoing wire of any gate. We assume that the wires $1, \dots, n$ are the input wires. There is a unique *output wire* w which is not an incoming wire to any gate.

We say that C is decomposable if there exists a smaller circuit C_{de} , called *the decomposition circuit*, that given a gate label $g \in [L]$ as input, outputs the associated tuple $C_{\text{de}}(g) = (f, w_a, w_b, w_c)$.

Definition 4.2. (*Decomposable Garbling*) A decomposable garbling scheme consists of a tuple of three deterministic polynomial-time algorithms (Gar.CirEn , Gar.InpEn , Gar.De) that work as follows:

- $\widehat{b}_i \leftarrow \text{Gar.InpEn}(i, b; K)$: takes as an input label $i \in [n]$, a bit $b \in \{0, 1\}$, and secret key $K \in \{0, 1\}^\lambda$, and outputs a garbled input bit \widehat{b}_i .
- $\widehat{G}_g \leftarrow \text{Gar.CirEn}(C_{\text{de}}, g; K)$: takes as input a decomposition circuit $C_{\text{de}} : \{0, 1\}^L \rightarrow \{0, 1\}^*$, a gate label $g \in [L]$, and secret key $K \in \{0, 1\}^\lambda$, and outputs a garbled gate \widehat{G}_g .
- $y \leftarrow \text{Gar.De}(\widehat{C}, \widehat{b})$: takes as input garbled gates $\widehat{C} = \{\widehat{G}_g\}_{g \in [L]}$, and garbled input bits $\widehat{b} = \{\widehat{b}_i\}_{i \in [n]}$, and outputs $y \in \{0, 1\}^m$.

The scheme should satisfy the following requirements:

1. **Correctness:** for every decomposable circuit C with decomposition circuit C_{de} and any input $b_1, \dots, b_n \in \{0, 1\}^n$, the decoding procedure Gar.De produces the correct output $y = C(b_1, \dots, b_n)$.
2. **(σ, τ, δ) -Decomposable Indistinguishability:** There are functions $\sigma(\Phi, s, \lambda)$, $\tau(\Phi) \in \mathbb{N}$, $\delta(\lambda) \leq 1$ such that for any security parameter λ , any input $x \in \{0, 1\}^n$, and any two circuits (C, C') that:
 - have the same topology Φ , and in particular the same size L and input-output lengths (n, m) ,
 - have decomposition circuits $(C_{\text{de}}, C'_{\text{de}})$ of the same size s
 - and agree on x : $C(x) = C'(x)$,

there exist hybrid circuits $\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)} \mid t \in [\tau] \right\}$, each being of size at most σ , as well as (possibly inefficient) hybrid functions $\left\{ \text{Gar.HPunc}^{(t)} \mid t \in [\tau] \right\}$ with the following syntax:

- $(K_{\text{pun}}^{(t)}, g_{\text{pun}}^{(t)}, i_{\text{pun}}^{(t)}) \leftarrow \text{Gar.HPunc}^{(t)}(K)$, given a key $K \in \{0, 1\}^\lambda$ and an index $t \in [\tau]$, outputs a *punctured key* $K_{\text{pun}}^{(t)}$, a gate label $g_{\text{pun}}^{(t)} \in [L]$, and an input label $i_{\text{pun}}^{(t)} \in [n]$.
- $\widehat{G}_g \leftarrow \text{Gar.HCirEn}^{(t)}(g; K)$, given a gate label $g \in [L]$, and a (possibly punctured) key K , outputs a *fake garbled gate* \widehat{G}_g .
- $\widehat{b}_i \leftarrow \text{Gar.HInpEn}^{(t)}(i, b; K)$, given an input label $i \in [n]$, and a (possibly punctured) key K , outputs a *fake garbled input bit* \widehat{b}_i .

We require that the following properties hold:

- (a) **The hybrids transition from C to C' :** For any $K \in \{0, 1\}^\lambda$, $g \in [L]$, $i \in [n]$, $b \in \{0, 1\}$,

$$\begin{aligned} \text{Gar.CirEn}(C_{\text{de}}, g; K) &= \text{Gar.HCirEn}^{(1)}(g; K), \\ \text{Gar.InpEn}(i, b; K) &= \text{Gar.HInpEn}^{(1)}(i, b; K), \\ \text{Gar.CirEn}(C'_{\text{de}}, g; K) &= \text{Gar.HCirEn}^{(\tau)}(g; K), \\ \text{Gar.InpEn}(i, b; K) &= \text{Gar.HInpEn}^{(\tau)}(i, b; K). \end{aligned}$$

- (b) **Punctured keys preserve functionality:** For any $K \in \{0, 1\}^\lambda$, and $t \in [\tau - 1]$, and letting $(K_{\text{pun}}^{(t)}, g_{\text{pun}}^{(t)}, i_{\text{pun}}^{(t)}) = \text{Gar.HPunc}^{(t)}(K)$, it holds that:

- For any $g \neq g_{\text{pun}}^{(t)}$, we have

$$\begin{aligned} \text{Gar.HCirEn}^{(t)}(g; K) &= \text{Gar.HCirEn}^{(t)}(g; K_{\text{pun}}^{(t)}) \\ &= \text{Gar.HCirEn}^{(t+1)}(g, K). \end{aligned}$$

- For any $i \neq i_{\text{pun}}^{(t)}$ and $b \in \{0, 1\}$, we have

$$\begin{aligned} \text{Gar.HInpEn}^{(t)}(i, b; K) &= \text{Gar.HInpEn}^{(t)}(i, b; K_{\text{pun}}^{(t)}) \\ &= \text{Gar.HInpEn}^{(t+1)}(i, b, K). \end{aligned}$$

- (c) **Indistinguishability on punctured inputs:** For any polysize distinguisher \mathcal{D} , security parameter $\lambda \in \mathbb{N}$, and circuits (C, C') as above,

$$\left| \Pr \left[\mathcal{D} \left(\widehat{g}_{\text{pun}}^{(t)}, \widehat{i}_{\text{pun}}^{(t)}, \text{Gar.HPunc}^{(t)}(K) \right) = 1 \right] - \Pr \left[\mathcal{D} \left(\widehat{g}_{\text{pun}}^{(t+1)}, \widehat{i}_{\text{pun}}^{(t+1)}, \text{Gar.HPunc}^{(t)}(K) \right) = 1 \right] \right| \leq \delta(\lambda),$$

where, for $t \geq 0$ we denote by $\hat{g}_{\text{pun}}^{(t)}$ the value $\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K)$ and by $\hat{i}_{\text{pun}}^{(t)}$ the value $\text{Gar.HInpEn}^{(t)}(i_{\text{pun}}^{(t)}, x_{i_{\text{pun}}^{(t)}}; K)$, with x being the input on which the two circuits C and C' agree on. The probability is over $K \leftarrow \{0, 1\}^\lambda$, and $(K_{\text{pun}}^{(t)}, g_{\text{pun}}^{(t)}, i_{\text{pun}}^{(t)}) = \text{Gar.HPunc}^{(t)}(K)$.

We show that Yao’s garbled circuit scheme, in fact, gives rise to a decomposable garbling scheme where the security loss and size of the hybrid circuits scale with the depth of the garbled circuits.

Theorem 4.1. *Let $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$ be a class of boolean circuits where each $C \in \mathcal{C}_\lambda$ has circuit size at most $L(\lambda)$, input size at most $n(\lambda)$, depth at most $d(\lambda)$, fan-out at most $\varphi(\lambda)$, and decomposition circuit of size at most $\Delta(\lambda)$. Then, assuming the existence of δ -secure one-way functions, \mathcal{C} has a decomposable garbling scheme with (σ, τ, δ) -decomposable indistinguishability where the bound on the size of hybrid circuits is $\sigma = \text{poly}(\lambda, d, \log L, \varphi, \Delta)$, the number of hybrids is $\tau = L \cdot 2^{O(d)}$, and the indistinguishability gap is $\delta^{\Omega(1)}$.*

The proof of the above theorem spans the rest of this section. We rely heavily on the ideas of Hemenway et al. [39] which considered an orthogonal question of adaptively secure garbling schemes but (for entirely different reasons) developed ideas that are useful for decomposable garbling.

4.2. Yao’s Garbled Circuits: Construction

Let $(\text{SEnc}, \text{SDec})$ be a CPA-secure symmetric-key encryption scheme with key space $\{0, 1\}^\lambda$. Furthermore, assume it satisfies *special correctness* so that for all messages m we have:

$$\Pr[\text{SDec}_k(\text{SEnc}_{k'}(m)) \neq \perp : k, k' \leftarrow \{0, 1\}^\lambda] = \text{negl}(\lambda).$$

Such encryption schemes can be based on one-way functions.

Let $\mathcal{PPRF} = (\text{PRF.Gen}, \text{PRF.Ev}, \text{PRF.Punc})$ be a PPRF with:

- domain $\mathcal{D} = ([W] \times \{0, 1\}) \cup [L]$, where $[W]$ is the set of wires and $[L]$ is the set of gates
- range $\mathcal{R} = \{0, 1\}^\lambda$

and furthermore assume that the keys output by PRF.Gen are just random values $K \in \{0, 1\}^\lambda$. Such PPRFs can be constructed based on one-way functions.

Yao’s Garbled Circuit Construction The key K of the garbling scheme is just a key for the PPRF. We define the two functions Gar.InpEn , Gar.CirEn as follows.

- $\hat{b}_i \leftarrow \text{Gar.InpEn}(i, b; K)$: Output $\hat{b}_i = \text{PRF.Ev}_K((i, b))$.
- $\hat{G}_g \leftarrow \text{Gar.CirEn}(C_{\text{de}}, g; K)$: Let $C_{\text{de}}(g) = (f, w_a, w_b, w_c)$. Compute the 6 wire labels $k_\alpha^\beta = \text{PRF.Ev}_K(w_\alpha, \beta)$ for $\alpha \in \{a, b, c\}$ and $\beta \in \{0, 1\}$, unless w_c is an output wire in which case set $k_c^0 = 0, k_c^1 = 1$. Compute 4 ciphertexts:

$$\begin{aligned}
c_{0,0} &\leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^0}(k_c^{f(0,0)})) \\
c_{0,1} &\leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^1}(k_c^{f(0,1)})) \\
c_{1,0} &\leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^0}(k_c^{f(1,0)})) \\
c_{1,1} &\leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^0}(k_c^{f(1,1)}))
\end{aligned}$$

where the encryption randomness is derived using $\text{PRF.Ev}_K(g)$. Set $\widehat{G}_g = [c_1, c_2, c_3, c_4]$ to be a lexicographic ordering of the above ciphertexts.

- $y \leftarrow \text{Gar.De}(\widehat{C}, \widehat{b})$: On input garbled gates $\widehat{C} = \{\widehat{G}_g\}_{g \in [L]}$ and garbled input bits $\widehat{b} = \{\widehat{b}_i\}_{i \in [n]}$, iteratively compute a wire label for every wire in the circuit.

For the input wires, the labels these are given by \widehat{b} . For any gate g with incoming wires w_a, w_b and outgoing wire w_c such that the labels k_a, k_b of w_a, w_b have already been computed, derive the label k_c of w_c as follows. Parse $\widehat{G}_g = [c_1, c_2, c_3, c_4]$ and set $k_c = \text{SDec}_{k_b}(\text{SDec}_{k_a}(c_i))$ for the first one of $i = 1, 2, 3, 4$ for which the outcome is not \perp .

Finally, output the value y which is the wire label of the output wire.

The correctness of this construction follows by inspection and from the special correctness of the symmetric-key encryption scheme.

4.3. Hybrid Circuits and Pebbling

We now show that Yao's construction is decomposable by defining the hybrid functions as in Definition 4.2.

Half-Sequence We now define the sequence of hybrid functions

$$\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)}, \text{Gar.HPunc}^{(t)} \mid t \in [\tau] \right\}.$$

We will actually only define a *half-sequence* which ensures that:

- $\text{Gar.CirEn}(C_{\text{de}}, g; K) = \text{Gar.HCirEn}^{(1)}(g; K), \text{Gar.InpEn}(i, b; K) = \text{Gar.HInpEn}^{(1)}(i, b; K) = \text{Gar.HInpEn}^{(\tau)}(i, b; K),$
- $\text{Gar.HCirEn}^{(\tau)}(g; K)$ does not depend on C_{de} but only on its topology Φ and the output bit $C(x) = C'(x)$.

By adding a symmetric sequence of additional hybrid functions

$$\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)}, \text{Gar.HPunc}^{(t)} \mid t \in \{\tau + 1, \dots, 2\tau - 1\} \right\}$$

where we define the functions with $t = \tau + i$ the same way as those with $t = \tau - i$ but with C'_{de} instead of C_{de} we then get the full hybrid sequence satisfying the definition.

Pebbling We define the following pebbling game over a circuit with some topology Φ . Each gate of the circuit can either have no pebble, a black pebble, or a gray pebble. The rules of the game are as follows:

- I. We can place or remove a black pebble on a gate as long as both predecessors of that gate have black pebbles on them (or the gate is an input gate).

II. We can replace a black pebble with a gray pebble on any non-output gate as long as all successors of that gate have black or gray pebbles on them.

A pebbling of the topology Φ is a sequence of steps where we start with no pebbles on any gate and sequentially apply rules I and II so as to end up with a gray pebble on every non-output gate and a black pebble on the output. Let us assume that the topology Φ admits a pebbling consisting of steps $\text{pstep}_1, \dots, \text{pstep}_{\tau'}$.

Pebbling and Circuit Configurations We will define the hybrids, each of which being parametrized by a circuit configuration conf . For every gate g , the configuration conf specifies if the gate is in one of four possible modes

$$\{\text{RealGate}, \text{SimGate}, \text{CompDepSimGate}, \frac{1}{2}\text{CompDepSimGate}\}.$$

Furthermore, for every wire w which is an incoming or outgoing wire of some gate in mode CompDepSimGate or $\frac{1}{2}\text{CompDepSimGate}$, the configuration specifies a bit $v(w)$.

Assume we are given an input x and a circuit C having topology Φ that admits a pebbling consisting of steps $\text{pstep}_1, \dots, \text{pstep}_{\tau'}$. We first define a sequence of configurations $\text{conf}'_1, \dots, \text{conf}'_{\tau'}$ where every configuration conf'_i corresponds to the pebble placement after step i of the pebbling: every gate that has no pebble is in RealGate mode, every gate that has a black pebble is in CompDepSimGate mode, every gate that has a gray pebble is in SimGate mode. Furthermore, for every wire w which is an incoming or outgoing wire of some gate in mode CompDepSimGate we set $v(w)$ to be the bit going over the wire w during the computation $C(x)$. Next we add several intermediate configurations as follows: if the transition from conf'_i , conf'_{i+1} corresponds to an application of rule I in the pebbling, meaning that we either add or remove a black pebble on some gate g , then we add an intermediate configuration in between these, which is identical to conf'_i , conf'_{i+1} except that the gate g is in $\frac{1}{2}\text{CompDepSimGate}$ mode. This results in a sequence $\text{conf}_1, \dots, \text{conf}_{\tau}$ where $\tau \leq 2\tau'$.

Configurations and Hybrids We now define the hybrid functions

$$\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)}, \text{Gar.HPunc}^{(t)} \mid t \in [\tau] \right\}.$$

First, for a configuration conf we define the functions $\text{Gar.HInpEn}^{(\text{conf})}$ and $\text{Gar.HCirEn}^{(\text{conf})}$ parameterized by conf . For every $t \in [\tau]$, we then define $\text{Gar.HInpEn}^{(t)} = \text{Gar.HInpEn}^{(\text{conf}'_t)}$ and $\text{Gar.HCirEn}^{(t)} = \text{Gar.HCirEn}^{(\text{conf}'_t)}$ where $\text{conf}_1, \dots, \text{conf}_{\tau}$ is the sequence of configuration given by the pebbling game as above.

- $\widehat{G}_g \leftarrow \text{Gar.HCirEn}^{(\text{conf})}(g; K)$: Let $C_{\text{de}}(g) = (f, w_a, w_b, w_c)$.

If the key K is a standard PPRF key, then proceed as follows. Compute the 6 wire labels $k_{\alpha}^{\beta} = \text{PRF.EV}_K(w_{\alpha}, \beta)$ for $\alpha \in \{a, b, c\}$ and $\beta \in \{0, 1\}$, unless w_c is an output wire in which case set $k_c^0 = 0, k_c^1 = 1$. Compute 4 ciphertexts $c_{0,0}, c_{0,1}, c_{1,0}, c_{1,1}$ as in Fig. 2 depending on the mode of g , where the encryption randomness is derived from $\text{PRF.EV}_K(g)$. Let $\widehat{G}_g = [c_1, c_2, c_3, c_4]$ to be a lexicographic ordering of the above ciphertexts.

RealGate	SimGate	CompDepSimGate
$c_{0,0} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^0}(k_c^{f(0,0)}))$	$c_{0,0} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^0}(k_c^0))$	$c_{0,0} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^0}(k_c^{v(w_c)}))$
$c_{0,1} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^1}(k_c^{f(0,1)}))$	$c_{0,1} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^1}(k_c^0))$	$c_{0,1} \leftarrow \text{SEnc}_{k_a^0}(\text{SEnc}_{k_b^1}(k_c^{v(w_c)}))$
$c_{1,0} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^0}(k_c^{f(1,0)}))$	$c_{1,0} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^0}(k_c^0))$	$c_{1,0} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^0}(k_c^{v(w_c)}))$
$c_{1,1} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^1}(k_c^{f(1,1)}))$	$c_{1,1} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^1}(k_c^0))$	$c_{1,1} \leftarrow \text{SEnc}_{k_a^1}(\text{SEnc}_{k_b^1}(k_c^{v(w_c)}))$

$\frac{1}{2}$ CompDepSimGate
$c_{1-v(w_a),0} \leftarrow \text{SEnc}_{k_a^{1-v(w_a)}}(\text{SEnc}_{k_b^0}(k_c^{v(w_c)}))$
$c_{1-v(w_a),1} \leftarrow \text{SEnc}_{k_a^{1-v(w_a)}}(\text{SEnc}_{k_b^1}(k_c^{v(w_c)}))$
$c_{v(w_a),0} \leftarrow \text{SEnc}_{k_a^{v(w_a)}}(\text{SEnc}_{k_b^0}(k_c^{f(v(w_a),0)}))$
$c_{v(w_a),1} \leftarrow \text{SEnc}_{k_a^{v(w_a)}}(\text{SEnc}_{k_b^1}(k_c^{f(v(w_a),1)}))$

Fig. 2. Garbling Gate modes: RealGate, SimGate, CompDepSimGate, $\frac{1}{2}$ CompDepSimGate.

If the key K is not a standard PPRF key, then parse $K = (K\{S\}, H)$. If there is a tuple of the form (g, \widehat{G}_g) in H then output \widehat{G}_g . Else attempt to compute \widehat{G}_g as above using the punctured PPRF key $K\{S\}$ and output \perp if this is not possible.

- **Gar.HInpEn^(conf)** $(i, b; K)$: If the key K is a standard PPRF key, then output $\widehat{b}_i = \text{PRF.Ev}_K((i, b))$. Else parse $K = (K\{S\}, H)$ and output $\widehat{b}_i = \text{PRF.Ev}_{K\{S\}}((i, b))$ or \perp if this computation fails.

We define **Gar.HPunc^(t)** (K) as follows. Assume that the transition from conf_t to conf_{t+1} changes the mode of a gate $g_{\text{pun}}^{(t)}$ such that $g_{\text{pun}}^{(t)}$ has incoming wires w_a, w_b and outgoing wire w_c .

- If the transition changes $g_{\text{pun}}^{(t)}$ from RealGate to $\frac{1}{2}$ CompDepSimGate (or vice versa):
Gar.HPunc^t $(K) := (K_{\text{pun}}^t = ((K\{S\}, H), g_{\text{pun}}^{(t)}, i_a)$ where $S = \{(w_a, 1 - v(w_a)), g_{\text{pun}}^{(t)}\}$, $K\{S\} = \text{PRF.Punc}(K, S)$, $i_a = \perp$ if w_a is not an input wire, else i_a is the index of input wire w_a . For every gate $g \neq g_{\text{pun}}^{(t)}$ that has w_a as an input wire compute $\widehat{G}_g \leftarrow \text{Gar.HCirEn}^{(\text{conf}_t)}(g; K)$ and add \widehat{G}_g to H .
- If the transition changes $g_{\text{pun}}^{(t)}$ from $\frac{1}{2}$ CompDepSimGate to CompDepSimGate (or vice versa):
Gar.HPunc^t $(K) := (K_{\text{pun}}^t = ((K\{S\}, H), g_{\text{pun}}^{(t)}, i_b)$ where $S = \{(w_b, 1 - v(w_b)), g_{\text{pun}}^{(t)}\}$, $K\{S\} = \text{PRF.Punc}(K, S)$, $i_b = \perp$ if w_b is not an input wire, else i_b is the index of input wire w_b . For every gate $g \neq g_{\text{pun}}^{(t)}$ that has w_b as an input wire compute $\widehat{G}_g \leftarrow \text{Gar.HCirEn}^{(\text{conf}_t)}(g; K)$ and add \widehat{G}_g to H .
- If the transition changes $g_{\text{pun}}^{(t)}$ from CompDepSimGate to SimGate (or vice versa):
Gar.HPunc^t $(K) := (K_{\text{pun}}^t = ((K\{S\}, H), g_{\text{pun}}^{(t)}, \perp)$ with S being the set $\{(w_c, 0), (w_c, 1), g_{\text{pun}}^{(t)}\}$, $K\{S\} = \text{PRF.Punc}(K, S)$. For every gate $g \neq g_{\text{pun}}^{(t)}$ that has w_c as an input wire compute $\widehat{G}_g \leftarrow \text{Gar.HCirEn}^{(\text{conf}_t)}(g; K)$ and add \widehat{G}_g to H .

4.4. Proof of Security

We now show that the above hybrid functions satisfy properties (a), (b), (c) of Definition 4.2.

Property (a) Firstly, $\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)}, \text{Gar.HPunc}^{(t)} \mid t \in [\tau] \right\}$ represents a valid half-sequence, meaning that:

- $\text{Gar.CirEn}(C_{\text{de}}, g; K) = \text{Gar.HCirEn}^{(1)}(g; K)$, $\text{Gar.InpEn}(i, b; K) = \text{Gar.HInpEn}^{(1)}(i, b; K) = \text{Gar.HInpEn}^{(\tau)}(i, b; K)$,
- $\text{Gar.HCirEn}^{(\tau)}(g; K)$ does not depend on C_{de} but only on its topology Φ and the output bit $C_{\text{de}}(x)$.

In particular, $\text{Gar.HCirEn}^{(\tau)}(g; K) = \text{Gar.HCirEn}^{(\text{conf}_\tau)}(g; K)$ corresponds to a configuration where all non-output gates are in `SimGate` mode and the output gate is in `CompDepSimGate` mode, meaning that $\text{Gar.HCirEn}^{(\text{conf}_\tau)}(g; K)$ does not depend on the boolean function f implemented by any of the gates. Furthermore, the configuration conf_τ only specifies the output value $v(w_{\text{out}}) = C(x)$. Therefore, these functions are a valid half-sequence. As we discussed, by adding a symmetric half-sequence in the reverse direction for the circuit C'_{de} we get a sequence that satisfies property (a).

Property (b) This property follows by inspection. In particular,

- For any $g \neq g_{\text{pun}}^{(t)}$, we have

$$\text{Gar.HCirEn}^{(t)}(g; K) = \text{Gar.HCirEn}^{(t)}(g; K_{\text{pun}}^{(t)}) = \text{Gar.HCirEn}^{(t+1)}(g, K) .$$

Since $\text{Gar.HCirEn}^{(t)}(g; K)$ and $\text{Gar.HCirEn}^{(t+1)}(g, K)$ only differ on the mode of the gate $g_{\text{pun}}^{(t)}$ and the punctured key $K_{\text{pun}}^{(t)}$ hardwires the value of all other gates affected by PRF puncturing to match those of $\text{Gar.HCirEn}^{(t)}(g; K)$

- For any $i \neq i_{\text{pun}}^{(t)}$ and $b \in \{0, 1\}$, we have $\text{Gar.HInpEn}^{(t)}(i, b; K) = \text{Gar.HInpEn}^{(t)}(i, b; K_{\text{pun}}^{(t)}) = \text{Gar.HInpEn}^{(t+1)}(i, b, K)$. Actually, we have $\text{Gar.HInpEn}^{(t)}(i, b; K) = \text{Gar.InpEn}(i, b; K)$ as well as $\text{Gar.HInpEn}^{(t)}(i, b; K_{\text{pun}}^{(t)}) = \text{Gar.InpEn}(i, b; K)$, for all $i \neq i_{\text{pun}}^{(t)}$.

Property (c) For property (c), we must show that

$$\begin{aligned} & \left(\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K), \text{Gar.HInpEn}^{(t)}(i_{\text{pun}}^{(t)}, x_{i_{\text{pun}}^{(t)}}; K), K_{\text{pun}}^{(t)} \right) \\ & \approx \left(\text{Gar.HCirEn}^{(t+1)}(g_{\text{pun}}^{(t)}; K), \text{Gar.HInpEn}^{(t+1)}(i_{\text{pun}}^{(t)}, x_{i_{\text{pun}}^{(t)}}; K), K_{\text{pun}}^{(t)} \right) \end{aligned}$$

are computationally indistinguishable, where the probability is over $K \leftarrow \{0, 1\}^\lambda$, and $(K_{\text{pun}}^{(t)}, g_{\text{pun}}^{(t)}, i_{\text{pun}}^{(t)}) = \text{Gar.HPunc}^{(t)}(K)$.

Since in our case $\text{Gar.HInpEn}^{(t)}(i, b; K) = \text{Gar.HInpEn}^{(t+1)}(i, b; K) = \text{Gar.InpEn}(i, b; K)$ for all i, b , we can simplify the above to:

$$\begin{aligned} & \left(\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K), \text{Gar.InpEn}(i_{\text{pun}}^{(t)}, x_{i_{\text{pun}}^{(t)}}; K), K_{\text{pun}}^{(t)} \right) \\ & \approx \left(\text{Gar.HCirEn}^{(t+1)}(g_{\text{pun}}^{(t)}; K), \text{Gar.InpEn}(i_{\text{pun}}^{(t)}, x_{i_{\text{pun}}^{(t)}}; K), K_{\text{pun}}^{(t)} \right) \end{aligned}$$

We consider several cases. Assume that the transition from conf_t to conf_{t+1} changes the mode of a gate $g_{\text{pun}}^{(t)}$ such that $g_{\text{pun}}^{(t)}$ has incoming wires w_a, w_b and outgoing wire w_c .

Case 1: If the transition changes $g_{\text{pun}}^{(t)}$ from RealGate to $\frac{1}{2}\text{CompDepSimGate}$ (or vice versa):

The difference between $\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K)$, $\text{Gar.HCirEn}^{(t+1)}(g_{\text{pun}}^{(t)}; K)$ is only in the value encrypted by the ciphertexts:

$$\begin{aligned} c_{1-v(w_a),0} &\leftarrow \text{SEnc}_{k_a^{1-v(w_a)}}(\dots) \\ c_{1-v(w_a),1} &\leftarrow \text{SEnc}_{k_a^{1-v(w_a)}}(\dots) \end{aligned}$$

The key $k_a^{1-v(w_a)} = \text{PRF.Ev}_K(w_a, 1 - v(w_a))$ is computed using a PPRF and the encryption randomness is derived using $r = \text{PRF.Ev}_K(g_{\text{pun}}^{(t)})$. Furthermore, $K_{\text{pun}}^{(t)} = (K\{S\}, H)$ where $K\{S\}$ is punctured at $S = \{(w_a, 1 - v(w_a)), g_{\text{pun}}^{(t)}\}$. The set H only contains other ciphertexts created under the key $k_a^{1-v(w_a)}$ but otherwise does not contain any other information related to $k_a^{1-v(w_a)}$ or r . Therefore, we can first rely on punctured PRF security to switch $k_a^{1-v(w_a)}$ and r to random, then on CPA security to switch the value encrypted by the ciphertexts $c_{1-v(w_a),0}, c_{1-v(w_a),1}$ from that of $\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K)$ to that of $\text{Gar.HCirEn}^{(t+1)}(g_{\text{pun}}^{(t)}; K)$ and then on punctured PRF security again to switch $k_a^{1-v(w_a)}$ and the encryption randomness back to PRF outputs. This proves indistinguishability.

Case 2: If the transition changes $g_{\text{pun}}^{(t)}$ from $\frac{1}{2}\text{CompDepSimGate}$ to the mode CompDepSimGate (or vice versa):

This case is identical to the previous one with w_b instead of w_a .

Case 3: If the transition changes $g_{\text{pun}}^{(t)}$ from CompDepSimGate to SimGate (or vice versa):

The difference between $\text{Gar.HCirEn}^{(t)}(g_{\text{pun}}^{(t)}; K)$, $\text{Gar.HCirEn}^{(t+1)}(g_{\text{pun}}^{(t)}; K)$ is only a switch from $k_c^{v(w_c)}$ to k_c^0 in the garbled gate $g_{\text{pun}}^{(t)}$.

The keys $k_c^0 = \text{PRF.Ev}_K(w_c, 0)$, $k_c^1 = \text{PRF.Ev}_K(w_c, 1)$ are computed using the PPRF. Furthermore, $K_{\text{pun}}^{(t)} = (K\{S\}, H)$ where $K\{S\}$ is punctured at $S = \{(w_c, 0), (w_c, 1)\}$. The keys k_c^0, k_c^1 are used to compute some values in H , but they are used in a completely symmetric manner (this is because all successors of $g_{\text{pun}}^{(t)}$ are in SimGate , CompDepSimGate modes and therefore the 0 and 1 keys are used identically). Therefore, we first rely on PPRF security to switch k_c^0, k_c^1 to random values, then by symmetry we can exchange $k_c^{v(w_c)}$ for k_c^0 in the garbled gate $g_{\text{pun}}^{(t)}$, and then we rely on PPRF security again to switch them back to PRF outputs.

This completes the proof of property (c).

4.5. Pebble Complexity and Parameters

We now analyze the parameters $\sigma(\Phi, s, \lambda)$, $\tau(\Phi) \in \mathbb{N}$, $\delta(\lambda)$ as defined in Definition 4.2 that our construction achieves. Assume that the topology Φ has maximum fan-out $\varphi(\Phi)$ and can be pebbled in $\tau'(\Phi)$ steps using at most $p(\Phi)$ black pebbles. Furthermore, assume that the pebbling is *succinct* meaning that in any step t of the pebbling there is an index $g_{gray}^{(t)} \in [L]$ such that a gate g has a gray pebble on it if and only if g is not the output gate and $g \geq g_{gray}^{(t)}$. Lastly assume that the size of the compact circuits C_{de} , C'_{de} is at most Δ and that the symmetric-key encryption scheme and the PPRF have security $\delta_{ENC}(\lambda)$ and $\delta_{PPRF}(\lambda)$, respectively, meaning that the advantage of any polynomial-time attacker against these schemes is bounded by $\delta_{ENC}(\lambda)$ and $\delta_{PPRF}(\lambda)$, respectively.

Lemma 4.1. *The construction given achieves $\tau(\Phi) \leq 4\tau'(\Phi)$, $\sigma(\Phi, s, \lambda) \leq \text{poly}(\lambda, \varphi(\Phi), p(\Phi), \log L) + \Delta$, and $\delta(\lambda) \leq O(\delta_{ENC}(\lambda) + \delta_{PPRF}(\lambda))$.*

Lemma 4.1. It's easy to see that $\tau(\Phi) \leq 4\tau'(\Phi)$ since we added at most one intermediate configuration in between any pebbling steps to define the half-sequence of hybrid functions of length at most $2\tau'(\Phi)$, and therefore the length of the full sequence is of length at most $4\tau'(\Phi)$.

We now compute $\sigma(\Phi, s, \lambda)$ which is a bound on the circuit size of

$$\left\{ \text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)} \mid t \in [\tau] \right\}.$$

First, we note that the size of the punctured key K_{pun}^t is $|K_{\text{pun}}^t| \leq \varphi(\Phi)\text{poly}(\lambda)$ since it hardwires at most $\varphi(\Phi)$ garbled gates. Secondly we note that the size of any configuration conf_t can be described by at most $O(p(\Phi) \log L)$ bits since it only needs to specify the $p(\Phi)$ gates that are in `CompDepSimGate` mode and in $\frac{1}{2}$ `CompDepSimGate` mode, the bits $v(w_c)$ for each outgoing wire w_c of such gates, and the index $g_{gray}^{(t)}$ to be able to decide if a gate is in `SimGate` or `RealGate` mode. Lastly, the function $\text{Gar.HCirEn}^{(t)}$ needs to run the circuit C_{de} or C'_{de} once and then apply some processing on the output. Therefore, by observation, the circuit size is bounded by $\text{poly}(\lambda, \varphi(\Phi), p(\Phi), \log L) + \Delta$.

Finally, the bound on $\delta(\lambda)$ follows directly from the security proof. \square

In [39] (Section 6, Strategy 2), it is shown that any circuit topology of depth d and circuit size L can be pebbled using $p(\Phi) \leq 2d$ black pebbles and in $\tau'(\Phi) \leq L4^d$ steps. Furthermore, it is easy to see that this pebbling is succinct.⁴ Therefore, by plugging these parameters into Lemma 4.1 we get the proof of Theorem 4.1.

⁴Indeed, if the gates $1, \dots, L$ are topologically sorted so that L is the output gate, then the pebbling can be made to place gray pebbles in according to this ordering, meaning that the first gray pebble is placed on gate $L - 1$ and then $L - 2$ and finally continuing down to 1. One minor difference between our version and the one in [39] is that the latter allows replacing a black pebble with a gray pebble at the output gate, whereas our does not. This only requires us to slightly modify the pebbling strategy to keep the pebble at the output gate black.

5. Single-Key Succinct PKFE from SXIO and PKE

This section consists of four subsections. The main part is constructing a weakly succinct PKFE scheme for boolean functions in Sect. 5.2. First, in Sect. 5.1, we construct a succinct IBE scheme with γ -compression, which we shall use as a building block in Sect. 5.2. In Sect. 5.3, we present a transformation from weakly succinct PKFE schemes for boolean functions into ones for non-boolean functions. Lastly, we explain how the pieces come together to give IO from SKFE in Sect. 5.4.

5.1. A Succinct IBE Scheme

In this section, we construct a succinct IBE scheme from SXIO and PKE. Here by *succinct*, we mean that the size of the encryption circuit (and then also of the master public key) is sublinear in the size of the identity space (roughly, $|\mathcal{ID}|^\gamma$, with $\gamma < 1$, see Sect. 2.6 for the complete definition). In the following subsections, we will use the constructed IBE as a building block in our construction of succinct PKFE.

Theorem 5.1. *For any $\beta < \gamma < 1$, assuming there exists a β -compressing SXIO scheme for P/poly, a puncturable PRF, and a plain PKE scheme, there exists a succinct IBE scheme with γ -compression. Moreover, assuming the underlying primitives are δ -secure so is the resulting IBE scheme.*

We start by describing the IBE construction and then argue its security. Let $\mathcal{PPRF} = (\text{PRF.Gen}, \text{PRF.Ev}, \text{PRF.Punc})$ be a puncturable PRF, $\text{PKE} = (\text{PKE.Gen}, \text{PKE.Enc}, \text{PKE.Dec})$ a PKE scheme, and sxiO a β -compressing SXIO scheme. Let $[s]$ denote the identity space.

Construction The scheme $\text{IBE} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Dec})$ for an identity space $\mathcal{ID} = [s]$ is given by the following algorithms.

Setup(1^λ):

- Choose puncturable PRF key $S \leftarrow \text{PRF.Gen}(1^\lambda)$.
- Construct a circuit $\text{KG}_{\text{ibe}}[S]$ as described in Fig. 3 that computes $r_i = \text{PRF.Ev}_S(i)$ for an input $i \in [s]$ and generates a pair of keys $(\text{PKE.PK}_i, \text{PKE.SK}_i) \leftarrow \text{PKE.Gen}(1^\lambda; r_i)$.
- Output $\text{MPK} = \text{sxiO}(\text{KG}_{\text{ibe}}[S])$ and $\text{MSK} = \{\text{PKE.SK}_i\}_{i \in [s]}$.

KeyGen(MSK, id):

- Parse $\text{MSK} = \{\text{PKE.SK}_i\}_{i \in [s]}$.
- Output $\text{SK}_{\text{id}} = \text{PKE.SK}_{\text{id}}$.

Enc(MPK, id, x):

- Compute $\text{PKE.PK}_{\text{id}}$ by running $\text{MPK}(\text{id}) = \text{sxiO}(\text{KG}_{\text{ibe}}[S])(\text{id})$.
- Output $\text{CT} \leftarrow \text{PKE.Enc}(\text{PKE.PK}_{\text{id}}, x)$.

Dec($\text{SK}_{\text{id}}, \text{CT}$):

- Output $\text{PKE.Dec}(\text{SK}_{\text{id}}, \text{CT})$.

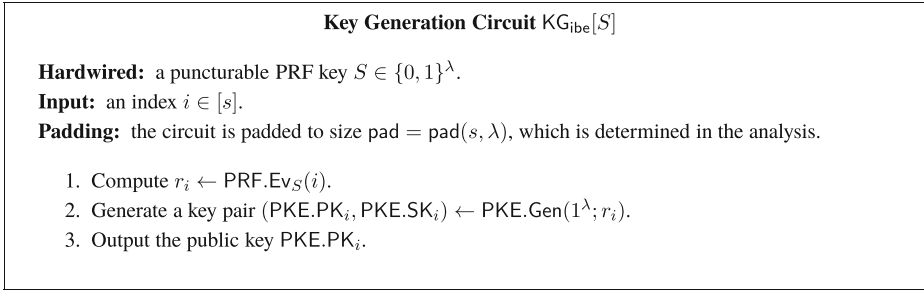


Fig. 3. Circuit $\text{KG}_{\text{ibe}}[S]$.

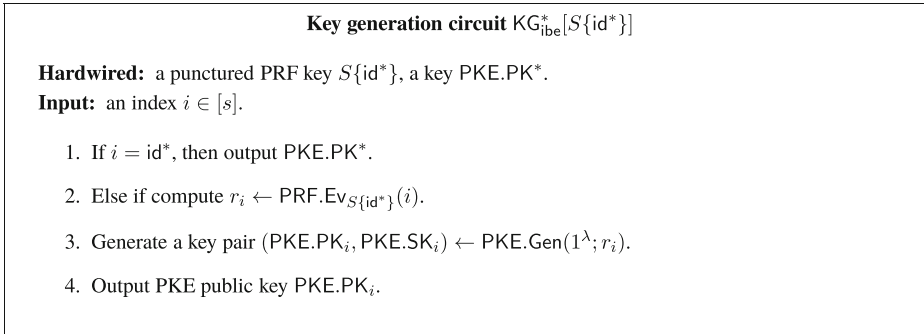


Fig. 4. Circuit $\text{KG}_{\text{ibe}}^*[S\{\text{id}^*\}]$.

Proof of Theorem 5.1. We now prove Theorem 5.1, starting by analyzing succinctness, and then moving on to the security proof.

Padding Parameter

The proof of security relies on indistinguishability of obfuscations of circuits KG_{ibe} and KG_{ibe}^* defined in Figs. 3 and 4. Accordingly, we set $\text{pad} = \max(|\text{KG}_{\text{ibe}}|, |\text{KG}_{\text{ibe}}^*|)$. The circuits KG_{ibe} and KG_{ibe}^* compute a puncturable PRF over domain $[s]$, a PKE key pair, and may have punctured PRF keys and public keys hardwired. Thus,

$$\text{pad} \leq \text{poly}(\lambda, \log(s)) .$$

Succinctness

The input space for KG_{ibe} is $[s]$. Therefore, by the SXIO guarantee, the size of the encryption circuit (dominated by running the obfuscated KG_{ibe}) is

$$s^\beta \cdot \text{poly}(\lambda, \ell, \log s) \leq s^\gamma \cdot \text{poly}(\lambda, \ell) ,$$

where ℓ is a bound on the length of encrypted messages.

Security Proof

Let us assume that the underlying primitives are δ -secure. We define a sequence of hybrid games.

Hyb₀: The first game is the original selective security experiment for $b = 0$, $\text{Expt}_{\mathcal{A}}^{\text{IBE}}(1^\lambda, 0)$. In this game, the adversary first selects the challenge identity id^* and messages (m_0^*, m_1^*) and then gets an encryption of m_0^* for identity id^* and the master public key. It can also query polynomially many secret keys for identities different from id^* (See Definition 2.12 for more details).

Hyb₁: We change KG_{ibe} into KG_{ibe}^* described in Fig. 4. In this hybrid game, we set $r^* = \text{PRF.Ev}_S(\text{id}^*)$ and $(\text{PKE.PK}^*, \text{PKE.SK}^*) \leftarrow \text{PKE.Gen}(1^\lambda; r^*)$. Thus, the behaviors of KG_{ibe} and KG_{ibe}^* are totally the same, and so are their size since we pad circuit KG_{ibe} to have the same size as KG_{ibe}^* . Then, we can use the indistinguishability guarantee of $\text{sxi}\mathcal{O}$ and it holds $\text{Hyb}_0 \approx_\delta \text{Hyb}_1$.

Hyb₂: We change $r^* = \text{PRF.Ev}_S(\text{id}^*)$ into uniformly random r^* . Due to the pseudorandomness at punctured points, it holds $\text{Hyb}_1 \approx_\delta \text{Hyb}_2$.

Hyb₃: We change CT^* from $\text{PKE.Enc}(\text{id}^*, m_0^*)$ to $\text{PKE.Enc}(\text{id}^*, m_1^*)$. In Hyb_2 and Hyb_3 , we do not need randomness used to generate PKE.PK^* . We just use the hardwired PKE.PK^* . Therefore, $\text{Hyb}_2 \approx_\delta \text{Hyb}_3$ follows directly from the semantic security of the PKE scheme.

Hyb₄: This is $\text{Expt}_{\mathcal{A}}^{\text{IBE}}(1^\lambda, 1)$. We can show the indistinguishability between Hyb_3 and Hyb_4 in a reverse manner.

This completes the proof of Theorem 5.1. □

5.2. Weakly Succinct PKFE for Boolean Functions

We now construct a single-key weakly succinct PKFE scheme for the class of boolean functions. The construction is based on succinct IBE, decomposable garbling, and SXIO.

Theorem 5.2. *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of circuits with a single output bit and let $n(\lambda)$, $s(\lambda)$, $d(\lambda)$ be bounds on their input length, size, and depth (respectively). For any constants β, γ such that $3\beta < \gamma < 1$, assuming a δ -secure, β -compressing SXIO for P/poly , there exists a constant α , such that given any δ -secure, α -compressing IBE, and δ -secure one-way functions, there exists a $2^d s \delta$ -secure succinct PKFE for \mathcal{C} with compression factor γ .*

Depth preserving universal circuits To prove the above theorem, we recall the existence of depth preserving universal circuits [24]. Concretely, any family of circuits \mathcal{C} as considered in Theorem 5.2 has a uniform family of universal circuits $\{U_\lambda\}_{\lambda \in \mathbb{N}}$ with fan-out λ ,⁵ depth $O(d)$, and size $s^3 \cdot \text{polylog}(s)$, for some fixed polynomial poly . Each such circuit takes as input a description (f_1, \dots, f_s) of a function in \mathcal{C} and an input

⁵The restriction regarding fan-out is not stated explicitly in [24], but can always be achieved by blowing up the size and depth by a factor of at most $O(1)$.

(x_1, \dots, x_n) and outputs $f(x)$. Furthermore, uniformity here means that each circuit has a decomposition circuit of size $\text{polylog}(s)$.

Ingredients and notation used in the construction

- We denote by $U^{(x)} : \{0, 1\}^s \rightarrow \{0, 1\}$ the universal circuit, with $x \in \{0, 1\}^n$ being a hardwired bit string, such that on input (f_1, \dots, f_s) , the circuit $U^{(x)}$ outputs $f(x)$. This circuit has a decomposition circuit of size $\text{poly}(n, \log(s))$, which we denote by $U_{\text{de}}^{(x)}$. We also denote by L the number of gates in the circuit $U^{(x)}$.
- Let SxiO be a δ -secure, β -compressing SXIO scheme.
- Let $\text{IBE} = (\text{IBE.Setup}, \text{IBE.KeyGen}, \text{IBE.Enc}, \text{IBE.Dec})$ be δ -secure, succinct, IBE scheme with α -compression for the identity space being $\mathcal{ID} = [s] \times \{0, 1\}$.
- Let $(\text{Gar.CirEn}, \text{Gar.InpEn}, \text{Gar.De})$ be a decomposable garbling scheme with (σ, τ, δ) -decomposable indistinguishability where $\tau = s^{2^{O(d)}}$ and $\sigma = \text{poly}(\lambda, n, d, \log(s))$. Such schemes are implied by δ -secure one-way functions (Theorem 4.1).
- Let $\mathcal{PPRF} = (\text{PRF.Gen}, \text{PRF.Ev}, \text{PRF.Punc})$ be a δ -secure puncturable PRF. These are implied by δ -secure one-way functions [11,23,30,45].

Construction The scheme consists of the following algorithms.

PKFE.Setup (1^λ) :

- Run $(\text{MSK}_{\text{ibe}}, \text{MPK}_{\text{ibe}}) \leftarrow \text{IBE.Setup}(1^\lambda)$.
- Set $\text{MSK} = \text{MSK}_{\text{ibe}}, \text{MPK} = \text{MPK}_{\text{ibe}}$.

PKFE.Key (MSK, f) :

- Compute $\text{SK}_{i, f_i} \leftarrow \text{IBE.KeyGen}(\text{MSK}_{\text{ibe}}, (i, f_i))$ for $i \in [s]$, where $f = (f_1, \dots, f_s)$.
- Return $\text{SK}_f = \{\text{SK}_{i, f_i}\}_{i \in [s]}$.

PKFE.Enc (MPK, x) :

- Compute $U_{\text{de}}^{(x)}$ and pick a garbling key $K \leftarrow \{0, 1\}^\lambda$ and a punctured key $S \leftarrow \text{PRF.Gen}(1^\lambda)$;
- Generate an obfuscation $\widetilde{\text{IGC}} = \text{SxiO}(1^\lambda, \text{IGC}[K, S, \text{MPK}])$ of the input garbling circuit defined in Fig. 5;
- Generate an obfuscation $\widetilde{\text{GGC}} = \text{SxiO}(1^\lambda, \text{GGC}[K, U_{\text{de}}^{(x)}])$ of the gate garbling circuit defined in Fig. 6;
- Return $\text{CT}_x = (\widetilde{\text{IGC}}, \widetilde{\text{GGC}})$.

PKFE.Dec $(\text{SK}_f, \text{CT}_x)$:

- For $i \in [s]$, run $\widetilde{\text{IGC}}(i, f_i)$ to obtain an IBE ciphertext, and decrypt the output using SK_{i, f_i} to obtain \hat{f}_i .
- For all $g \in [L]$, run $\widetilde{\text{GGC}}(g)$, in order to obtain the garbled gate \hat{G}_g .
- Return $y \leftarrow \text{Gar.De}(\hat{C}, \hat{f})$, with $\hat{C} = \{\hat{G}_g\}_{g \in [L]}$ and $\hat{f} = \{\hat{f}_i\}_{i \in [s]}$.

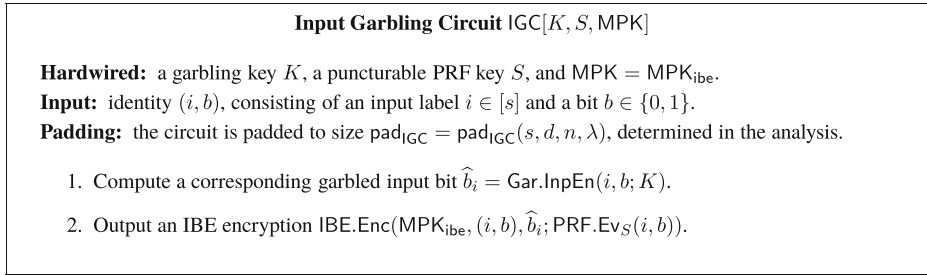


Fig. 5. Circuit $\text{IGC}[K, S, \text{MPK}]$.

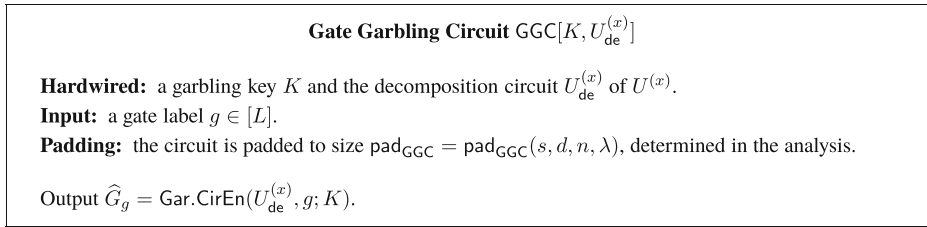


Fig. 6. Circuit $\text{GGC}[K, U_{\text{de}}^{(x)}]$.

Proof of Theorem 5.2. We now prove Theorem 5.2, starting by correctness, continuing to succinctness, and ending with the proof of security. In what follows, let s , d , and n be bounds on the size, the depth, and the input length of functions in \mathcal{C} .

Correctness

Correctness immediately follows from the correctness of the underlying identity-based encryption and decomposable garbling schemes.

Padding Parameter

The proof of security relies on the indistinguishability guarantee of SXIO with respect to two hybrid sequences of circuits, one corresponding to input garbling, and the other one corresponding to gate garbling. Thus, we pad every circuit of each sequence to the maximal size of any circuit in the sequence. That is, we consider

$$\begin{aligned} \mathcal{S}_{\text{IGC}} &= \{\text{IGC}, \text{HIGC}_t, \text{HIGC}_t^{(j)} \mid 1 \leq t \leq \tau, 1 \leq j \leq 5\} \\ \mathcal{S}_{\text{GGC}} &= \{\text{GGC}, \text{HGGC}_t, \text{HGGC}_t^{(j)} \mid 1 \leq t \leq \tau, 1 \leq j \leq 5\}, \end{aligned}$$

and let $\text{pad}_{\text{IGC}} = \max_{C \in \mathcal{S}_{\text{IGC}}} |C|$ and $\text{pad}_{\text{GGC}} = \max_{C \in \mathcal{S}_{\text{GGC}}} |C|$.

We bound pad_{IGC} and pad_{GGC} as follows:

- Any circuit in \mathcal{S}_{IGC} consists of the following:
 - A punctured PRF computation (deriving randomness for IBE encryption) over a domain of size $O(s)$, using keys that may be punctured at at most con-

stant number of points. The contribution to the circuit size is bounded by $\text{poly}(\lambda, \log(s))$.

- Input garbling (possibly via one of the hybrid input garbling circuits). The contribution to the circuit size is bounded by $\sigma = \text{poly}(\lambda, d, \log(s), n)$.
- IBE encryption of garbled inputs, using a hardwired public key and up to a constant number of hardwired ciphertexts. The contribution to the circuit size is bounded by $s^\alpha \cdot \text{poly}(\lambda, \sigma) = s^\alpha \cdot \text{poly}(\lambda, d, \log(s), n)$.

Overall,

$$\text{pad}_{\text{IGC}} \leq s^\alpha \cdot \text{poly}(\lambda, n, d, \log(s)) .$$

- Any circuit in \mathcal{S}_{GGC} performs a gate garbling operation (possibly via one of the hybrid gate garbling circuits). The overall contribution to the circuit size is bounded by

$$\text{pad}_{\text{GGC}} \leq \sigma \leq \text{poly}(\lambda, n, d, \log(s)) .$$

Succinctness

We show that for any $\beta < \gamma < 1$ and an appropriate choice of $\alpha < 1$, the size of the encryption circuit is bounded by $s^\gamma \cdot \text{poly}(\lambda, n, d)$.

Let SxiO be β -compressing. Let e be the size of the obfuscated circuits $\widetilde{\text{IGC}}$ and $\widetilde{\text{GGC}}$ created during encryption. Since the input spaces of IGC and GGC are, respectively, $[2s]$ and $[L]$, SXIO guarantees:

$$\begin{aligned} e &\leq (2s)^\beta \cdot \text{poly}(\lambda, |\text{IGC}|) + L^\beta \cdot \text{poly}(\lambda, |\text{GGC}|) \\ &\leq s^\beta \cdot \text{poly}(\lambda, \text{pad}_{\text{IGC}}) + L^\beta \cdot \text{poly}(\lambda, \text{pad}_{\text{GGC}}) , \end{aligned}$$

where poly is a fixed polynomial. Then, using the above bounds on pad_{IGC} and pad_{GGC} , and denoting by $c = O(1)$ the polynomial blowup in the circuit size incurred in SxiO , we obtain:

$$e \leq s^{\beta+c\alpha} \cdot \text{poly}(\lambda, n, d, \log(s)) + L^\beta \cdot \text{poly}(\lambda, n, d, \log(s)) .$$

Recalling that L is bounded by $s^3 \cdot \text{polylog}(s)$, and that $3\beta < \gamma$, we deduce that

$$e \leq s^\gamma \cdot \text{poly}(\lambda, n, d) ,$$

provided that we choose α such that $\beta + \alpha c < \gamma$ (which is possible since $\beta < \gamma$ and c is a constant).

Security Proof

Let $x_0, x_1 \in \{0, 1\}^n$ denote the challenge messages and f denote the function query provided by the adversary, and assume $f(x_0) = f(x_1)$. Let $U^{(x_0)}$ and $U^{(x_1)}$ be the universal circuits defined similarly to $U^{(x)}$ above, and $U_{\text{de}}^{(x_0)}$ and $U_{\text{de}}^{(x_1)}$ their respective

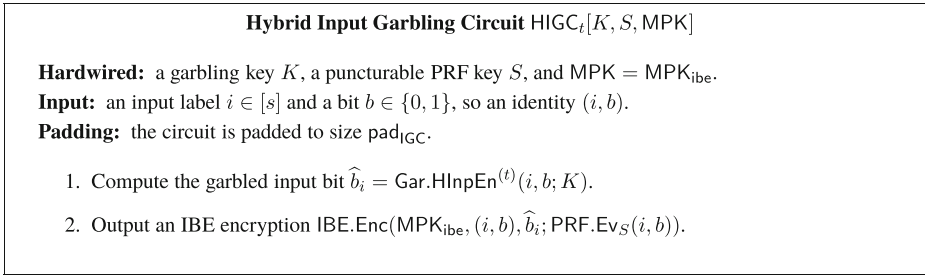


Fig. 7. Circuit $\text{HIGC}_t[K, S, \text{MPK}]$.

decomposition circuits. Note that $U^{(x_0)}$ and $U^{(x_1)}$ have the same topology (they are the same universal circuit with a different hardwired input of the same size n), and in particular, their decomposition circuits are of the same size. Furthermore, these two circuits also satisfy

$$U^{(x_0)}(f) = f(x_0) = f(x_1) = U^{(x_1)}(f) .$$

$U^{(x_0)}$ and $U^{(x_1)}$ satisfy the properties required by the (σ, τ, δ) -decomposable indistinguishability security of the decomposable garbling scheme, and thus, there exist circuits $\{\text{Gar.HInpEn}^{(t)}, \text{Gar.HCirEn}^{(t)} \mid t \in [\tau]\}$, whose size is at most σ , as well as (possibly inefficient) hybrid functions $\{\text{Gar.HPunc}^{(t)} \mid t \in [\tau]\}$ as given by Definition 4.1.

We accordingly consider a sequence of $\tau + 2$ hybrid games:

Hyb₀: The first game is $\text{Expt}_{\mathcal{A}}^{\text{FE}}(1^\lambda, 0)$, where the adversary hands x_0, x_1, f and gets back from the challenger an encryption of x_0 .

Hyb_t ($1 \leq t \leq \tau$): The t -th game is defined similarly to $\text{Expt}_{\mathcal{A}}^{\text{FE}}(1^\lambda, 0)$ except that the challenge ciphertext CT^* consists of obfuscations of corresponding hybrid circuits:

$$\text{CT}^* = (\text{sxiO}(1^\lambda, \text{HIGC}_t[K, S, \text{MPK}]), \text{sxiO}(1^\lambda, \text{HGGC}_t[K])) ,$$

where circuits $\text{HIGC}_t[K, \text{MPK}]$ and $\text{HGGC}_t[K]$ are defined in Fig. 7 and Fig. 8, respectively, and K is the key used in the decomposable garbling scheme for computing the challenge ciphertext.

Hyb_{τ+1}: The last game is $\text{Expt}_{\mathcal{A}}^{\text{FE}}(1^\lambda, 1)$, where x_1 is encrypted (rather than x_0).

We first note that $\text{Hyb}_0 \approx_\delta \text{Hyb}_1$ and $\text{Hyb}_\tau \approx_\delta \text{Hyb}_{\tau+1}$. Indeed, by the guarantee of the decomposable garbling scheme (Definition 4.2), the obfuscated circuits in the respective hybrids compute the exact same function (and are padded to the same size). δ -indistinguishability follows by the SXIO guarantee.

We now argue that hybrids Hyb_t and Hyb_{t+1} , for $1 \leq t \leq \tau - 1$, are computationally indistinguishable.

Indistinguishability of Hyb_t and Hyb_{t+1} ($1 \leq t \leq \tau - 1$)

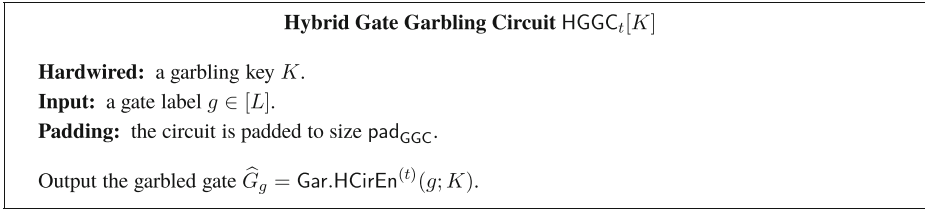


Fig. 8. Circuit $\text{HGGC}_t[K]$.

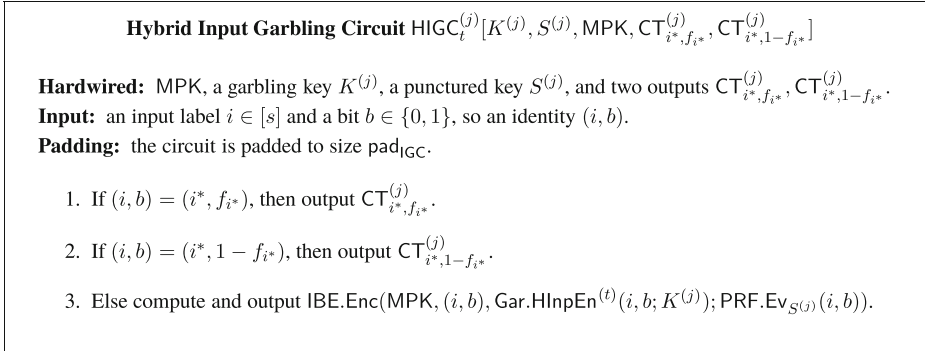


Fig. 9. Circuit $\text{HIGC}_t^{(j)}[K^{(j)}, S^{(j)}, \text{MPK}, \text{CT}_{i^*, f_{i^*}}^{(j)}, \text{CT}_{i^*, 1-f_{i^*}}^{(j)}]$, $1 \leq j \leq 5$.

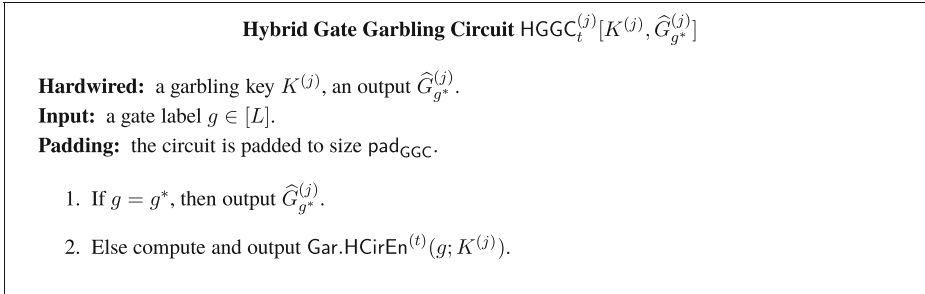


Fig. 10. Circuit $\text{HGGC}_t^{(j)}[K^{(j)}, \widehat{G}_{g^*}^{(j)}]$, $1 \leq j \leq 5$.

The indistinguishability of each such two hybrid games is proven by a sequence of five intermediate hybrid games $\text{Hyb}_t^{(1)}, \dots, \text{Hyb}_t^{(5)}$ as follows. The five hybrid games only differ from Hyb_t in the way the challenge ciphertext CT^* is computed. For $1 \leq j \leq 5$, the challenge ciphertext is computed in $\text{Hyb}_t^{(j)}$ as the obfuscations of the two circuits $\text{HIGC}_t^{(j)}$ and $\text{HGGC}_t^{(j)}$, described in Figs. 9 and 10, where the role of the hardwired values $K^{(j)}, S^{(j)}, \text{CT}_{i^*, f_{i^*}}^{(j)}, \text{CT}_{i^*, 1-f_{i^*}}^{(j)}$ and $\widehat{G}_{g^*}^{(j)}$ is described below. We also let $(K^*, g^*, i^*) \leftarrow \text{Gar.HPunc}^{(t)}(K)$. Then, f_{i^*} denotes the i^* -th bit of the function queried by the adversary.

We now prove indistinguishability of Hyb_t and Hyb_{t+1} as follows:

- **Hyb_t to Hyb_t⁽¹⁾**: in Hyb_t⁽¹⁾, we fix $S^{(1)} \leftarrow \text{PRF.Punc}(S, \{(i^*, 1 - f_{i^*})\})$ and $K^{(1)} \leftarrow K^*$. We hardwire the value $\text{CT}_{i^*, f_{i^*}}^{(1)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, f_{i^*}), \text{Gar.HInpEn}^{(t)}(i^*, f_{i^*}; K); \text{PRF.Ev}_{S^{(1)}}(i^*, f_{i^*}))$, as well as $\text{CT}_{i^*, 1-f_{i^*}}^{(1)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, 1 - f_{i^*}), \text{Gar.HInpEn}^{(t)}(i^*, 1 - f_{i^*}; K); \text{PRF.Ev}_S(i^*, 1 - f_{i^*}))$ in the first circuit, and $\widehat{G}_{g^*}^{(1)} \leftarrow \text{Gar.HCirEn}^{(t)}(g^*; K)$ in the second circuit. Therefore, it is immediate that, by definitions, both circuits are functionally equivalent in both games since hardwired values are correctly computed, so the two hybrid games are computationally indistinguishable under the security of sxiO . Hence, we have $\text{Hyb}_t \approx_\delta \text{Hyb}_t^{(1)}$.
- **Hyb_t⁽¹⁾ to Hyb_t⁽²⁾**: The difference between Hyb_t⁽¹⁾ and Hyb_t⁽²⁾ is that we now define $\text{CT}_{i^*, f_{i^*}}^{(2)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, f_{i^*}), \text{Gar.HInpEn}^{(t+1)}(i^*, f_{i^*}; K); \text{PRF.Ev}_{S^{(2)}}(i^*, f_{i^*}))$ in the first circuit, and $\widehat{G}_{g^*}^{(2)} \leftarrow \text{Gar.HCirEn}^{(t+1)}(g^*; K)$. Other values remain the same as before. Thus, these two hybrid games are computationally indistinguishable assuming decomposable indistinguishability of the decomposition garbling scheme, and we have $\text{Hyb}_t^{(1)} \approx_\delta \text{Hyb}_t^{(2)}$.
- **Hyb_t⁽²⁾ to Hyb_t⁽³⁾**: In the third hybrid game, we only modify the previous game by letting $\text{CT}_{i^*, 1-f_{i^*}}^{(3)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, 1 - f_{i^*}), \text{Gar.HInpEn}^{(t)}(i^*, 1 - f_{i^*}; K); r)$, with r being a fresh uniformly random value. Other values remain the same as before. Assuming PRF.Ev is a δ -secure puncturable pseudorandom function, these two hybrid games are computationally indistinguishable, and we have $\text{Hyb}_t^{(2)} \approx_\delta \text{Hyb}_t^{(3)}$.
- **Hyb_t⁽³⁾ to Hyb_t⁽⁴⁾**: In the fourth hybrid game, we define $\text{CT}_{i^*, 1-f_{i^*}}^{(4)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, 1 - f_{i^*}), \text{Gar.HInpEn}^{(t+1)}(i^*, 1 - f_{i^*}; K); r)$, with r being a fresh uniformly random value. Other values remain the same as before. Assuming the security of the identity-based encryption scheme, these two hybrid games are computationally indistinguishable, and we have $\text{Hyb}_t^{(3)} \approx_\delta \text{Hyb}_t^{(4)}$. The reduction is immediate but crucially relies on the fact that $\text{CT}_{i^*, 1-f_{i^*}}^{(4)}$ is computed as an encryption for the identity $(i^*, 1 - f_{i^*})$ and not for the identity (i^*, f_{i^*}) , since the secret key for identity (i^*, f_{i^*}) is revealed to the adversary when giving the functional secret key for f , but not the secret key for identity $(i^*, 1 - f_{i^*})$.
- **Hyb_t⁽⁴⁾ to Hyb_t⁽⁵⁾**: In this fifth hybrid game, we once again use randomness computed with the PPRF by letting $\text{CT}_{i^*, 1-f_{i^*}}^{(5)} \leftarrow \text{IBE.Enc}(\text{MPK}, (i^*, 1 - f_{i^*}), \text{Gar.HInpEn}^{(t+1)}(i^*, 1 - f_{i^*}; K); \text{PRF.Ev}_S(i^*, 1 - f_{i^*}))$. Other values remain the same as before. Once again, assuming PRF.Ev is a secure puncturable pseudorandom function, we have $\text{Hyb}_t^{(4)} \approx_\delta \text{Hyb}_t^{(5)}$.
- **Hyb_t⁽⁵⁾ to Hyb_{t+1}**: By definition, it is clear that the two circuits used in Hyb_t⁽⁵⁾ are functionally equivalent to the two circuits used in Hyb_t⁽⁵⁾, since $\text{Gar.HInpEn}^{(t)}(i, b; K^{(5)}) = \text{Gar.HInpEn}^{(t+1)}(i, b; K)$ for any (i, b) with that $i \neq i^*$. Therefore, under the security of sxiO , these two hybrid games are computationally indistinguishable, and we have $\text{Hyb}_t^{(5)} \approx_\delta \text{Hyb}_{t+1}$.

This concludes the proof of Theorem 5.2. \square

5.3. Weakly Succinct PKFE for Non-boolean Functions

In this section, we give a transformation from weakly succinct PKFE schemes for boolean functions into ones for *non-boolean* functions.

Theorem 5.3. *Let $\mathcal{C} = \{C_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of circuits (with multiple output bits) and let $n(\lambda)$, $s(\lambda)$, $d(\lambda)$ be bounds on their input length, size, and depth (respectively). For any constants $\beta < \gamma < 1$, assuming a β -compressing SXIO for \mathbf{P}/poly , there exists a constant α , such that given any α -compressing weakly succinct PKFE for boolean functions of size $s \cdot \text{polylog}(s)$ and depth $O(d)$, and one-way functions, there exists a succinct PKFE for \mathcal{C} with compression factor γ . If all primitives are δ -secure so is the resulting scheme.*

The transformation is essentially the same transformation presented in [51, Section 4], with the following differences:

- They use XIO rather than SXIO, which results in a PKFE scheme where only the size of ciphertexts is compressed, whereas the time to encrypt may be large. They then make an extra step, based on LWE, to make encryption efficient. Using SXIO directly as we do, allows avoiding this step.
- They start from weakly succinct PKFE for boolean functions where the size of ciphertexts is completely independent of the size s of the function class considered. Due to this, they can start from XIO with any compression factor $\beta < 1$. In our notion of weakly succinct, there is dependence on s^α , for some $\alpha < 1$, and we need to make sure that β and α are appropriately chosen to account for this.
- As stated, their notion of weak succinctness for PKFE does not explicitly scale with the depth of the function class considered. Eventually, they apply their transformation to function classes in NC^1 , assuming puncturable PRFs in NC^1 (which exist under LWE). Our succinctness notion allows polynomial dependence on the depth, which should be roughly preserved through the transformation.

For the sake of completeness, we describe the transformation in full and then analyze the efficiency aspects. The proof of security is identical to the one in [51] and is omitted.

Construction Let $\text{BFE} = (\text{BFE.Setup}, \text{BFE.Key}, \text{BFE.Enc}, \text{BFE.Dec})$ be a weakly succinct single-key public-key FE scheme for boolean functions. Our weakly succinct single-key public-key FE scheme for multi-bit functions is given by the following algorithms.

Setup(1^λ):

- Compute $(\text{BFE.MPK}, \text{BFE.MSK}) \leftarrow \text{BFE.Setup}(1^\lambda)$.
- Output $(\text{MPK}, \text{MSK}) = (\text{BFE.MPK}, \text{BFE.MSK})$.

KeyGen(MSK, f):

- Run $\text{BFE.SK}_{C_f} \leftarrow \text{BFE.Key}(\text{BFE.MSK}, C_f)$ where $C_f : \{0, 1\}^n \times \{0, 1\}^{\log \ell}$ is a function that takes (x, i) as inputs and outputs the i -th bit (out of ℓ) of $f(x)$ (this is a boolean function).

Enc($\text{MPK}, x \in \{0, 1\}^n$):

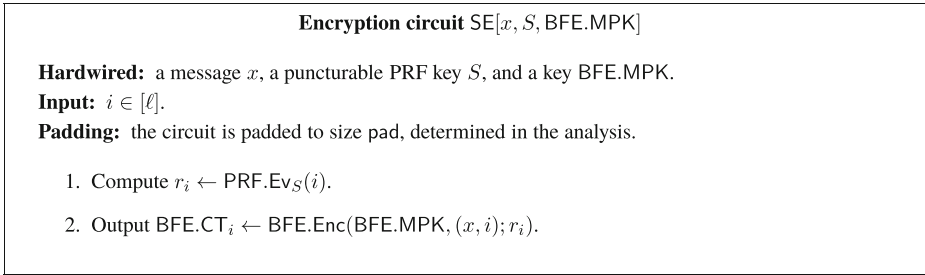


Fig. 11. Circuit $\text{SE}[x, S, \text{BFE.MPK}]$.

- Sample a puncturable PRF key $S \leftarrow \text{PRF.Gen}(1^\lambda)$,
- Generate $\text{sxiO}(\text{SE}[x, S, \text{BFE.MPK}])$ where the circuit $\text{SE}[x, S, \text{BFE.MPK}]$ is described in Fig. 11.
- Output $\text{CT} = \text{sxiO}(\text{SE}[x, S, \text{BFE.MPK}])$.

$\text{Dec}(\text{SK}_f, \text{CT}_x)$:

- Compute $\text{BFE.CT}_i \leftarrow \text{CT}(i)$, and $y_i \leftarrow \text{BFE.Dec}(\text{BFE.SK}_{C_f}, \text{BFE.CT}_i)$ for all $i \in [\ell]$.
- Output $y = (y_1, \dots, y_\ell)$.

Proof of Theorem 5.3. We now prove Theorem 5.3. We focus on succinctness. The proof of security is based on a standard probabilistic IO argument [25], is identical to the one given by Lin et al. [51, Theorem 6], and thus omitted. In what follows, let s , d , n , and ℓ be bounds on the size, the depth, input and output lengths of functions in \mathcal{C} .

Padding Parameter

In the proof of security [51], the indistinguishability guarantee of SXIO is invoked for the circuit SE and several hybrid versions of this circuit. All of these circuits compute a puncturable pseudorandom function on an input $i \in [\ell]$ and an encryption using BFE.Enc . Therefore, their size is bounded by $\text{poly}(\lambda, \log(\ell)) + e'$ where e' is the size of the encryption circuit for the class of boolean circuits $\mathcal{C}' = \{C_f \mid f \in \mathcal{C}\}$.

To bound e' , we note that the size and depth of circuits in \mathcal{C}' are preserved up to polylogarithmic factors (induced by selecting one of $\ell \leq s$ output bits):

$$s' := |C_f| \leq s \cdot \text{polylog}(s) \quad \text{and} \quad d' := \text{depth}(C_f) \leq O(d) .$$

Thus, by the succinctness guarantee of the PKFE for boolean functions, we know that

$$e' \leq s'^{\alpha} \cdot \text{poly}(d', \lambda, n + \log \ell) \leq s^{\alpha} \cdot \text{poly}(d, \log(s), \lambda, n) ,$$

where α is the compression factor of the PKFE. Hence, we obtain:

$$\text{pad} \leq s^{\alpha} \cdot \text{poly}(d, \log(s), \lambda, n) .$$

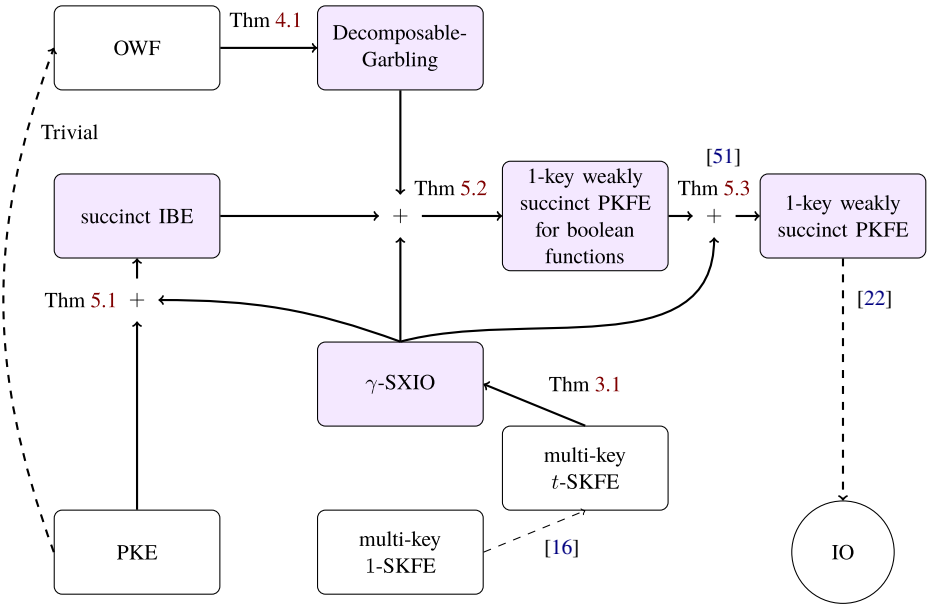


Fig. 12. An illustration of the path from PKE and 1-SKFE to IO. Dashed lines denote known results. White backgrounds denote our ingredients or goal. Primitives in rounded rectangles are subexponentially secure. γ -SXIO denotes SXIO with compression factor γ , which is an arbitrary constant less than 1. We ignore puncturable PRF in this figure since it is implied by OWF.

Succinctness

We show that for any $\beta < \gamma < 1$ and an appropriate choice of $\alpha < 1$, the size of the encryption circuit is bounded by $s^\gamma \cdot \text{poly}(\lambda, n, d)$.

Let sxiO be β -compressing. Then, by the SXIO guarantee, the size e of the obfuscated encryption circuit is bounded by

$$e \leq \ell^\beta \cdot \text{poly}(\lambda, |\text{SE}|) \leq s^\beta \cdot \text{poly}(\lambda, \text{pad}) ,$$

where SE is the obfuscated encryption circuit, and poly is a fixed polynomial. Then, letting $c = O(1)$ represent the polynomial blowup in circuit size incurred by sxiO , we can bound

$$e \leq s^\beta \cdot s^{c \cdot \alpha} \cdot \text{poly}(d, n, \lambda, \log(s)) \leq s^\gamma \cdot \text{poly}(d, n, \lambda) ,$$

where the last inequality holds provided that we choose the constant α such that $c\alpha + \beta < \gamma$. (Such a constant indeed exist since $\beta < \gamma$ and c is a constant.) □

5.4. Putting It All Together: From SKFE and PKE to IO

We now show how the results proved in this section come together to give our main result. The main implications are also illustrated in Fig. 12.

Theorem 5.4. *Let $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of circuits (with multiple output bits) and let $n(\lambda)$, $s(\lambda)$, $d(\lambda)$ be bounds on their input length, size, and depth (respectively). Then, for any constant $\gamma < 1$, there exists a constant β , such that given any δ -secure, β -compressing SXIO for \mathbf{P}/poly , and δ -secure PKE, there exists $2^d s \delta$ -secure, γ -compressing, weakly succinct PKFE for \mathcal{C} .*

Proof. Fix some constant γ . The theorem follows from the previous theorems in this section as follows:

- By Theorem 5.3, there exist constants β' , α' , such that $2^d s \delta$ -secure, γ -compressing, weakly succinct PKFE for \mathcal{C} , follows from $2^d s \delta$ -secure, β' -compressing SXIO and $2^d s \delta$ -secure, α' -compressing, weakly succinct PKFE for boolean functions of size $s \cdot \text{polylog}(s)$ and depth $O(d)$, and $2^d s \delta$ -secure one-way functions.
- By Theorem 5.2, there exist constants β'' , α'' such that PKFE as required in the previous item follows from δ -secure, β'' -compressing SXIO and δ -secure, α'' -compressing IBE, and δ -secure one-way functions.
- By Theorem 5.1, there exists a constant β''' (in fact any $\beta''' < \alpha''$), such that IBE as required in the previous item follows from δ -secure, β''' -compressing SXIO and δ -secure PKE.
- In all of the above δ -secure one-way functions follow from δ -secure PKE.

Setting $\beta := \min(\beta', \beta'', \beta''')$, we derive the theorem. \square

Combining the above theorem with the result from Sect. 3, we obtain the following corollary.

Corollary 5.1. *If there exist (1-input) SKFE for \mathbf{P}/poly and PKE, both subexponentially secure, then there exists IO for \mathbf{P}/poly .*

Proof. Fix any constants ε , $\gamma < 1$. For $\delta(\lambda) = 2^{-\lambda^\varepsilon}$.

- By Theorem 2.2 [22], for any constant $\gamma < 1$, IO is implied by δ -secure, γ -compressing, weakly succinct PKFE.
- By the Theorem 5.4, there exists β , such that PKFE as required in the previous item follows from β -compressing, δ -secure SXIO and δ -secure PKE, when setting the security parameter for the SXIO and PKE to $\tilde{\lambda} = \lambda \cdot d^{2/\varepsilon}$ —this accounts for the $\text{poly}(2^d)$ security loss, and only incurs $\text{poly}(\lambda, d, \log(s))$ overhead, which satisfies the succinctness requirements of the PKFE.
- By Theorem 3.1, letting $t = \frac{1}{\beta} - 1$, SXIO as required in the previous item follows from δ -secure t -input SKFE.
- By Theorem 2.1 [16], the required t -input SKFE follows from any 1-input, δ -secure SKFE.

The corollary follows. \square

Remark 5.1. (The security loss) In order, the known reductions [3,22] of IO to weakly succinct PKFE incur a subexponential loss. Accordingly, reducing IO to SKFE based on our results incurs a similar loss. However, when restricting attention, to the transformation from SKFE to (weakly succinct) PKFE, then the loss is $\text{poly}(2^d, \lambda)$, for circuits of

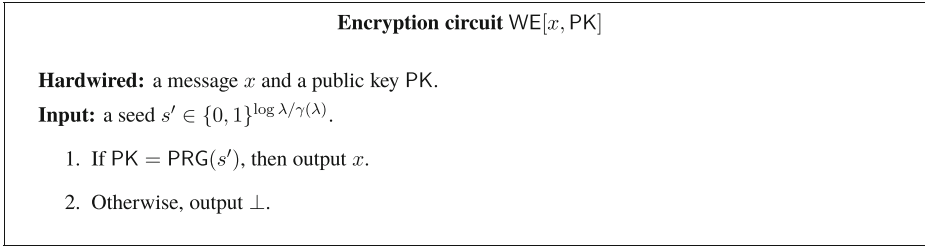


Fig. 13. Circuit $WE[x, PK]$.

depth d . In particular, for NC^1 , our transformation incurs only polynomial security loss. Such a PKFE for NC^1 can then be bootstrapped to all polynomial-size circuits using the transformation of [1], and assuming also weak PRFs in NC^1 .

In concurrent work [36,50], it is shown that weakly succinct single-key PKFE can then be polynomially reduced to PKFE. In summary, SKFE and PRFs in NC^1 can be polynomially reduced to PKFE for all polynomial-size circuits.

6. Polynomially Secure PKE from Secret-Key FE

In this section, we construct PKE from SKFE. Our starting point is Corollary 3.1 that directly follows from Theorems 2.1 and 3.1.

We now show how to construct a PKE scheme from such SXIO.

The construction Let $\{PRG : \{0, 1\}^n \rightarrow \{0, 1\}^{2n}\}_{n \in \mathbb{N}}$ be a length-doubling pseudorandom generator that is $2^{-n/\log \log n}$ -secure. Let $sxiO$ be a SXIO with compression factor $\gamma(\lambda) = O(1/\log \log \lambda)$ (and $\text{poly}(\lambda)$ security) for circuits of size at most $2^{O((\log \lambda)^\epsilon)}$. The scheme $PKE = (\text{KeyGen}, \text{Enc}, \text{Dec})$ is defined as follows:

KeyGen(1^λ):

- Sample a PRG seed $s \leftarrow \{0, 1\}^{\log \lambda / \gamma(\lambda)}$.
- Output $PK = PRG(s)$ and $SK = s$.

Enc(PK, x):

- Construct the (witness encryption) circuit $WE[x, PK]$ described in Fig. 13 that takes $s' \in \{0, 1\}^{\log \lambda / \gamma(\lambda)}$ as input and outputs x if and only if $PK = PRG(s')$ holds.
- Output $CT = sxiO(WE[x, PK])$

Dec(SK, CT):

- Compute $x' = CT(SK)$.

Proposition 6.1. *PKE is a (polynomially secure) public-key encryption scheme.*

Proof sketch. The scheme is clearly correct. We now note that the scheme is efficient; namely, all algorithms run in time $\text{poly}(\lambda)$ in the security parameter λ . This is clearly the

case for **KeyGen** and **Dec**. As for **Enc**, by the **SXIO** compression guarantee, the time required to compute the obfuscation is $2^{\gamma(\lambda) \cdot \log \lambda / \gamma(\lambda)} \cdot \text{poly}(\lambda, |\text{WE}[b, \text{PK}]|) = \text{poly}(\lambda)$. Note that we can apply **sxiO** to $\text{WE}[b, \text{PK}]$ since $|\text{WE}[b, \text{PK}]|$ is bounded by $\text{poly}(\log \lambda)$ and it is less than $2^{O((\log \lambda)^\epsilon)}$. (This bound comes from Corollary 3.1.)

We turn to prove that the scheme is semantically secure. For this purpose, we first consider an alternative encryption scheme where the public key is generated as a truly random string $\text{PK} \leftarrow \{0, 1\}^{2 \log \lambda / \gamma(\lambda)}$. By the security of the PRG, any polysize distinguisher cannot tell apart a real public key from such a fake public key except with advantage

$$2^{-\Omega(n(\lambda) / \log \log n(\lambda))} = 2^{-\Omega\left(\frac{\log \lambda / \gamma(\lambda)}{\log \log (\log \lambda / \gamma(\lambda))}\right)} \leq 2^{-\Omega\left(\frac{-\log \lambda \cdot \log \log \lambda}{\log \log \log \lambda}\right)} = 2^{-\omega(\log \lambda)} .$$

We next note that, by a union bound over all possible seeds, a random **PK** does not have any preimage under **PRG**, except with probability

$$2^{\log \lambda / \gamma(\lambda)} \cdot 2^{-2 \log \lambda / \gamma(\lambda)} = 2^{-\log \lambda / \gamma(\lambda)} \leq 2^{-\omega(\log \lambda)} .$$

In this case, $\text{WE}[b, \text{PK}]$ is functionally equivalent to a circuit $\text{WE}[\perp, \text{PK}]$ that is independent of b and always outputs \perp . Security thus follows from the usual IO guarantee. \square

Acknowledgements

We thank Vinod Vaikuntanathan and Hoeteck Wee for valuable discussions.

References

- [1] P. Ananth, Z. Brakerski, G. Segev, V. Vaikuntanathan, From selective to adaptive security in functional encryption, in R. Gennaro, M.J.B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II, Santa Barbara, CA, USA, August 16–20, 2015*. LNCS, vol. 9216 (Springer, Heidelberg, Germany, 2015), pp. 657–677
- [2] S. Agrawal, S. Gorbunov, V. Vaikuntanathan, H. Wee, Functional encryption: new perspectives and lower bounds, in R. Canetti, J.A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II, Santa Barbara, CA, USA, August 18–22, 2013*. LNCS, vol. 8043 (Springer, Heidelberg, Germany, 2013), pp. 500–518
- [3] P. Ananth, A. Jain, Indistinguishability obfuscation from compact functional encryption, in R. Gennaro, M.J.B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part I, Santa Barbara, CA, USA, August 16–20, 2015*. LNCS, vol. 9215 (Springer, Heidelberg, Germany, 2015), pp. 308–326
- [4] P. Ananth, A. Jain, A. Sahai, Indistinguishability obfuscation from functional encryption for simple functions. Cryptology ePrint Archive, Report 2015/730 (2015). <http://eprint.iacr.org/2015/730>.
- [5] P.V. Ananth, A. Sahai, Functional encryption for turing machines, in E. Kushilevitz, T. Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I, Tel Aviv, Israel*. LNCS, vol. 9562 (Springer, Heidelberg, Germany, January 10–13, 2016), pp. 125–153
- [6] G. Asharov, G. Segev, Limits on the power of indistinguishability obfuscation and functional encryption. *SIAM J. Comput.* **45**(6), 2117–2176 (2016)
- [7] P. Ananth, A. Sahai, Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps, in J.-S. Coron, J.B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT*

- 2017, Part I, Paris, France, May 8–12, 2017. LNCS, vol. 10210 (Springer, Heidelberg, Germany), pp. 152–181
- [8] N. Bitansky, R. Canetti, S. Garg, J. Holmgren, A. Jain, H. Lin, R. Pass, S. Telang, V. Vaikuntanathan, Indistinguishability obfuscation for RAM programs and succinct randomized encodings. *SIAM J. Comput.* **47**(3), 1123–1210 (2018)
- [9] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, D. Vinayagamurthy, Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits, in P.Q. Nguyen, E. Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014, Copenhagen, Denmark, May 11–15, 2014*. LNCS, vol. 8441 (Springer, Heidelberg, Germany, 2014), pp. 533–556
- [10] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S.P. Vadhan, K. Yang. On the (im)possibility of obfuscating programs. *JACM* **59**(2), 6 (2012)
- [11] E. Boyle, S. Goldwasser, I. Ivan, Functional signatures and pseudorandom functions, in H. Krawczyk, editor, *PKC 2014: 17th International Conference on Theory and Practice of Public Key Cryptography, Buenos Aires, Argentina, March 26–28, 2014*. LNCS, vol. 8383 (Springer, Heidelberg, Germany, 2014), pp. 501–519
- [12] S. Badrinarayanan, D. Gupta, A. Jain, A. Sahai, Multi-input functional encryption for unbounded arity functions, in T. Iwata and J.H. Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I, Auckland, New Zealand, November 30–December 3, 2015*. LNCS, vol. 9452 (Springer, Heidelberg, Germany, 2015), pp. 27–51
- [13] N. Bitansky, S. Garg, H. Lin, R. Pass, S. Telang, Succinct randomized encodings and their applications, in R.A. Servedio, R. Rubinfeld, editors, *47th Annual ACM Symposium on Theory of Computing, Portland, OR, USA, June 14–17, 2015*. (ACM Press, 2015), pp. 439–448
- [14] M. Bellare, V.T. Hoang, P. Rogaway. Foundations of garbled circuits, in T. Yu, G. Danezis, V.D. Gligor, editors, *ACM CCS 12: 19th Conference on Computer and Communications Security, Raleigh, NC, USA, October 16–18, 2012* (ACM Press, 2012), pp. 784–796
- [15] Z. Brakerski, A. Jain, I. Komargodski, A. Passelègue, D. Wichs, Non-trivial witness encryption and null-IO from standard assumptions, in *SCN 18: 11th International Conference on Security in Communication Networks*. LNCS (Springer, Heidelberg, Germany, 2018), pp. 425–441
- [16] Z. Brakerski, I. Komargodski, G. Segev, Multi-input functional encryption in the private-key setting: stronger security from weaker assumptions. *J. Cryptol.* **31**(2), 434–520 (April 2018)
- [17] N. Bitansky, H. Lin, O. Paneth, On removing graded encodings from functional encryption, in J.-S. Coron, J.B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II, Paris, France, 2017, May 8–12, 2017*. LNCS, vol. 10211 (Springer, Heidelberg, Germany, 2017), pp. 3–29
- [18] D. Boneh, K. Lewi, M. Raykova, A. Sahai, M. Zhandry, J. Zimmerman, Semantically secure order-revealing encryption: multi-input functional encryption without obfuscation, in E. Oswald, M. Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part II, Sofia, Bulgaria, April 26–30, 2015*. LNCS, vol. 9057 (Springer, Heidelberg, Germany, 2015), pp. 563–594
- [19] M. Bellare, A. O’Neill, Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition, in M. Abdalla, C. Nita-Rotaru, R. Dahab, editors, *CANS 13: 12th International Conference on Cryptology and Network Security, Paraty, Brazil, November 20–22, 2013*. LNCS, vol. 8257 (Springer, Heidelberg, Germany, 2013), pp. 218–234
- [20] Z. Brakerski, G. Segev, Function-private functional encryption in the private-key setting. *J. Cryptol.* **31**(1), 202–225 (January 2018)
- [21] D. Boneh, A. Sahai, B. Waters, Functional encryption: definitions and challenges, in Y. Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference, Providence, RI, USA, March 28–30, 2011*. LNCS, vol. 6597 (Springer, Heidelberg, Germany, 2011), pp. 253–273
- [22] N. Bitansky, V. Vaikuntanathan, Indistinguishability obfuscation from functional encryption. *JACM* **65**(6), 39:1–39:37 (2018)
- [23] D. Boneh, B. Waters, Constrained pseudorandom functions and their applications, in K. Sako, P. Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part II, Bangalore, India, December 1–5, 2013*. LNCS, vol. 8270 (Springer, Heidelberg, Germany, 2013), pp. 280–300
- [24] S.A. Cook, H. James Hoover, A depth-universal circuit. *SIAM J. Comput.* **14**(4), 833–839 (1985)
- [25] R. Canetti, H. Lin, S. Tessaro, V. Vaikuntanathan, Obfuscation of probabilistic circuits and applications, in Y. Dodis, J.B. Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II, Warsaw, Poland, March 23–25, 2015*. LNCS, vol. 9015 (Springer, Heidelberg, Germany, 2015), pp. 468–497

- [26] A. De Caro, V. Iovino, A. Jain, A. O'Neill, O. Paneth, G. Persiano, On the achievability of simulation-based security for functional encryption, in R. Canetti, J.A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II, Santa Barbara, CA, USA, August 18–22, 2013*. LNCS, vol. 8043 (Springer, Heidelberg, Germany, 2013), pp. 519–535
- [27] S. Goldwasser, S. Dov Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, H.-S. Zhou, Multi-input functional encryption, in P.Q. Nguyen, E. Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014, Copenhagen, Denmark, May 11–15, 2014*. LNCS, vol. 8441 (Springer, Heidelberg, Germany, 2014), pp. 578–602
- [28] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, B. Waters, Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.* **45**(3), 882–929 (2016)
- [29] S. Garg, C. Gentry, S. Halevi, M. Zhandry, Functional encryption without obfuscation, in E. Kushilevitz and T. Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part II, Tel Aviv, Israel, January 10–13, 2016*. LNCS, vol. 9563 (Springer, Heidelberg, Germany, 2016), pp. 480–511
- [30] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions. *JACM* **33**(4), 792–807 (1986)
- [31] S. Garg, C. Gentry, A. Sahai, B. Waters, Witness encryption and its applications, in D. Boneh, T. Roughgarden, J. Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing, Palo Alto, CA, USA, June 1–4, 2013* (ACM Press, 2013), pp. 467–476
- [32] V. Goyal, A. Jain, V. Koppula, A. Sahai, Functional encryption for randomized functionalities, in Y. Dodis and J.B. Nielsen, editors, *TCC 2015: 12th Theory of Cryptography Conference, Part II, Warsaw, Poland, March 23–25, 2015*. LNCS, vol. 9015 (Springer, Heidelberg, Germany, 2015), pp. 325–351
- [33] S. Goldwasser, Y.T. Kalai, R.A. Popa, V. Vaikuntanathan, N. Zeldovich, Reusable garbled circuits and succinct functional encryption, in D. Boneh, T. Roughgarden, J. Feigenbaum, editors, *45th Annual ACM Symposium on Theory of Computing, Palo Alto, CA, USA, June 1–4, 2013* (ACM Press, 2013), pp. 555–564
- [34] S. Garg, O. Pandey, A. Srinivasan, Revisiting the cryptographic hardness of finding a nash equilibrium, in M. Robshaw, J. Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II, Santa Barbara, CA, USA, August 14–18, 2016*. LNCS, vol. 9815 (Springer, Heidelberg, Germany, 2016), pp. 579–604
- [35] S. Garg, O. Pandey, A. Srinivasan, M. Zhandry, Breaking the sub-exponential barrier in obfustopia, in J.-S. Coron, J.B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part II, Paris, France, May 8–12, 2017*. LNCS, vol. 10211 (Springer, Heidelberg, Germany, 2017), pp. 156–181
- [36] S. Garg, A. Srinivasan, Single-key to multi-key functional encryption with polynomial loss, in M. Hirt, A.D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II, Beijing, China, October 31–November 3, 2016*. LNCS, vol. 9986 (Springer, Heidelberg, Germany, 2016), pp. 419–442
- [37] S. Gorbunov, V. Vaikuntanathan, H. Wee, Functional encryption with bounded collusions via multi-party computation, in R. Safavi-Naini, R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012, Santa Barbara, CA, USA, August 19–23, 2012*. LNCS, vol. 7417 (Springer, Heidelberg, Germany, 2012), pp. 162–179
- [38] S. Gorbunov, V. Vaikuntanathan, H. Wee, Predicate encryption for circuits from LWE, in R. Gennaro, M.J.B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II, Santa Barbara, CA, USA, August 16–20, 2015*. LNCS, vol. 9216 (Springer, Heidelberg, Germany, 2015), pp. 503–523
- [39] B. Hemenway, Z. Jafargholi, R. Ostrovsky, A. Scafuro, D. Wichs, Adaptively secure garbled circuits from one-way functions, in M. Robshaw, J. Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part III, Santa Barbara, CA, USA, August 14–18, 2016*. LNCS, vol. 9816 (Springer, Heidelberg, Germany, 2016), pp. 149–178
- [40] R. Impagliazzo, A personal view of average-case complexity, in *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19–22, 1995* (IEEE Computer Society, 1995), pp. 134–147
- [41] F. Kitagawa, Ryo Nishimaki, Keisuke Tanaka, From single-key to collusion-resistant secret-key functional encryption by leveraging succinctness. Cryptology ePrint Archive, Report 2017/638 (2017). <http://eprint.iacr.org/2017/638>.
- [42] F. Kitagawa, R. Nishimaki, K. Tanaka, Indistinguishability obfuscation for all circuits from secret-key functional encryption. Cryptology ePrint Archive, Report 2017/361 (2017). <http://eprint.iacr.org/2017/361>.

- [43] F. Kitagawa, R. Nishimaki, K. Tanaka, Obfuscation built on secret-key functional encryption, in J.B. Nielsen, V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018, Part II*. LNCS, vol. 10821 (Springer, Heidelberg, Germany, 2018), pp. 603–648
- [44] F. Kitagawa, R. Nishimaki, K. Tanaka, Simple and generic constructions of succinct functional encryption, in M. Abdalla, R. Dahab, editors, *Advances in Cryptology – PKC 2018, Part II*. LNCS, vol. 10770 (Springer, Heidelberg, Germany, 2018), pp. 187–217
- [45] A. Kiayias, S. Papadopoulos, N. Triandopoulos, T. Zacharias, Delegatable pseudorandom functions and applications, in A.-R. Sadeghi, V.D. Gligor, M. Yung, editors, *ACM CCS 13: 20th Conference on Computer and Communications Security, Berlin, Germany, November 4–8, 2013* (ACM Press, 2013), pp. 669–684
- [46] I. Komargodski, G. Segev, From minicrypt to obfuscation via private-key functional encryption, in J.-S. Coron, J. B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I, Paris, France, May 8–12, 2017*. LNCS, vol. 10210 (Springer, Heidelberg, Germany, 2017), pp. 122–151
- [47] I. Komargodski, G. Segev, E. Yosev. Functional encryption for randomized functionalities in the private-key setting from minimal assumptions. *Journal of Cryptology* **31**(1), 60–100 (2018)
- [48] H. Lin, Indistinguishability obfuscation from constant-degree graded encoding schemes, in M. Fischlin, J.-S. Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I, Vienna, Austria, May 8–12, 2016*. LNCS, vol. 9665 (Springer, Heidelberg, Germany, 2016), pp. 28–57
- [49] H. Lin, Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs, in J. Katz, H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I, Santa Barbara, CA, USA, August 20–24, 2017*. LNCS, vol. 10401 (Springer, Heidelberg, Germany, 2017), pp. 599–629
- [50] B. Li, D. Micciancio, Compactness vs collusion resistance in functional encryption, in Martin Hirt and Adam D. Smith, editors, *TCC 2016-B: 14th Theory of Cryptography Conference, Part II, Beijing, China, October 31–November 3, 2016*. LNCS, vol. 9986 (Springer, Heidelberg, Germany, 2016), pp. 443–468
- [51] H. Lin, R. Pass, K. Seth, S. Telang, Indistinguishability obfuscation with non-trivial efficiency, in C.-M. Cheng, K.-M. Chung, G. Persiano, B.-Y. Yang, editors, *PKC 2016: 19th International Conference on Theory and Practice of Public Key Cryptography, Part II, Taipei, Taiwan, March 6–9, 2016*. LNCS, vol. 9615 (Springer, Heidelberg, Germany, 2016), pp. 447–462
- [52] H. Lin, R. Pass, K. Seth, S. Telang, Output-compressing randomized encodings and applications, in E. Kushilevitz, T. Malkin, editors, *TCC 2016-A: 13th Theory of Cryptography Conference, Part I, Tel Aviv, Israel, January 10–13, 2016*. LNCS, vol. 9562 (Springer, Heidelberg, Germany, 2016), pp. 96–124
- [53] H. Lin, S. Tessaro. Indistinguishability obfuscation from trilinear maps and block-wise local PRGs, in J. Katz, H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I, Santa Barbara, CA, USA, August 20–24, 2017*. LNCS, vol. 10401 (Springer, Heidelberg, Germany, 2017), pp. 630–660
- [54] H. Lin, V. Vaikuntanathan, Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings, in I. Dinur, editor, *57th Annual Symposium on Foundations of Computer Science, New Brunswick, NJ, USA, October 9–11, 2016* (IEEE Computer Society Press, 2016), pp. 11–20
- [55] A. O’Neill, Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010). <http://eprint.iacr.org/2010/556>
- [56] O. Regev, On lattices, learning with errors, random linear codes, and cryptography. *JACM* **56**(6), 34:1–34:40 (2009)
- [57] A. Shamir, Identity-based cryptosystems and signature schemes, in G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO’84, Santa Barbara, CA, USA, August 19–23, 1984*. LNCS, vol. 196 (Springer, Heidelberg, Germany, 1984), pp. 47–53
- [58] A. Sahai, H. Seyalioglu, Worry-free encryption: functional encryption with public keys, in E. Al-Shaer, A.D. Keromytis, V. Shmatikov, editors, *ACM CCS 10: 17th Conference on Computer and Communications Security, Chicago, Illinois, USA, October 4–8, 2010* (ACM Press, 2010), pp. 463–472
- [59] A. Sahai, B. Waters, How to use indistinguishability obfuscation: deniable encryption, and more, in D.B. Shmoys, editor, *46th Annual ACM Symposium on Theory of Computing, New York, NY, USA, May 31–June 3, 2014* (ACM Press, 2014), pp. 475–484
- [60] B. Waters, A punctured programming approach to adaptively secure functional encryption, in R. Gennaro, M.J.B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II, Santa Barbara, CA, USA, August 16–20, 2015*. LNCS, vol. 9216 (Springer, Heidelberg, Germany, 2015), pp. 678–697

- [61] A.C.-C. Yao, Protocols for secure computations (extended abstract), in *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, November 3–5, 1982* (IEEE Computer Society Press, 1982) pp. 160–164

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.