Indistinguishability Obfuscation: Simpler Constructions using Secret-Key Functional Encryption

Fuyuki Kitagawa*[1] Ryo Nishimaki [2] Keisuke Tanaka [1]

[1] Tokyo Institute of Technology, Japan
{kitagaw1,keisuke}@titech.ac.jp
[2] Secure Platform Laboratories, NTT Corporation, Japan
{nishimaki.ryo}@lab.ntt.co.jp

Abstract

We propose simple generic constructions of indistinguishability obfuscator (IO). Our key tool is exponentially-efficient indistinguishability obfuscator (XIO), which is the same as IO except that the size of an obfuscated circuit (or the running-time of an obfuscator) is *slightly* smaller than that of a brute-force canonicalizer that outputs the entire truth table of a circuit to be obfuscated. A "compression factor" of XIO indicates how much XIO compresses the brute-force canonicalizer. In this study, we show that XIO is a powerful enough to achieve cutting-edge cryptography. In particular, we propose the following constructions:

- A single-key weakly succinct secret-key functional encryption (SKFE) scheme is constructed from XIO (even with a bad compression factor) and one-way function.
- A single-key weakly succinct public-key functional encryption (PKFE) scheme is constructed from XIO with a good compression factor and public-key encryption scheme.
- A single-key weakly succinct PKFE scheme is constructed from XIO (even with a bad compression factor) and identity-based encryption scheme.

It is known that sub-exponentially secure single-key weakly succinct PKFE scheme implies IO and that single-key weakly succinct (resp. multi-key non-succinct) SKFE implies XIO with a bad (resp. good) compression factor. Thus, we developed two methods of constructing IO. One uses multi-key SKFE and plain public-key encryption schemes and the other uses single-key weakly succinct SKFE (or XIO) and identity-based encryption schemes. It is not known whether single-key weakly succinct SKFE implies IO (if we use fully black-box reduction in a certain model, it is impossible), but our single-key weakly succinct SKFE scheme gives many interesting by-products.

Keywords: Indistinguishability Obfuscation, Functional Encryption, Succinctness.

*This work was done while the author was visiting NTT Secure Platform Laboratories as a summer internship student.

Contents

1 Introduction

1.1 Background

Indistinguishability obfuscator (IO) converts computer programs into those that hide secret information in the original programs while preserving their functionalities. An obvious application of IO is protecting software from reverse engineering. Moreover, IO enables us to achieve many cutting-edge cryptographic tasks that other standard cryptographic tools do (or can) not achieve such as (collusion-resistant) functional encryption, program watermarking, and deniable encryption [SW14, GGH+13b, CHN+16].

Many IO constructions have been proposed since the celebrating invention of a candidate IO by Garg, Gentry, Halevi, Raykova, Sahai, and Waters [GGH+13b]. However, regarding designing secure IO, we are still at the "embryonic" stage[1] and understand little of how to construct secure IO. Roughly speaking, there are two main methods of constructing IO. One is instantiating IO concretely by using graded encoding schemes [GGH+13b, BGK+14, BR14, AGIS14, PST14, Zim15, AB15, BMSZ16, GMM+16, Lin16a, LV16, Lin16b, AS16]. A few candidates of graded encoding schemes have been proposed [GGH13a, CLT13, GGH15]. However, basically speaking, all are attacked, and most applications that use graded encoding schemes are also insecure [CHL+15, CGH+15, CFL+16, HJ16, MSZ16, ADGM16, CLLT16, CGH16]. As an exception, a few IO constructions are still standing [GMM+16, FRS16][2]. The other method is using general cryptographic primitives such as functional encryption, which enables us to generate functional keys that are tied with a certain function f. Given such a functional key, we can obtain $f(x)$ by decryption of ciphertext $\mathsf{Enc}(x)$ where x is a plaintext. Ananth and Jain [AJ15] and Bitansky and Vaikuntanathan [BV15] show how to construct IO from public-key functional encryption (PKFE). Bitansky, Nishimaki, Passelégue, and Wichs [BNPW16a] show how to construct IO from secret-key functional encryption (SKFE) and plain public-key encryption. The main purpose of our study is exploring how to construct secure IO. We follow the work aimed at constructing IO based on general cryptographic primitives. We basically focus on IO, SKFE, and PKFE for $\mathsf{P/poly}$ in this study.

Size matters. We look closer at the work of Bitansky and Vaikuntanathan [BV15] (or that of Ananth and Jain [AJ15]). They introduce the notion of succinctness for functional encryption schemes, which means the encryption-time is independent of the function-size. The succinctness of functional encryption is key to achieve IO. Precisely speaking, Bitansky and Vaikuntanathan show that a sub-exponentially secure single-key weakly succinct PKFE implies IO. "Single-key" means only one functional key is issued. We also say q-key when q functional keys are issued. "Collusion-resistant" means q is an unbounded polynomial. Weak succinctness[3] means the size of the encryption circuit is $s^{\gamma} \cdot \mathrm{poly}(\lambda, n)$ where λ is a security parameter, s is the size of f that is embedded in a functional key, n is the length of a plaintext, and γ is a constant such that $0 < \gamma < 1$.

Not only the encryption-time of functional encryption but also the size of obfuscated circuits (or the running time of the obfuscator) is also an important measure. Lin, Pass, Seth, and Telang [LPST16] introduced the notion of exponentially-efficient indistinguishability obfuscator (XIO), which is a weaker variant of IO. XIO is almost the same as IO, but the size of the obfuscated circuits is $\mathrm{poly}(\lambda, |C|) \cdot 2^{\gamma n}$ where λ is a security parameter, C is a circuit to be obfuscated, n is the length of input for C, and a compression factor γ is some value such that $0 < \gamma < 1$. They prove that if we assume that there exists XIO for circuits and the learning with errors (LWE) problem is hard, then there exists IO. Moreover, if the running time of the obfuscator is $\mathrm{poly}(\lambda, |C|) \cdot 2^{\gamma n}$, then we say it is strong XIO (SXIO) [LPST16, BNPW16a]. Bitansky *et al.* show that SXIO and public-key encryption imply IO. Thus, (S)XIO is useful enough to achieve IO. In this study, we discuss more applications of SXIO. In particular, we discuss significantly simple generic constructions of IO and weakly succinct functional encryption by using SXIO.

1.2 Our Results

We propose simple generic constructions of single-key weakly succinct functional encryption by using SXIO. More specifically, we prove the following theorems:

[1] We borrow this term from the talk by Amit Sahai at MIT, "State of the IO: Where we stand in the quest for secure obfuscation" http://toc.csail.mit.edu/node/981

[2] Martin Albrecht and Alex Davidson maintain the status of graded encoding schemes and IO constructions at http://malb.io/are-graded-encoding-schemes-broken-yet.html.

[3] In some papers, the term weak "compactness" is used for this property, but we use the term by Bitansky and Vaikuntanathan [BV15] in this study.

Main theorem 1 (informal): A single-key weakly succinct SKFE is implied by one-way function and SXIO with a compression factor that is only *slightly smaller than* 1.

Main theorem 2 (informal): A single-key weakly succinct PKFE is implied by public-key encryption and SXIO with a *sufficiently small* compression factor.

Main theorem 3 (informal): A single-key weakly succinct PKFE is implied by identity-based encryption and SXIO with a compression factor that is only *slightly smaller than* 1.

We highlight that all these theorems incur *only polynomial* security loss. These main theorems imply many new facts. Before we explain the implications of the first theorem, we explain the second and third ones since they are related to IO.

Implication of second and third theorems. When the second or third theorems are combined with the result by Bitansky and Vaikuntanathan, which proves that sub-exponentially secure single-key weakly succinct PKFE implies IO [BV15], we obtain two new constructions of IO. One is based on public-key encryption and collusion-resistant (non-succinct) SKFE since collusion-resistant (non-succinct) SKFE implies SXIO with an arbitrarily small constant compression factor (security loss is only polynomial) [BNPW16a]. The other is based on identity-based encryption and single-key weakly succinct SKFE since single-key weakly succinct SKFE implies SXIO with a compression factor that is slightly smaller than 1 [BNPW16a]. Regarding the second theorem, Bitansky *et al.* already proved the same theorem. However, our single-key weakly succinct PKFE scheme is significantly simpler than that of Bitansky *et al.* [BNPW16a] and easy to understand since we do not need a complicated tool, called decomposable garbled circuit [BNPW16a]. See the discussion in the next paragraph for details. The third theorem is new since it is not known whether single-key weakly succinct SKFE implies collusion-resistant SKFE (though it is known that single-key weakly succinct *PKFE* implies collusion-resistant PKFE with polynomial security loss [GS16, LM16]). As well as one-way function and public-key encryption, identity-based encryption is also a standard cryptographic primitive since there are many instantiations of identity-based encryption based on widely believed number theoretic assumptions and lattice assumptions. Thus, all one needs is to *slightly compress* the brute-force canonicalizer that outputs an entire truth table of a circuit to be obfuscated to construct IO (or single-key weakly succinct PKFE).

Differences from construction by Bitansky *et al.* As we explain above, our second theorem is the same as that of Bitansky *et al.* [BNPW16a]. We summarize differences between their single-key weakly succinct PKFE scheme and ours in Table 1. Decomposable garbled circuit is an extension of Yao's garbled circuit [Yao86] proposed by Bitansky *et al.* [BNPW16a]. Decomposable garbled circuit incurs $2^{O(d)}$ security loss where d is the depth of circuits [BNPW16a].

Second theorem: Our second theorem does not require decomposable garbled circuit and avoids $2^{O(d)}$ security loss. On the other hand, Bitansky *et al*'s requires decomposable garbled circuit to directly achieve weak succinctness. Our unified design strategy significantly simplifies a construction of single-key weakly succinct PKFE based on SXIO. In fact, our second theorem uses decomposable randomized encoding [IK00, AIK06], but decomposable randomized encoding is a simple tool and *does not incur* $2^{O(d)}$ *security loss*.[4] This fact gives us an advantage over the construction of Bitansky *et al.* as follows. As we explain above, there are transformations from a single-key weakly succinct PKFE scheme to a collusion-resistant one with polynomial security loss [GS16, LM16]. Thus, if we construct a single-key weakly succinct PKFE scheme from collusion-resistant SKFE and public-key encryption schemes with polynomial security loss, we can obtain a collusion-resistant PKFE scheme with polynomial security loss from the same ingredients. However, the construction of Bitansky *et al.* need a weak pseudo-random function (PRF) in NC^1 to do that due to $2^{O(d)}$ security loss [BNPW16a, Section 5.3]. Such a PRF exists under the decisional Diffie-Hellman or LWE assumption. *Ours does not need a weak PRF in* NC^1.

Third theorem: Our third theorem is new since it uses a single-key weakly succinct SKFE scheme, while Bitansky et al's uses a collusion-resistant non-succinct SKFE scheme. We can say that we relax the requirements on functional encryption to achieve IO since it is not known whether a single-key (weakly) succinct SKFE

[4] See Section 2.5 for more details on the difference between decomposable garbled circuit and decomposable randomized encoding.

scheme implies a collusion-resistant non-succinct SKFE scheme[5] though the opposite is known [AJS15]. Of course, regarding additional assumptions (public-key encryption and identity-based encryption), the existence of identity-based encryption is a stronger assumption than that of public-key encryption. However, identity-based encryption is a standard cryptographic primitive and the assumption is reasonably mild since many instantiations of identity-based encryption are known (we omit references since there are too many).

Readers who are familiar with the construction of Bitansky *et al.* might think the third theorem is easily obtained from the single-key weakly succinct PKFE scheme by Bitansky *et al.* [BNPW16a], which actually uses an identity-based encryption scheme constructed from SXIO and public-key encryption as a building block.[6] This is not the case because their construction uses an SXIO *three times in a nested manner* to construct their single-key weakly succinct PKFE scheme. They construct a single-key weakly succinct PKFE scheme for Boolean functions by using SXIO and identity-based encryption and transform it into a single-key weakly succinct PKFE scheme for non-Boolean functions by using SXIO again. This is because their construction must use decomposable garbled circuit for Boolean circuits to achieve weak succinctness (if we use decomposable garbled circuit for Boolean circuits in parallel to achieve a multi-bit output, then weak succinctness can not be achieved). Even if we replace their identity-based encryption scheme based on SXIO and public-key encryption with an assumption that there exists identity-based encryption, their construction still requires the use of SXIO *two times in a nested manner*. Thus, SXIO based on single-key weakly succinct SKFE does not work in their construction.

	ingredients for IO			compression factor	SXIO is based on
[BNPW16a]	PKE,	dGC,	γ-SXIO	sufficiently small	collusion-resistant SKFE
2nd thm.	PKE,	dRE,	γ-SXIO	sufficiently small	collusion-resistant SKFE
3rd thm.	IBE,	GC, dRE,	$\widetilde{\gamma}$-SXIO	slightly smaller than 1	1-key weakly succinct SKFE

Table 1: Difference between the construction by Bitansky *et al.* and ours. PKE, IBE, GC, dGC, and dRE denote public-key encryption, identity-based encryption, garbled circuit, decomposable garbled circuit, and decomposable randomized encoding, respectively. Let γ-SXIO denote SXIO with compression factor γ. Our second theorem is the same as that of Bitansky *et al.* , but our single-key weakly succinct PKFE scheme and proof are significantly simpler than those of Bitansky *et al.* One notable feature of our second theorem is that it avoids $2^{O(d)}$ security loss since we do not rely on decomposable garbled circuit. Third theorem is new. It is known that (decomposable) garbled circuit and randomized encoding are implied by one-way function.

Regarding difference from construction by Komargodski and Segev [KS17]. Komargodski and Segev construct an IO for circuits with inputs of poly-logarithmic length and sub-polynomial size from a quasi-polynomially secure and *collusion-resistant* SKFE scheme for $\mathsf{P/poly}$ [KS17]. They also construct a PKFE scheme for circuits with inputs of poly-logarithmic length and sub-polynomial size from a quasi-polynomially secure and *collusion-resistant* SKFE scheme for $\mathsf{P/poly}$ and sub-exponentially secure one-way function. We highlight two differences between their and ours.

Supported circuits: The IO and PKFE scheme by Komargodski and Segev support circuits with *inputs of poly-logarithmic length and sub-polynomial size* while ours supports any polynomial size circuits (i.e., $\mathsf{P/poly}$). Moreover, as we explained in the paragraph on the difference from Bitansky *et al*'s constructions, we obtain a collusion-resistant succinct PKFE scheme for $\mathsf{P/poly}$ from collusion-resistant SKFE for $\mathsf{P/poly}$ and public-key encryption with only polynomial security loss. Thus, their construction *does not* imply our second theorem.

Underlying assumptions: They assume *collusion-resistant* SKFE scheme as their starting point while we assume *single-key* weakly succinct SKFE scheme for the first and third theorems. Thus, their construction *does not* imply our first and third theorems. Moreover, they need sub-exponentially secure one-way function for their PKFE scheme while ours need polynomially secure plain public-key encryption. Thus, again, their construction *does not* imply our second theorem.

[5] In fact, in a concurrent study, it is proved that a single-key weakly succinct SKFE scheme implies a collusion-resistant SKFE scheme [Ano17]. However, our third theorem has an advantage over that of the concurrent study. Our single-key weakly succinct PKFE scheme from identity-based encryption and SXIO uses less intermediate tools, is simple, and can avoid $2^{O(d)}$ security loss, while the concurrent study's scheme cannot since it still relies on the construction by Bitansky *et al.* to achieve single-key weakly succinct PKFE (in turn IO).

[6] Note that our requirements on an identity-based encryption scheme is the same as theirs on their identity-based encryption scheme.

Implication of first theorem. The first theorem does not imply IO since how to construct IO without public-key primitives is not known [BNPW16a]. In fact, it it impossible to construct IO from only SKFE via fully black-box reductions in a certain model [AS15]. However, we can obtain interesting by-products from the first theorem.

By-product 1: We show that single-key weakly succinct SKFE is equivalent to one-way function and SXIO since it is known that such SKFE implies SXIO with a compression factor that is only slightly smaller than 1 [BNPW16b].

By-product 2: We show that if there exists single-key *weakly*-selective secure SKFE (see Definition 2.14) that is weakly succinct, there exists single-key *selectively* secure SKFE that is weakly succinct since it is known that we can construct $\widetilde{\gamma}$-SXIO such that $0 < \widetilde{\gamma} < 1$ from single-key *weakly*-selective secure SKFE that is weakly succinct [BNPW16b].

By-product 3: We show that if there exists one-way function and $\widetilde{\gamma}$-SXIO where $\widetilde{\gamma}$ is a constant such that $0 < \widetilde{\gamma} < 1$, then there exists γ-SXIO where γ is an arbitrarily small constant such that $0 < \gamma < 1$. This is obtained with our first theorem and the following facts. Single-key weakly succinct SKFE implies collusion-resistant SKFE [Ano17]. Collusion-resistant SKFE implies γ-SXIO [BNPW16a]. That is, we can decrease the compression factor to an arbitrarily small constant by using the power of one-way function.

By-product 4: We show that constant-arity multi-input functional encryption (MIFE)[7] [GGG+14] is equivalent to SXIO and one-way function. This result is obtained with our first theorem and the following facts. Single-key weakly succinct SKFE implies collusion-resistant SKFE [Ano17] and collusion-resistant SKFE implies constant-arity MIFE [BKS16]. Previously, it was known that constant-arity MIFE implies SXIO [BNPW16a] and polynomial-arity MIFE is equivalent to IO and one-way function [GGG+14].

By-product 5: We show that the existence of output-compact updatable randomized encoding with unbounded number of updates [ACJ16] and one-way function is equivalent to that of single-key weakly succinct SKFE. Previously, it is known that the existence of output-compact updatable randomized encoding with unbounded number of updates and *the hardness of the LWE problem* imply the existence of single-key weakly succinct SKFE [ACJ16]. It is also known that single-key weakly succinct SKFE implies output-compact updatable randomized encoding with unbounded number of updates. Thus, we replace the LWE assumption in the results by Ananth, Cohen, and Jain [ACJ16] with one-way function. We do not explain output-compact updatable randomized encoding proposed by Ananth *et al.* since it is not the purpose of this study.

1.3 Overview of Our Construction Technique

Our core schemes are q-key weakly collusion-succinct functional encryption schemes that are constructed from SXIO and an additional cryptographic primitive (one-way function, public-key encryption, or identity-based encryption). Weak collusion-succinctness means the size of the encryption circuit is *sub-linear in the number of issuable functional keys*. See Definition 2.18 for more details on succinctness. We explain our ideas to achieve q-key collusion-succinct functional encryption schemes below.

Our main idea in one sentence. Our main idea is compressing a parallelized encryption circuit by using SXIO to achieve weak collusion-succinctness.

Starting point. A naive idea to construct a q-key functional encryption scheme from a single-key non-succinct functional encryption scheme is running q single-key non-succinct functional encryption schemes in parallel where q is a polynomial fixed in advance. A master secret/public key consist of q master secret/public keys of the single-key scheme, respectively. A ciphertext consists of q ciphertexts of a plaintext x under q master secret or public keys. This achieves q-key functional encryption.[8] However, this simple-parallel scheme is apparently not weakly collusion-succinct since the size of the encryption circuit is linear in q. . Note that a single-key non-succinct functional encryption scheme is constructed from a standard cryptographic primitive (such as one-way function, public-key encryption) [SS10, GVW12].

[7] In MIFE, a functional decryption key is associated with a multi-arity function and a decryption algorithm takes multiple ciphertexts as inputs.

[8] In fact, the functional key generation algorithm takes an additional input called index and is stateful. We ignore this issue here. See Section 2 regarding this issue.

Compressing by SXIO. Our basic idea is compressing the encryption circuit of the simple-parallel scheme by using SXIO. Instead of embedding all q keys in an encryption circuit, our encryption algorithm obfuscates a circuit that generates the i-th master secret/public key of the simple-parallel scheme and uses it to generate a ciphertext under the i-th key where i is an input to the circuit. For simplicity, we consider the SKFE case. We set a pseudo-random function (PRF) key K as a master secret key. For a plaintext x, our weakly collusion-succinct encryption algorithm generates a circuit $\mathsf{E}'[K, x]$ that takes as an input an index $i \in [q]$, generates the i-th master secret key MSK_i by using the hard-wired K and the index i, and outputs a ciphertext $\mathsf{Enc}(\mathsf{MSK}_i, x)$ of the single-key scheme[9]. A ciphertext of our scheme is $\mathsf{sxi}\mathcal{O}(\mathsf{E}'[K, x])$. In $\mathsf{E}'[K, x]$, each master secret key is generated in an on-line manner by using the PRF (it is determined only by K and input i). The encryption circuit size of each $\mathsf{Enc}(\mathsf{MSK}_i, x)$ is independent of q because it is the encryption algorithm of the single-key scheme. The input space of $\mathsf{E}'[K, x]$ is $[q]$. Thus, if we apply an SXIO to $\mathsf{E}'[K, x]$, then the size of our encryption circuit is $\mathrm{poly}(\lambda, |x|, |f|) \cdot q^{\gamma}$. This achieves weak collusion-succinctness. The size depends on $|f|$, but it is not an issue since our goal at this step is not (weak) succinctness. The security is proved using the standard punctured programming technique [SW14].

We achieve a q-key weakly collusion-succinct PKFE by a similar idea to the SKFE case. Only one exception is that we need an SXIO for not only an encryption circuit but also a master public-key generation circuit to avoid embedding all q public-keys in an encryption algorithm. That is, a master public-key is an obfuscated circuit that outputs a master public-key of a single-key scheme by using a PRF key. This incurs two applications of SXIO in a nested manner (i.e., we obfuscate a circuit where another obfuscated circuit is hard-wired). Thus, a better compression factor of SXIO is required to achieve weakly collusion-succinctness in this case. Such better SXIO is implied by *collusion-resistant* (non-succinct) SKFE [BNPW16a].

Using power of identity-based encryption. To overcome the nested applications of SXIO, we directly construct a q-key weakly collusion-succinct PKFE from an SXIO, identity-based encryption, and garbled circuit. However, the main idea is the same. Our starting point is the single-key non-succinct PKFE scheme of Sahai and Seyalioglu [SS10], which is based on a public-key encryption scheme PKE. We use a universal circuit $U_x(\cdot)$ in which a plaintext x is hard-wired in and takes as an input a function f, which will be embedded in a functional key. Let $s := |f|$. The scheme of Sahai and Seyalioglu is as follows.

Setup: A master public-key consists of $2s$ public-keys of PKE, $\{\mathsf{pk}_0^j, \mathsf{pk}_1^j\}_{j\in[s]}$.
Functional Key: A functional key for f consists of s secret-keys of PKE, $\{\mathsf{sk}_{f_j}\}_{j\in[s]}$ where $f = f_1 \dots f_s$.
Encryption: A ciphertext consists of a garbled circuit of U_x and encryptions of $2s$ labels of the garbled circuit under pk_b^j.
Decryption: We obtain labels corresponding to f by using $\{\mathsf{sk}_{f_j}\}_{j\in[s]}$ and evaluate the garbled circuits.

We can replace PKE with an identity-based encryption scheme IBE by using identities in $[s] \times \{0, 1\}$. That is, $\{\mathsf{pk}_0^j, \mathsf{pk}_1^j\}_{j\in[s]}$ is aggregated into a master public-key of IBE. A functional key for f consists of secret keys for identities $(1, f_1), \dots, (s, f_s)$. In addition, encryptions of $2s$ labels consist of $2s$ ciphertexts for identities (j, b) for all $j \in [s]$ and $b \in \{0, 1\}$. We parallelize this by extending the identity space into $[q] \times [s] \times \{0, 1\}$ to achieve a q-key scheme. We need compression to achieve weakly collusion-succinctness since simple parallelization incurs the linearity in q.

Our encryption algorithm obfuscates the following circuit $\widetilde{\mathsf{E}}$ by using an SXIO. A master public-key of IBE is hard-wired in $\widetilde{\mathsf{E}}$. Given index i, $\widetilde{\mathsf{E}}$ generates a garbled circuit of $U_x(\cdot)$ with $2s$ labels and outputs the garbled circuit and encryptions of the $2s$ labels under appropriate identities.An identity consists of $(i, j, f_j) \in [q] \times [s] \times \{0, 1\}$. A ciphertext of our scheme is $\mathsf{sxi}\mathcal{O}(\widetilde{\mathsf{E}})$. Therefore, if secret keys for identities $\{(i, j, f_j)\}_{j\in[s]}$ are given as functional keys, then we can obtain labels only for f from corresponding ciphertexts of IBE output by $\mathsf{sxi}\mathcal{O}(\widetilde{\mathsf{E}})$ and compute $U_x(f) = f(x)$. A master public-key and encryption circuit of the identity-based encryption are succinct in the sense that their size is sub-linear in $|\mathcal{ID}|$ where $\mathcal{ID}$ is the identity space of IBE. That is, the size depends on $|\mathcal{ID}|^{\alpha}$ for sufficiently small constant α.[10] The garbled circuit part is independent of q. Therefore, the encryption circuit that generates an obfuscated circuit of $\widetilde{\mathsf{E}}$ is weakly collusion-succinct from the property of SXIO because the input space of $\widetilde{\mathsf{E}}$ is just $[q]$. Note that we can choose sufficiently small constant α for IBE. See Section 3.3 for more details. In fact, this PKFE construction is similar to that of Bitansky *et al.* (It originally comes from construction by Sahai and Seyalioglu [SS10]), but we do not need decomposable garbled circuit because our goal is achieving weak

[9] We ignore the issue regarding randomness of the ciphertext in this section.

[10] When we say identity-based encryption, we assume that it satisfies this type of succinctness. In fact, most identity-based encryption schemes based on number theoretic or lattice assumptions satisfy it. See Definition 2.8.

collusion-succinctness, which allows encryption circuits to polynomially depend on the size of f (our goal is *not weak succinctness* at this stage). Thus, a standard garbled circuit is sufficient for our construction. Moreover, SXIO with a bad compression factor is sufficient since we use an SXIO only once.

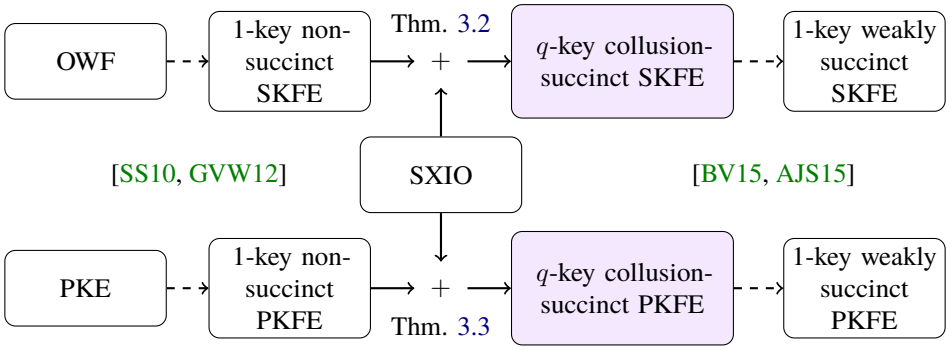

Figure 1: Illustration of our first and second theorems. Dashed lines denote known constructions. White boxes denote our ingredients or goal. Purple boxes denote our core schemes. Primitives in rounded boxes should be sub-exponentially-secure to arrive at IO. We ignore puncturable PRF in this figure since it is implied by OWF.

It is known that public-key encryption (resp. one-way function) implies single-key non-succinct PKFE (resp. SKFE) [SS10, GVW12] and bounded-key weakly collusion-succinct PKFE (resp. SKFE) implies single-key weakly succinct PKFE (resp. SKFE) [BV15, AJS15]. Thus, we can obtain single-key weakly succinct PKFE (resp. SKFE) by our weakly collusion-succinct PKFE (resp. SKFE). Figure 1 illustrates our first and second theorems.

Concurrent and Independent Work. Lin and Tessaro [LT17] proved that a collusion-resistant PKFE scheme for P/poly is constructed from any single-key PKFE scheme for P/poly (e.g., a PKFE scheme based on public-key encryption proposed by Gorbunov, Vaikuntanathan, and Wee [GVW12]) and IO for $\omega(\log \lambda)$-bit-input circuits. By combining previous results [AJS15, AJ15, BV15], their result also implies that IO for P/poly can be constructed from public-key encryption and IO for $\omega(\log \lambda)$-bit-input circuits both of which is sub-exponentially secure.

Their construction technique is similar to that of our single-key weakly succinct PKFE scheme for P/poly from public-key encryption and SXIO. We emphasize that our work is completely independent of and concurrent with theirs. One notable difference is that they use IO for $\omega(\log \lambda)$-bit-input circuits while we use SXIO for P/poly based on collusion-resistant SKFE for P/poly with polynomial security loss, that is, a special case of IO for $O(\log \lambda)$-bit-input circuits. It is not known whether IO for $\omega(\log \lambda)$-bit-input circuits is constructed from collusion-resistant SKFE for P/poly even if we allow sub-exponential security loss, though IO for $O(\mathrm{poly}(\log \lambda))$-bit-input and sub-polynomial size circuits is constructed from collusion-resistant SKFE with quasi-polynomial security loss [KS17]. Thus, our assumptions are milder than theirs to construct IO for P/poly (or single-key weakly succinct PKFE for P/poly)[11]

Organization. The main body of this paper consists of the following parts. In Section 2, we provide preliminaries and basic definitions. In Section 3, we present our constructions of weakly collusion-succinct functional encryption shcemes based on SXIO and standard cryptographic primitives. In Section 4, we provide a statement about how to transform weakly collusion-succinct functional encryption schemes into single-key weakly succinct functional encryption schemes. In Section 5, we summarize our results. Due to limited space, we put omitted definitions, constructions, statements, and proofs into the appendix.

2 Preliminaries

We now define some notations and cryptographic primitives.

[11] IO for P/poly is equivalent to single-key weakly succinct PKFE for P/poly [AJ15, BV15] or collusion-resistant PKFE for P/poly [AJS15] up to sub-exponential security loss. Moreover, single-key weakly succinct PKFE for P/poly is equivalent to collusion-resistant PKFE for P/poly up to *polynomial security loss* [GS16, LM16].

2.1 Notations and Basic Concepts

In this paper, $x \leftarrow X$ denotes selecting an element from a finite set X uniformly at random, and $y \leftarrow \mathsf{A}(x)$ denotes assigning to y the output of an probabilistic or deterministic algorithm A on an input x. When we explicitly show that A uses randomness r, we write $y \leftarrow \mathsf{A}(x; r)$. For strings x and y, $x\|y$ denotes the concatenation of x and y. Let $[\ell]$ denote the set of integers $\{1, \cdots, \ell\}$, λ denote a security parameter, and $y := z$ denote that y is set, defined, or substituted by z.

- We say that a Turing machine is probabilistic polynomial-time (PPT) if it is probabilistic and runs in polynomial time.
- We model any efficient adversary as a family of polynomial-size (we write poly-size for shorthand) circuits $\{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$. We omit the subscript λ when it is clear from the context.
- A function $f : \mathbb{N} \to \mathbb{R}$ is a negligible function if for any constant c, there exists $\lambda_0 \in \mathbb{N}$ such that for any $\lambda > \lambda_0$, $f(\lambda) < \lambda^{-c}$. We write $f(\lambda) = \mathsf{negl}(\lambda)$ to denote $f(\lambda)$ being a negligible function.
- 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 any poly-size distinguisher $\mathcal{D}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\Delta := |\Pr[\mathcal{D}(X_\lambda^{(0)}) = 1] - \Pr[\mathcal{D}(X_\lambda^{(1)}) = 1]| \leq \mathsf{negl}(\lambda).$$

  We write $\mathcal{X}^{(0)} \overset{\mathsf{c}}{\approx}_\delta \mathcal{X}^{(1)}$ to denote that the advantage Δ is bounded by δ.

2.2 Basic Cryptographic Primitives

Definition 2.1 (Pseudo-Random Function). *Let $\mathcal{PRF} := \{\mathsf{F}_K : \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2} \mid K \in \{0,1\}^\lambda\}$ be a family of polynomially computable functions, where ℓ_1 and ℓ_2 are some polynomials of λ. We say that $\mathcal{PRF}$ is a pseudo-random function (PRF) family if for any poly-size adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that*

$$\mathsf{Adv}^{\mathsf{prf}}_{\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}^{\mathsf{F}_K(\cdot)}(1^\lambda) = 1 \mid K \leftarrow \{0,1\}^\lambda] - \Pr[\mathcal{A}^{\mathsf{R}(\cdot)}(1^\lambda) = 1 \mid \mathsf{R} \leftarrow \mathcal{U}]| \leq \mathsf{negl}(\lambda),$$

where $\mathcal{U}$ is the set of all functions from $\{0,1\}^{\ell_1}$ to $\{0,1\}^{\ell_2}$. We further say that $\mathcal{PRF}$ is δ-secure, for some concrete negligible function $\delta(\cdot)$, if the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.

Puncturable PRFs, defined by Sahai and Waters [SW14], 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 2.2 (Puncturable PRF). *For sets D, R, a* puncturable PRF *consists of a tuple of algorithms $\mathcal{PPRF} = (\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Punc})$ that satisfy the following two conditions.*

Functionality preserving under puncturing: *For any polynomial size set $S \subseteq \mathsf{D}$ and any $x \in \mathsf{D} \setminus S$, it holds that*

$$\Pr[\mathsf{F}_K(x) = \mathsf{F}_{K\{S\}}(x) \mid K \leftarrow \mathsf{PRF.Gen}(1^\lambda), K\{S\} \leftarrow \mathsf{Punc}(K, S)] = 1.$$

Pseudorandom at punctured points: *For any poly-size set $S \subseteq \mathsf{D}$ with $S = \{x_1, \ldots, x_{k(\lambda)}\}$ and any poly-size distinguisher $\mathcal{A}$, there exists a negligible function negl, such that*

$$|\Pr[\mathcal{A}(\mathsf{F}_{K\{S\}}, \{\mathsf{F}_K(x_i)\}_{i\in[k]}) = 1] - \Pr[\mathcal{A}(\mathsf{F}_{K\{S\}}, U^k) = 1]| \leq \mathsf{negl}(\lambda),$$

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

The Goldwasser-Goldreich-Micali tree-based construction of PRFs [GGM84] from one-way function 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 [BW13, BGI14, KPTZ13]. Thus, we have:

Theorem 2.3 ([GGM84, BW13, BGI14, KPTZ13]). *If one-way function exists, then for any efficiently computable functions $n(\lambda)$ and $m(\lambda)$, there exists a puncturable PRF that maps n-bits to m-bits (i.e., $\mathsf{D} := \{0,1\}^{n(\lambda)}$ and $\mathsf{R} := \{0,1\}^{m(\lambda)}$).*

Definition 2.4 (Secret-Key Encryption). *A secret-key encryption scheme SKE is a two tuple $(\mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms.*

- *The encryption algorithm Enc, given a key $K \in \{0,1\}^\lambda$ and a message $m \in \mathcal{M}$, outputs a ciphertext c, where $\mathcal{M}$ is the plaintext space of SKE.*
- *The decryption algorithm D, given a key K and a ciphertext c, outputs a message $\tilde{m} \in \{\bot\} \cup \mathcal{M}$. This algorithm is deterministic.*

Correctness: *We require $\mathsf{Dec}(K, \mathsf{Enc}(K, m)) = m$ for any $m \in \mathcal{M}$ and key K.*

Definition 2.5 (One-Time Secure Secret-Key Encryption:). *A tuple of algorithms $\mathsf{SKE} = (\mathsf{Enc}, \mathsf{Dec})$ is a secure SKE scheme for $\mathcal{M}$ if it satisfies the following requirement, formalized from the experiment $\mathsf{Expt}^{\mathsf{ske}}_{\mathcal{A}}(1^\lambda, b)$ between an adversary $\mathcal{A}$ and a challenger. Below, let n be a fixed polynomial of λ.*

1. *The challenger selects a challenge bit $b \leftarrow \{0,1\}$, generates a key $K \leftarrow \{0,1\}^\lambda$, and sends 1^λ to $\mathcal{A}$.*
2. *$\mathcal{A}$ sends $(m_0, m_1) \in \mathcal{M} \times \mathcal{M}$ to the challenger. Then, the challenger returns $c \leftarrow \mathsf{Enc}(K, m_b)$.*
3. *$\mathcal{A}$ outputs $b' \in \{0,1\}$. The experiment outputs 1 if $b = b'$; otherwise 0.*

We say the SKE scheme is one-time secure if, for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\mathsf{Adv}^{\mathsf{ske}}_{\mathcal{A}} := |\Pr[\mathsf{Expt}^{\mathsf{ske}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{ske}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that SKE is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.

Definition 2.6 (Plain Public-key Encryption). *Let $\mathcal{M}$ be a message space. A public-key encryption scheme for $\mathcal{M}$ is a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ where:*

- $\mathsf{KeyGen}(1^\lambda)$ *takes as input the security parameter and outputs a public key pk and secret key sk.*
- $\mathsf{Enc}(\mathsf{pk}, m)$ *takes as input pk and a message $m \in \mathcal{M}$ and outputs a ciphertext ct.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *takes as input sk and ct, and outputs some $m' \in \mathcal{M}$, or $\bot$.*

We also require the following property:

Correctness: *For any $m \in \mathcal{M}$ and $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, we have that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$.*

We also recall the standard notion of security.

Definition 2.7 (Secure Public-key Encryption). *A tuple of algorithms $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a secure public-key encryption for $\mathcal{M}$ if it satisfies the following requirement, formalized from the experiment $\mathsf{Expt}^{\mathsf{pke}}_{\mathcal{A}}(1^\lambda, b)$ between an adversary $\mathcal{A}$ and challenger:*

1. *The challenger runs $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, 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 $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b^*)$ and sends CT^* to $\mathcal{A}$.*
4. *$\mathcal{A}$ outputs a guess b' for b. The experiment outputs 1 if $b' = b$; otherwise 0.*

We say PKE is secure if, for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$, such that

$$\mathsf{Adv}^{\mathsf{pke}}_{\mathcal{A}} := |\Pr[\mathsf{Expt}^{\mathsf{pke}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{pke}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that PKE is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.

Definition 2.8 (Succinct Identity-Based Encryption). *Let $\mathcal{M}$ be a message space and $\mathcal{ID}$ be an identity space. A succinct identity-based encryption scheme with α-compression for $\mathcal{M}$ and $\mathcal{ID}$ is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Key}, \mathsf{Enc}, \mathsf{Dec})$ *where:*

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

We require the following properties:

Correctness: *For any* $m \in \mathcal{M}$*, any* $\mathsf{id} \in \mathcal{ID}$*,* $(\mathsf{MSK}, \mathsf{MPK}) \leftarrow \mathsf{Setup}(1^\lambda)$*, and* $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{KG}(\mathsf{MSK}, \mathsf{id})$*, we have that* $\mathsf{Dec}(\mathsf{sk}_{\mathsf{id}}, \mathsf{Enc}(\mathsf{MPK}, \mathsf{id}, m)) = m$.

Succinctness: *For any security parameter $\lambda \in \mathbb{N}$ and identity space $\mathcal{ID}$, the size of the encryption circuit* Enc *for $\mathcal{ID}$ and messages of size ℓ is at most $|\mathcal{ID}|^\alpha \mathrm{poly}(\lambda, \ell)$ where $0 < \alpha < 1$.*

The efficiency property is not explicitly stated in many papers on identity-based encryption scheme since identity-based encryption schemes based on number theoretic or lattice assumptions satisfy the efficiency (in fact, the size of most schemes is bounded by $\mathrm{poly}(\lambda, \ell, \log |\mathcal{ID}|)$). This was defined by Bitansky *et al.* [BNPW16a].

In this study, we considered the following security, which is a weaker variant of standard selective-security as Bitansky *et al.* [BNPW16a].

Definition 2.9 (Selectively-Secure Identity-Based Encryption). *A tuple of algorithms* $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{Key}, \mathsf{Enc}, \mathsf{Dec})$ *is a* selectively-secure *identity-based encryption scheme for $\mathcal{M}$ and $\mathcal{ID}$ if it satisfies the following requirement, formalized from the experiment* $\mathsf{Expt}^{\mathsf{ibe}}_{\mathcal{A}}(1^\lambda, b)$ *between an adversary $\mathcal{A}$ and a challenger:*

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

We say the identity-based encryption scheme is selectively-secure if, for any PPT $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$*, such that*

$$\mathsf{Adv}^{\mathsf{ibe}}_{\mathcal{A}} := |\Pr[\mathsf{Expt}^{\mathsf{ibe}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{ibe}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that IBE *is δ-selectively secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

2.3 Functional Encryption

In this subsection we review the different notions of functional encryption.

Secret-Key Functional Encryption

We introduce the syntax of an index based variant SKFE scheme that we call an *index based SKFE (iSKFE)* scheme. "Index based" means that, to generate the i-th functional decryption key, we need to feed an index i to a key generation algorithm. For a single-key scheme, an iSKFE scheme is just a standard SKFE scheme in which the key generation algorithm does not take an index as an input since the index is always fixed to 1.

Definition 2.10 (Index Based Secret-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, $\mathcal{I} := \{\mathcal{I}_\lambda\}_{\lambda\in\mathbb{N}}$ an index space, and $\mathcal{F} := \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ a class of functions $f : \mathcal{M} \to \mathcal{Y}$. An iSKFE scheme for $\mathcal{M}, \mathcal{Y}, \mathcal{I}$, and $\mathcal{F}$ is a tuple of algorithms $\mathsf{SKFE} = (\mathsf{Setup}, \mathsf{iKG}, \mathsf{Enc}, \mathsf{Dec})$ where:*

- $\mathsf{Setup}(1^\lambda)$ *takes as input the security parameter and outputs a master secret key* MSK.
- $\mathsf{iKG}(\mathsf{MSK}, f, i)$ *takes as input* MSK*, a function $f \in \mathcal{F}$, and an index $i \in \mathcal{I}$, and outputs a secret key sk_f for f.*
- $\mathsf{Enc}(\mathsf{MSK}, x)$ *takes as input* MSK *and a message $x \in \mathcal{M}$ and outputs a ciphertext c.*
- $\mathsf{Dec}(\mathsf{sk}_f, c)$ *takes as input sk_f for $f \in \mathcal{F}$ and c and outputs $y \in \mathcal{Y}$, or $\perp$.*

Correctness: *We require* $\mathsf{Dec}(\mathsf{iKG}(\mathsf{MSK}, f, i), \mathsf{Enc}(\mathsf{MSK}, x)) = f(x)$ *for any $x \in \mathcal{M}$, $f \in \mathcal{F}$, $i \in \mathcal{I}$, and* $\mathsf{MSK} \leftarrow \mathsf{Setup}(1^\lambda)$.

Next, we introduce selective-message message privacy [BS15].

Definition 2.11 (Selective-Message Message Privacy). *Let* SKFE *be an SKFE scheme whose message space, function space, and index space are $\mathcal{M}$, $\mathcal{F}$, and $\mathcal{I}$, respectively. We define the selective-message message privacy experiment $\mathsf{Exp}^{\mathsf{sm\text{-}fp}}_{\mathcal{A}}(1^\lambda, b)$ between an adversary $\mathcal{A}$ and a challenger as follows.*

1. *$\mathcal{A}$ is given 1^λ and sends $(x_0^{(1)}, x_1^{(1)}), \cdots, (x_0^{(q_m)}, x_1^{(q_m)})$ to the challenger, where q_m is a polynomial of λ.*
2. *The challenger chooses* $\mathsf{MSK} \leftarrow \mathsf{Setup}(1^\lambda)$ *and a challenge bit $b \leftarrow \{0, 1\}$.*
3. *The challenger generates $\mathsf{CT}^{(j)} \leftarrow \mathsf{Enc}(\mathsf{MSK}, x_b^{(j)})$ for $j \in [q_m]$ and sends them to $\mathcal{A}$.*
4. *$\mathcal{A}$ is allowed to make arbitrary $|\mathcal{I}|$ function queries. For the ℓ-th key query $(f_\ell, i_\ell) \in \mathcal{F} \times \mathcal{I}$ from $\mathcal{A}$, the challenger generates $sk_{f_\ell} \leftarrow \mathsf{iKG}(\mathsf{MSK}, f_\ell, i_\ell)$ and returns sk_{f_ℓ} to $\mathcal{A}$. If $\mathcal{A}$ sends the same index twice, that is, there are queries (f_{ℓ_1}, i_{ℓ_1}) and (f_{ℓ_2}, i_{ℓ_2}) where $i_{\ell_1} = i_{\ell_2}$ and $\ell_1 < \ell_2$, then the challenger ignores (f_{ℓ_2}, i_{ℓ_2}). W.l.o.g, we can consider $i_\ell = \ell$.*
5. *$\mathcal{A}$ outputs $b' \in \{0, 1\}$. The experiment output 1 if $b = b'$ and $f_\ell(x_0^{(j)}) = f_\ell(x_1^{(j)})$ for all $j \in [q_m]$ and $\ell \in [q_k]$, where q_k is the number of key queries made by $\mathcal{A}$; otherwise $\perp$.*

We say that SKFE *is (q_k, q_m)-selective-message message private (or selectively secure for short) if for any poly-size adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\mathsf{Adv}^{\mathsf{sm\text{-}mp}}_{\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{sm\text{-}mp}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{sm\text{-}mp}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that SKFE *is (q_k, q_m, δ)-selective-message message private, for some concrete negligible function $\delta(\cdot)$, if for any poly-size adversary $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Remark 2.12 (Regarding the number of queries). In this study, q_m in SKFE is basically an unbounded polynomial (i.e., not fixed in advance). Thus, we omit q_m and write (q_k, δ)-selective-message message private when q_m is an unbounded polynomial. Let FE be a functional encryption scheme. If q_k is an unbounded polynomial, then we say FE is a *collusion-resistant* functional encryption. If q_k is a bounded polynomial (i.e., fixed in advance), then we say FE is a *bounded collusion-resistant* functional encryption. If $q_k = 1$, we say FE is a *single-key* functional encryption. In this study, our constructions are bounded collusion-resistant. Thus, we can consider $|\mathcal{I}| = q_k$.

Remark 2.13 (Regarding an index for algorithm iKG). Our definitions of functional encryptions slightly deviates from the standard ones (ex. the definition by Ananth and Jain [AJ15] or Brakerski and Segev [BS15]). Our key generation algorithm takes not only a master secret key and a function but also an index, which is used to bound the number of functional key generations. This index should be different for each functional key generation. One might think this is a limitation, but this is not the case in this study because our goal is constructing IO and our functional encryption schemes are just intermediate tools for doing this. In particular, for a single-key scheme, $|\mathcal{I}| = 1$ and we do not need such an index. In fact, such an index have been introduced by Li and Micciancio in the context of PKFE [LM16].

Variants of Security.

Definition 2.14 (Weakly Selective-Message Message Privacy). *In* $\mathsf{Exp}^{\mathsf{sm\text{-}mp}}_{\mathcal{A}}(1,b)$*, if* $\mathcal{A}$ *must submit not only messages* $(x_0^{(1)}, x_1^{(1)}), \cdots, (x_0^{(q)}, x_1^{(q)})$ *but also functions* $(f_1, \ldots, f_{q_k})$ *to the challenger at the beginning of the experiment, then the modified experiment is defined as* $\mathsf{Exp}^{\mathsf{sm}^*\text{-}\mathsf{mp}}_{\mathcal{A}}(1,b)$ *and an SKFE is weakly selective-message message private (or weakly-selective secure for short) if* $\mathsf{Adv}^{\mathsf{sm}^*\text{-}\mathsf{mp}}_{\mathcal{A}}(\lambda)$ *(similarly defined) is negligible.*

Public-Key Functional Encryption

Definition 2.15 (Index Based 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,* $\mathcal{I} := \{\mathcal{I}_\lambda\}_{\lambda\in\mathbb{N}}$ *an index space, and* $\mathcal{F} := \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ *a class of functions* $f : \mathcal{M} \to \mathcal{Y}$*. An* index based PKFE *(iPKFE) scheme for* $\mathcal{M}, \mathcal{Y}, \mathcal{I}$*, and* $\mathcal{F}$ *is a tuple of algorithms* $\mathsf{PKFE} = (\mathsf{Setup}, \mathsf{iKG}, \mathsf{Enc}, \mathsf{Dec})$ *where:*

- $\mathsf{Setup}(1^\lambda)$ *takes as input the security parameter and outputs a master secret key* MSK *and master public key* MPK*.*
- $\mathsf{iKG}(\mathsf{msk}, f, i)$ *takes as input* MPK*, a function* $f \in \mathcal{F}$*, and an index* $i \in \mathcal{I}$*. It outputs a secret key* sk_f *for* f*.*
- $\mathsf{Enc}(\mathsf{mpk}, m)$ *takes as input* MPK *and a message* $m \in \mathcal{M}$*, and outputs a ciphertext* c*.*
- $\mathsf{Dec}(\mathsf{sk}_f, c)$ *takes as input* sk_f *for* $f \in \mathcal{F}$ *and* c*, and outputs* $y \in \mathcal{Y}$*, or* $\perp$*.*

Correctness: *For any* $m \in \mathcal{M}$*,* $i \in \mathcal{I}$*,* $f \in \mathcal{F}$*, and* $(\mathsf{MSK}, \mathsf{MPK}) \leftarrow \mathsf{Setup}(1^\lambda)$ *we have that* $\mathsf{Dec}(\mathsf{iKG}(\mathsf{MSK}, f, i), \mathsf{Enc}(\mathsf{MPK}, m)) = f(m)$*.*

Definition 2.16 (Selectively-Secure PKFE). *We say that a tuple of algorithms* $\mathsf{iPKFE} = (\mathsf{Setup}, \mathsf{iKey}, \mathsf{Enc}, \mathsf{Dec})$ *is a* selectively-secure *PKFE scheme for* $\mathcal{M}, \mathcal{Y}, \mathcal{I}$*, and* $\mathcal{F}$*, if it satisfies the following requirement, formalized from the experiment* $\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, b)$ *between an adversary* $\mathcal{A}$ *and challenger:*

1. $\mathcal{A}$ *submits a message pair* $x_0^*, x_1^* \in \mathcal{M}$ *to the challenger.*
2. *The challenger runs* $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and generates a ciphertext* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_b^*)$*. The challenger gives* $(\mathsf{mpk}, \mathsf{ct}^*)$ *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *is allowed to make arbitrary* $|\mathcal{I}| = q_k$ *function queries, where it sends a function and index* $(f_\ell, i_\ell) \in \mathcal{F} \times \mathcal{I}$ *to the challenger. The challenger checks that* $f_\ell(x_0^*) = f_\ell(x_1^*)$*. If the check fails, then the challenger aborts. Else if there are queries* $(f_{\ell'}, i_{\ell'})$ *and* (f_ℓ, i_ℓ) *where* $i_{\ell'} = i_\ell$ *and* $\ell' < \ell$*, then the challenger ignores* (f_ℓ, i_ℓ)*. Otherwise, the challenger responds with* $\mathsf{sk}_{f_\ell} \leftarrow \mathsf{iKG}(\mathsf{msk}, f_\ell, i_\ell)$ *for the* ℓ*-th query* f_ℓ*. W.l.o.g, we can consider* $i_\ell = \ell$*.*
4. $\mathcal{A}$ *outputs a guess* b' *for* b*.*
5. *The experiment outputs* 1 *if* $b = b'$*; otherwise* 0*.*

We say that a PKFE scheme is selectively-secure if, for any poly-size $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\mathsf{Adv}^{\mathsf{sel}}_{\mathcal{A}}(\lambda) := |\Pr[\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that iPKFE *is* (q_k, δ)*-selectively secure, for some concrete negligible function* $\delta(\cdot)$*, if for any poly-size* $\mathcal{A}$ *the above advantage is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Definition 2.17 (Weakly-Selective Secure [GS16]). *We say that a tuple of algorithms* $\mathsf{iPKFE} = (\mathsf{Setup}, \mathsf{iKey}, \mathsf{Enc}, \mathsf{Dec})$ *is a* weakly selectively-secure *iPKFE scheme for* $\mathcal{M}, \mathcal{Y}, \mathcal{I}$*, and* $\mathcal{F}$*, if it satisfies the following requirement, formalized from the experiment* $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, b)$ *between an adversary* $\mathcal{A}$ *and challenger:*

1. $\mathcal{A}$ *submits a message pair* $x_0^*, x_1^* \in \mathcal{M}$*, functions* $(f_1, \ldots, f_{q_k}) \in \mathcal{F}^{q_k}$*, and indices* $(i_1, \ldots, i_{q_k}) \in \mathcal{I}^{q_k}$ *to the challenger.*
2. *If there exists* i_ℓ *and* $i_{\ell'}$ *such that* $i_\ell = i_{\ell'}$ *and* $\ell' < \ell$*, then the challenger ignores* f_ℓ *at the next stage.*

3. *The challenger runs* $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda)$*, generates ciphertext* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_b^*)$ *and secret keys* $\mathsf{sk}_{f_\ell} \leftarrow \mathsf{Key}(\mathsf{msk}, f_\ell, i_\ell)$ *for all* $\ell \in [q_k]$ *and* $i_\ell \in \mathcal{I}$*. The challenger gives* $(\mathsf{mpk}, \mathsf{ct}^*, \mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_{q_k}})$ *to* $\mathcal{A}$*.*

4. $\mathcal{A}$ *outputs a guess* b' *for* b*.*

5. *The experiment outputs* 1 *if* $b = b'$ *and* $f_i(x_0^*) = f_i(x_1^*)$ *for all* $i \in [q_k]$*; otherwise* $\perp$*.*

We say that a PKFE scheme is weakly-selective secure if, for any poly-size adversary $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\mathsf{Adv}^{\mathsf{sel}^*}_{\mathcal{A}}(\lambda) := |\Pr[\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that iPKFE *is* (q_k, δ)*-weakly-selective secure, for some concrete negligible function* $\delta(\cdot)$*, if for any poly-size* $\mathcal{A}$ *the above advantage is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Next, we introduce notions regarding efficiency, called succinctness for functional encryption schemes.

Definition 2.18 (Succinctness of Functional Encryption [BV15]). *For a class of functions* $\mathcal{F} = \{\mathcal{F}_\lambda\}$ *over message domain* $\mathcal{M} = \{\mathcal{M}_\lambda\}$*, we let:*

- $n(\lambda)$ *be the input length of the functions in* $\mathcal{F}$*,*
- $s(\lambda) = \max_{f \in \mathcal{F}_\lambda} |f|$ *be a bound on the circuit size of functions in* $\mathcal{F}_\lambda$*,*
- $d(\lambda) = \max_{f \in \mathcal{F}_\lambda} \mathsf{depth}(f)$ *be a bound on the depth, and*

a functional encryption scheme is

- succinct *if the size of the encryption circuit is bounded by* $\mathrm{poly}(n, \lambda, \log s)$*, where* poly *is a fixed polynomial.*
- weakly succinct *if the size of the encryption circuit is bounded by* $s^\gamma \cdot \mathrm{poly}(n, \lambda)$*, where* poly *is a fixed polynomial, and* $\gamma < 1$ *is a constant. We call* γ *the compression factor.*
- weakly collusion-succinct *if the size of the encryption circuit is bounded by* $q^\gamma \cdot \mathrm{poly}(n, \lambda, s)$*, where* q *is the upper bound of issuable functional keys in bounded-key schemes,* poly *is a fixed polynomial, and* $\gamma < 1$ *is a constant. We call* γ *the compression factor.*

The following theorem by Bitansky and Vaikuntanathan [BV15, Section III] states that one can construct IO from any single-key weakly succinct PKFE.

Theorem 2.19 ([BV15]). *If there exists a single-key sub-exponentially weakly-selective secure weakly succinct PKFE scheme for* P/poly*, then there exists an indistinguishability obfuscator for* P/poly*.*

2.4 Garbling Scheme

Definition 2.20 (Garbling Scheme). *Let* $\{\mathcal{C}_n\}_{n \in \mathbb{N}}$ *be a family of circuits in which each circuit in* $\mathcal{C}_n$ *takes* n *bit inputs. A circuit garbling scheme* GC *consists of two algorithms* $(\mathsf{Grbl}, \mathsf{Eval})$*.*

$\mathsf{Grbl}(1^\lambda, C)$ *takes as inputs a security parameter* 1^λ *and a circuit* $C \in \mathcal{C}_n$ *and outputs a garbled circuit* $\widetilde{C}$*, together with* $2n$ *wire keys (a.k.a labels)* $\{w_{i,\alpha}\}_{i \in [n], \alpha \in \{0,1\}}$*.*

$\mathsf{Eval}(\widetilde{C}, \{w_{i,x_i}\}_{i \in [n]})$ *takes as inputs a garbled circuit* $\widetilde{C}$ *and* n *wire keys* $\{w_{x_i}\}_{i \in [n]}$ *where* $x_i \in \{0,1\}$ *and outputs* y*.*

A garbling scheme is required to satisfy the following properties.

Correctness: *It holds* $\mathsf{Eval}(\widetilde{C}, \{w_{i,x_i}\}_{i \in [n]}) = C(x)$ *for every* $n \in \mathbb{N}$*,* $x \in \{0,1\}^n$*, where* $(\widetilde{C}, \{w_{i,\alpha}\}_{i \in [n], \alpha \in \{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, C)$*.*

Security: *Let* $\mathsf{GC.Sim}$ *be a PPT simulator. We define the following experiments* $\mathsf{Expt}^{\mathsf{gc}}_{\mathcal{A}}(1^\lambda, b)$ *between a challenger and an adversary* $\mathcal{A}$ *as follows.*

1. *The challenger chooses a bit* $b \leftarrow \{0,1\}$ *and sends security parameter* 1^λ *to* $\mathcal{A}$*.*

2. *$\mathcal{A}$ sends a circuit $C \in \mathcal{C}_n$ and an input $x \in \{0,1\}^n$ to the challenger.*
3. *If $b = 0$, the challenger computes $(\widetilde{C}, \{w_{i,\alpha}\}_{i\in[n],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, C)$ and returns $(\widetilde{C}, \{w_{i,x_i}\}_{i\in[n]})$ to $\mathcal{A}$. Otherwise, the challenger returns $(\widetilde{C}, \{w_{x_i}\}_{i\in[n]}) \leftarrow \mathsf{GC.Sim}(1^\lambda, 1^{|C|}, C(x))$.*
4. *$\mathcal{A}$ outputs $b' \in \{0,1\}$. The experiment outputs 1 if $b = b'$;otherwise 0.*

We say that GC *is secure if there exists a simulator* GC.Sim*, for any poly-size $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$\mathsf{Adv}^{\mathsf{gc}}_{\mathcal{A},\mathsf{GC.Sim}}(\lambda) = |\Pr[\mathsf{Expt}^{\mathsf{gc}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{gc}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that GC *is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size adversary $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Theorem 2.21 ([Yao86]). *If there exists one-way function, there exists a secure garbling scheme for poly-size circuits.*

2.5 Decomposable Randomized Encoding

Definition 2.22 (Decomposable Randomized Encoding). *Let $c \geq 1$ be an integer constant. A c-local decomposable randomized encoding scheme* RE *for a function $f : \{0,1\}^n \to \{0,1\}^m$ consists of two polynomial-time algorithms* $(\mathsf{RE.E}, \mathsf{RE.D})$.

$\mathsf{RE.E}(1^\lambda, f, x)$ *takes as inputs the security parameter 1^λ, a function f, and an input x for f, chooses randomness r, and outputs an encoding $\widehat{f}(x; r)$ where $\widehat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^\mu$.*

$\mathsf{RE.D}(\widehat{f}(x; r))$ *takes as inputs $\widehat{f}(x; r)$ and outputs $f(x)$.*

A randomized encoding scheme satisfies the following properties. Let $s_{\widehat{f}}$ (resp. s_f) denote the size of the circuit computing $\widehat{f}$ (resp. f).

Correctness: *For any λ, f, and x, it holds that $\Pr[f(x) = \mathsf{RE.D}(\mathsf{RE.E}(1^\lambda, f, x))] = 1$.*

Decomposability: *Computation of $\widehat{f}$ can be decomposed into computation of μ functions. That is, $\widehat{f}(x; r) = (\widehat{f}_1(x; r), \cdots, \widehat{f}_\mu(x; r))$, where each $\widehat{f}_i$ depends on at most a single bit of x and c bits of r. We write $\widehat{f}(x; r) = (\widehat{f}_1(x; r_{S_1}), \cdots, \widehat{f}_\mu(x; r_{S_\mu}))$, where S_i denotes the subset of bits of r that $\widehat{f}_i$ depends on. Parameters ρ and μ are bounded by $s_f \cdot \mathrm{poly}(\lambda, n)$.*

Semantic Security: *Let* RE.Sim *be a PPT simulator. We define the following experiments $\mathsf{Expt}^{\mathsf{dre}}_{\mathcal{A}}(1^\lambda, b)$ between a challenger and an adversary $\mathcal{A}$ as follows.*

1. *The challenger chooses a bit $b \leftarrow \{0,1\}$ and sends security parameter 1^λ to $\mathcal{A}$.*
2. *$\mathcal{A}$ sends a function f and input $x \in \{0,1\}^n$ to the challenger.*
3. *If $b = 0$, the challenger computes $\left\{\widehat{f}_i(x; r)\right\}_{i=1}^{\mu} \leftarrow \mathsf{RE.E}(1^\lambda, f, x)$ and returns them to $\mathcal{A}$. Otherwise, the challenger returns $\left\{\widehat{f}_i(x; r)\right\}_{i=1}^{\mu} \leftarrow \mathsf{RE.Sim}(1^\lambda, 1^{|f|}, f(x))$.*
4. *$\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. The experiment outputs 1 if b'; otherwise 0.*

We say that RE *is semantically secure if there exists a simulator* RE.Sim*, for any poly-size adversary $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$*, such that*

$$|\Pr[\mathsf{Expt}^{\mathsf{dre}}_{\mathcal{A}}(1^\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{dre}}_{\mathcal{A}}(1^\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

We further say that RE *is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Theorem 2.23 ([Yao86, AIK06]). *If there exists one-way function, there exists a semantically secure decomposable randomized encoding for poly-size circuits.*

Difference between decomposable randomized encoding and decomposable garbled circuit. One might think decomposable randomized encoding is also a complicated tool since randomized encoding is similar notion to garbled circuit and we explain that decomposable garbled circuit is a complicated tool in Section 1. In fact, both are basically slight extensions of Yao's garbled circuit. However, decomposable randomized encoding is a simple tool and does not incur $2^{O(d)}$ security loss while decomposable garbled circuit does. The reason decomposable garbled circuit is complicated is that it is customized to be an IO-friendly (or SXIO-friendly) tool [BNPW16a]. We use neither IO nor SXIO when we use decomposable randomized encoding. Thus, we do not need an IO-friendly (or SXIO-friendly) tool for our purpose. See the paper by Bitansky *et al.* for details of decomposable garbled circuit [BNPW16a].

2.6 Indistinguishability Obfuscation

Definition 2.24 (Indistinguishability Obfuscator). *A PPT algorithm $i\mathcal{O}$ is an IO for a circuit class $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if it satisfies the following two conditions.*

Functionality: *For any security parameter $\lambda \in \mathbb{N}$, $C \in \mathcal{C}_\lambda$, and input x, we have that*

$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(C)] = 1 \ .$$

Indistinguishability: *For any poly-size distinguisher D, there exists $\mathsf{negl}(\cdot)$ such that the following holds: For any pair of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ such that for any input x, $C_0(x) = C_1(x)$ and $|C_0| = |C_1|$,*

$$|\Pr[\mathsf{D}(i\mathcal{O}(C_0)) = 1] - \Pr[\mathsf{D}(i\mathcal{O}(C_1)) = 1]| \leq \mathsf{negl}(\lambda) \ .$$

We further say that $i\mathcal{O}$ is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any poly-size distinguisher the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.

2.7 Strong Exponentially-Efficient Indistinguishability Obfuscation

Definition 2.25 (Strong Exponentially-Efficient Indistinguishability Obfuscation). *Let $\gamma < 1$ be a constant. An algorithm $\mathsf{sxi}\mathcal{O}$ is a γ-compressing SXIO for a circuit class $\{\mathcal{C}\}_{\lambda\in\mathbb{N}}$ if it satisfies the functionality and indistinguishability in Definition 2.24 and the following efficiency requirement:*

Non-trivial time efficiency *We require that the running time of $\mathsf{sxi}\mathcal{O}$ on input $(1^\lambda, C)$ is at most $2^{n\gamma} \cdot \mathrm{poly}(\lambda, |C|)$ for any $\lambda \in \mathbb{N}$ and any circuit $C \in \{C_\lambda\}_{\lambda\in\mathbb{N}}$ with input length n.*

3 Collusion-Succinct Functional Encryption from SXIO

In our bounded-key weakly collusion-succinct SKFE and PKFE schemes, we use single-key non-succinct SKFE and PKFE schemes that are implied from one-way function and public-key encryption, respectively.

Theorem 3.1 ([GVW12]). *If there exists a δ-secure one-way function, then there exists a $(1, \delta)$-selectively secure and non-succinct SKFE scheme for $\mathsf{P/poly}$. If there exists a δ-secure public-key encryption, then there exists a $(1, \delta)$-selectively-secure and non-succinct PKFE scheme for $\mathsf{P/poly}$.*

Throughout this paper, let n and s be the length of a message x and size of a function f of a functional encryption scheme, respectively as in Definition 2.18.

3.1 Collusion-Succinct SKFE from SXIO and One-Way Function

In this section, we discuss how to construct a bounded-key collusion-succinct SKFE scheme from SXIO and one-way function.

Our Construction. The construction of an iSKFE scheme qFE from a single-key SKFE and SXIO is as follows. Let $\mathsf{1FE} = (\mathsf{1FE.Setup}, \mathsf{1FE.KG}, \mathsf{1FE.Enc}, \mathsf{1FE.Dec})$ be a single-key non-succinct SKFE scheme, $(\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Punc})$ a puncturable PRF, and $\mathsf{sxi}\mathcal{O}$ a $\widetilde{\gamma}$-compressing SXIO.

$\mathsf{qFE.Setup}(1^\lambda)$:

- Generate $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- Return $\widehat{\mathsf{MSK}} \leftarrow K$.

$\mathsf{qFE.iKG}(\widehat{\mathsf{MSK}}, f, i)$:

- Parse $K \leftarrow \widehat{\mathsf{MSK}}$.
- Compute $r_i \leftarrow \mathsf{F}_K(i)$ and $\mathsf{MSK}_i \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$.
- Compute $sk_f^i \leftarrow \mathsf{1FE.KG}(\mathsf{MSK}_i, f)$.
- Return $\widehat{\mathsf{sk}}_f \leftarrow (i, sk_f^i)$.

$\mathsf{qFE.Enc}(\widehat{\mathsf{MSK}}, x)$:

- Parse $K \leftarrow \widehat{\mathsf{MSK}}$.
- Generate $K' \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and $\mathsf{E}_{\mathsf{1FE}}[K, K', x]$ defined in Figure 2.
- Return $\widehat{\mathsf{CT}} \leftarrow \mathsf{sxi}\mathcal{O}(\mathsf{E}_{\mathsf{1FE}}[K, K', x])$.

$\mathsf{qFE.Dec}(\widehat{\mathsf{sk}}_f, \widehat{\mathsf{CT}})$:

- Parse $(i, sk_f^i) \leftarrow \widehat{\mathsf{sk}}_f$.
- Compute $\mathsf{CT}_i \leftarrow \widehat{\mathsf{CT}}(i)$.
- Return $y \leftarrow \mathsf{1FE.Dec}(sk_f^i, \mathsf{CT}_i)$.

Encryption Circuit $\mathsf{E}_{\mathsf{1FE}}[K, K', x](i)$

Hardwired: puncturable PRF key K, K', and a message x.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad} := \mathsf{pad}(\lambda, n, s, q)$, which is determined in analysis.

1. Compute $r_i \leftarrow \mathsf{F}_K(i)$ and $r'_i \leftarrow \mathsf{F}_{K'}(i)$.
2. Compute $\mathsf{MSK}_i \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$.
3. Output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, m; r'_i)$.

Figure 2: Description of $\mathsf{E}_{\mathsf{1FE}}[K, K', x]$

Theorem 3.2. *If there exists non-succinct $(1, \delta)$-selective-message message private SKFE for $\mathsf{P/poly}$ and δ-secure $\widetilde{\gamma}$-compressing SXIO for $\mathsf{P/poly}$ where $0 < \widetilde{\gamma} < 1$ ($\widetilde{\gamma}$ might be close to 1), then there exists weakly collusion-succinct (q, δ)-selective-message message private SKFE for $\mathsf{P/poly}$ with compression factor γ' such that $0 < \widetilde{\gamma} < \gamma' < 1$, where q is any polynomial of λ.*

Proof of Theorem 3.2. We start with analyzing succinctness, then move to the security proof.

Padding Parameter. The proof of security relies on the indistinguishability of the obfuscated circuits of $\mathsf{E}_{\mathsf{1FE}}$ and $\mathsf{E}^{(j)}$ defined in Figure 2 and 3. Accordingly, we set $\mathsf{pad} := \max(|\mathsf{E}_{\mathsf{1FE}}|, |\mathsf{E}^{(j)}|)$. Let, n and s be the size of m and f, respectively. The circuits $\mathsf{E}_{\mathsf{1FE}}$ and $\mathsf{E}^{(j)}$ compute a puncturable PRF over domain $[q]$, an SKFE master secret key, and may have punctured PRF keys and a hardwired ciphertext. Note that $\mathsf{1FE}$ is independent of q. Thus,

$$\mathsf{pad} \leq \mathrm{poly}(\lambda, n, s, \log q) \ .$$

Weak Collusion-Succinctness. The input space for $\mathsf{E}_{\mathsf{1FE}}$ is $[q]$. Therefore, by the SXIO guarantee, the size of the encryption circuit (dominated by running the obfuscated $\mathsf{E}_{\mathsf{1FE}}$ or $\mathsf{E}^{(j)}$) is

$$\widetilde{q^{\gamma}} \cdot \mathrm{poly}(\lambda, n, s, \log q) < q^{\gamma'} \cdot \mathrm{poly}(\lambda, n, s) \ .$$

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

Hyb_0**:** The first game is the original selective security experiment for $b = 0$, $\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0)$. In this game, $\mathcal{A}$ first selects the challenge messages $(x_0^{(1)}, x_1^{(1)}), \ldots, (x_0^{(q_m)}, x_1^{(q_m)})$, then obtains encryptions of $x_0^{(1)}, \ldots, x_0^{(q_m)}$ and the master public key. After that, it also queries q functions $\{f_i\}_{i\in[q]}$ such that $f_i(x_0^{(j)}) = f_i(x_1^{(j)})$ for all $i \in [q]$ and $j \in [q_m]$ and receives functional keys (see Definition 2.10 for more details).

$\mathsf{Hyb}_1^{i^*}$**:** Let $i^* \in [q]$. For all $j \in [q_m]$, we change target ciphertexts $\mathsf{E}_{\mathsf{1FE}}[K, K', x_0^{(j)}]$ into $\mathsf{E}^{(j)}$ described in Figure 3. In this hybrid game, we set $r_{i^*} = \mathsf{F}_K(i^*)$, $r_{i^*}^{(j)} = \mathsf{F}_{K^{(j)}}(i^*)$, $\mathsf{MSK}_{i^*} \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_{i^*})$, and $\mathsf{CT}_{i^*}^{(j)} \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i^*, x_0^{(j)}; r_{i^*}^{(j)})$ where $K^{(j)} \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ is randomness for the j-th target ciphertext. Thus, when $i^* = 1$, the behaviors of $\mathsf{E}_{\mathsf{1FE}}[K, K', x_0^{(j)}]$ and $\mathsf{E}^{(j)}$ are the same since the hard-wired ciphertexts $\mathsf{CT}_1^{(j)}$ for all $j \in [q_m]$ are the same as those in Hyb_0. Their size is also the same since we pad circuit $\mathsf{E}_{\mathsf{1FE}}[K, K', x_0^{(j)}]$ to have the same size as $\mathsf{E}^{(j)}$. Then, we can use the indistinguishability guarantee of $\mathsf{sxi}\mathcal{O}$ and it holds that $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1^1$. (In fact, we use indistinguishability q_m times to change all q_m ciphertexts.)

Encryption Circuit $\mathsf{E}^{(j)}[K\{i^*\}, K^{(j)}\{i^*\}, x_0^{(j)}, x_1^{(j)}, \mathsf{CT}_{i^*}^{(j)}](i)$

Hardwired: punctured PRF keys $K\{i^*\}, K^{(j)}\{i^*\}$, index i^*, messages $x_0^{(j)}$, $x_1^{(j)}$, and $\mathsf{CT}_{i^*}^{(j)}$.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad} = \mathsf{pad}(\lambda, n, s, q)$, which is determined in the analysis.

1. If $i = i^*$, then output $\mathsf{CT}_{i^*}^{(j)}$.
2. Else if compute $r_i \leftarrow \mathsf{F}_{K\{i^*\}}(i)$ and $r_i^{(j)} \leftarrow \mathsf{F}_{K^{(j)}\{i^*\}}(i)$.
3. Compute $\mathsf{MSK}_i \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$.
4. For $i > i^*$, output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x_0^{(j)}; r_i^{(j)})$.
5. For $i < i^*$, output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x_1^{(j)}; r_i^{(j)})$.

Figure 3: Circuit $\mathsf{E}^{(j)}[K\{i^*\}, K^{(j)}\{i^*\}, i^*, x_0^{(j)}, x_1^{(j)}, \mathsf{CT}_{i^*}^{(j)}]$

$\mathsf{Hyb}_2^{i^*}$**:** We change $r_{i^*} = \mathsf{F}_K(i^*)$ and $r_{i^*}^{(j)} = \mathsf{F}_{K^{(j)}}(i^*)$ into uniformly random r_{i^*} and $r_{i^*}^{(j)}$ for all $j \in [q_m]$. Due to the pseudo-randomness at punctured points, it holds that $\mathsf{Hyb}_1^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_2^{i^*}$.

$\mathsf{Hyb}_3^{i^*}$**:** We change the hard-wired ciphertext $\mathsf{CT}_{i^*}^{(j)}$ from $\mathsf{1FE.Enc}(\mathsf{MSK}_{i^*}, x_0^{(j)})$ to $\mathsf{1FE.Enc}(\mathsf{MSK}_{i^*}, x_1^{(j)})$ for all $j \in [q_m]$. In $\mathsf{Hyb}_2^{i^*}$ and $\mathsf{Hyb}_3^{i^*}$, we do not need the master secret key MSK_{i^*} and randomness for ciphertexts, which are used to generate $\mathsf{CT}_{i^*}^{(j)}$. We just use the hardwired $\mathsf{CT}_{i^*}^{(j)}$ for $i = i^*$. Therefore, $\mathsf{Hyb}_2^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_3^{i^*}$ follows directly from the selective-message message privacy of 1FE.

$\mathsf{Hyb}_4^{i^*}$**:** We change r_{i^*} and $r_{i^*}^{(j)}$ into $r_{i^*} = \mathsf{F}_K(i^*)$ and $r_{i^*}^{(j)} = \mathsf{F}_{K^{(j)}}(i^*)$ and un-puncture $K\{i^*\}$ and $K^{(j)}\{i^*\}$. We can show that $\mathsf{Hyb}_3^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_4^{i^*}$ holds in a reverse manner.

From the definition of $\mathsf{E}^{(j)}[K\{i^*\}, K^{(j)}\{i^*\}, x_0^{(j)}, x_1^{(j)}, \mathsf{CT}_{i^*}^{(j)}]$ and $\mathsf{Hyb}_1^{i^*}$, the behaviors of $\mathsf{E}^{(j)}$ in $\mathsf{Hyb}_4^{i^*}$ and $\mathsf{Hyb}_1^{i^*+1}$ are the same. Thus, $\mathsf{Hyb}_4^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1^{i^*+1}$ holds due to $\mathsf{sxi}\mathcal{O}$. It also holds that $\mathsf{Hyb}_4^q \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 1)$. (Again, we use the security of $\mathsf{sxi}\mathcal{O}$ q_m times.) This completes the proof of Theorem 3.2.

■

3.2 Collusion-Succinct PKFE from SXIO and Public-Key Encryption

In this section, we discuss how to construct a bounded-key weakly collusion-succinct PKFE scheme from an SXIO and PKE scheme.

Our construction. The construction of a iPKFE scheme qFE from an SXIO and public-key encryption scheme is as follows. Let $\mathsf{1FE} = (\mathsf{1FE.Setup}, \mathsf{1FE.KG}, \mathsf{1FE.Enc}, \mathsf{1FE.Dec})$ be a single-key non-succinct PKFE scheme and $(\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Punc})$ a puncturable PRF.

$\mathsf{qFE.Setup}(1^\lambda)$:

- Generate $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- Generate $\mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}})$ where circuit $\mathsf{S}_{\mathsf{1fe}}$ is defined in Figure 4.
- Return $(\widehat{\mathsf{MPK}}, \widehat{\mathsf{MSK}}) := (\mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}}), K)$.

$\mathsf{qFE.iKG}(\widehat{\mathsf{MSK}}, f, i)$:

- Parse $K := \widehat{\mathsf{MSK}}$.
- Compute $r_i \leftarrow \mathsf{F}_K(i)$ and $(\mathsf{MSK}_i, \mathsf{MPK}_i) \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$.
- Compute $\mathsf{sk}_f^i \leftarrow \mathsf{1FE.KG}(\mathsf{MSK}_i, f)$.
- Return $\widehat{\mathsf{sk}}_f \leftarrow (i, \mathsf{sk}_f^i)$.

$\mathsf{qFE.Enc}(\widehat{\mathsf{MPK}}, x)$:

- Parse $\mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}}) := \widehat{\mathsf{MPK}}$.
- Generate $K' \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and $\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x]$ defined in Figure 5.
- Return $\widehat{\mathsf{CT}} \leftarrow \mathsf{sxi}\mathcal{O}(\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x])$.

$\mathsf{qFE.Dec}(\widehat{\mathsf{sk}}_f, \widehat{\mathsf{CT}})$:

- Parse $(i, \mathsf{sk}_f^i) := \widehat{\mathsf{sk}}_f$.
- Compute $\mathsf{CT}_i \leftarrow \widehat{\mathsf{CT}}(i)$.
- Return $y \leftarrow \mathsf{1FE.Dec}(\mathsf{sk}_f^i, \mathsf{CT}_i)$.

Setup Circuit $\mathsf{S}_{\mathsf{1fe}}[K](i)$

Hardwired: puncturable PRF key K.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad}_\mathsf{S} := \mathsf{pad}_\mathsf{S}(\lambda, n, s, q)$, which is determined in analysis.

1. Compute $r_i \leftarrow \mathsf{F}_K(i)$.
2. Compute $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$ and output MPK_i.

Figure 4: Description of $\mathsf{S}_{\mathsf{1fe}}[K]$.

Theorem 3.3. *If there exists $(1, \delta)$-selectively-secure non-succinct PKFE for P/poly and δ-secure γ-compressing SXIO for P/poly where γ is an arbitrarily small constant such that $0 < \gamma < 1$, then there exists (q, δ)-selectively-secure collusion-succinct PKFE for P/poly with compression factor β, where q is any polynomial of λ, and β is a constant such that $0 < \beta < 1$.*

Proof of Theorem 3.3. We start with analyzing succinctness, then move on to the security proof.

Encryption Circuit $\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x](i)$

Hardwired: master public key $\widehat{\mathsf{MPK}}$, puncturable PRF key K', and message x.
Input: an index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad}_{\mathsf{E}} := \mathsf{pad}_{\mathsf{E}}(\lambda, n, s, q)$, which is determined in analysis.

1. Compute $\mathsf{MPK}_i \leftarrow \widehat{\mathsf{MPK}}(i)$.
2. Compute $r'_i \leftarrow \mathsf{F}_{K'}(i)$ and output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x; r'_i)$.

Figure 5: Description of $\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x]$.

Padding Parameter. The proof of security relies on the indistinguishability of obfuscated $\mathsf{S}_{\mathsf{1fe}}$, $\mathsf{S}^*_{\mathsf{1fe}}$, $\mathsf{E}_{\mathsf{1fe}}$, and $\mathsf{E}^*_{\mathsf{1fe}}$ defined in Figures 4, 5, 6, and 7. Accordingly, we set $\mathsf{pad}_{\mathsf{S}} := \max(|\mathsf{S}_{\mathsf{1fe}}|, |\mathsf{S}^*_{\mathsf{1fe}}|)$ and $\mathsf{pad}_{\mathsf{E}} := \max(|\mathsf{E}_{\mathsf{1fe}}|, |\mathsf{E}^*_{\mathsf{1fe}}|)$. The circuits $\mathsf{S}_{\mathsf{1fe}}$ and $\mathsf{S}^*_{\mathsf{1fe}}$ compute a puncturable PRF over domain $[q]$, a key pair of 1FE, and may have punctured PRF keys and a master public key hardwired. The circuits $\mathsf{E}_{\mathsf{1fe}}$ and $\mathsf{E}^*_{\mathsf{1fe}}$ run $\widehat{\mathsf{MPK}}$ and compute a puncturable PRF over domain $[q]$, a ciphertext of 1FE, and may have punctured PRF keys and a hard-wired ciphertext. Note that 1FE is independent of q. Thus, it holds that

$$\mathsf{pad}_{\mathsf{S}} \leq \mathrm{poly}(\lambda, n, s, \log q),$$
$$\mathsf{pad}_{\mathsf{E}} \leq \mathrm{poly}(\lambda, n, s, \log q, |\widehat{\mathsf{MPK}}|).$$

Weak Collusion-Succinctness. Let γ' be a compression factor of the SXIO for $\mathsf{S}_{\mathsf{1fe}}$. The input space for $\mathsf{S}_{\mathsf{1fe}}$ and $\mathsf{E}_{\mathsf{1fe}}$ are $[q]$. Therefore, by the γ'-compressing SXIO guarantee, the size of the setup circuit (dominated by running the obfuscated $\mathsf{S}_{\mathsf{1fe}}$) is

$$q^{\gamma'} \cdot \mathrm{poly}(\lambda, n, s, \log q) \ .$$

Let γ be a compression factor of the SXIO for $\mathsf{E}_{\mathsf{1fe}}$. The size of the encryption circuit $\mathsf{E}_{\mathsf{1fe}}$ (dominated by running the obfuscated $\mathsf{E}_{\mathsf{1fe}}$) is

$$q^{\gamma} \cdot \mathrm{poly}(\lambda, n, s, \log q, |\mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}})|) < q^{\gamma + c\gamma'} \cdot \mathrm{poly}(\lambda, n, s),$$

where c is some constant. We assume there exists SXIO with an arbitrarily small compression factor. Thus, we can take γ' such that $\beta := \gamma + c\gamma' < 1$.

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

Hyb_0**:** The first game is the original selective security experiment for $b = 0$, $\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0)$. $\mathcal{A}$ first selects the challenge messages (x^*_0, x^*_1) and receives the master public key $\mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}}[K])$ and target ciphertext $\mathsf{sxi}\mathcal{O}(\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x^*_0])$. Next, $\mathcal{A}$ queries q pairs $(f_1, i_1), \ldots, (f_q, i_q)$ such that $f_\ell(x^*_0) = f_\ell(x^*_1)$ and receives functional keys $\mathsf{SK}_{f_1}, \ldots, \mathsf{SK}_{f_q}$. (see Definition 2.15 for more details).

$\mathsf{Hyb}^{i^*}_1$**:** Let $i^* \in [q]$ We change $\mathsf{S}_{\mathsf{1fe}}$ into $\mathsf{S}^*_{\mathsf{1fe}}$ described in Figure 6. In this hybrid game, we set $r_{i^*} := \mathsf{F}_K(i^*)$, and $(\mathsf{MPK}_{i^*}, \mathsf{MSK}_{i^*}) := \mathsf{1FE.Setup}(1^\lambda; r_{i^*})$. Thus, when $i^* = 1$, the behavior of $\mathsf{S}_{\mathsf{1fe}}$ is the same as that of $\mathsf{S}^*_{\mathsf{1fe}}$ since the hard-wired MPK_1 is generated by $\mathsf{F}_K(1)$. Their size is also the same since we pad circuit $\mathsf{S}_{\mathsf{1fe}}$ to have the same size as $\mathsf{S}^*_{\mathsf{1fe}}$. Then, we can use the indistinguishability guarantee of $\mathsf{sxi}\mathcal{O}$ and it holds that $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^1_1$.

$\mathsf{Hyb}^{i^*}_2$**:** We change $\mathsf{E}_{\mathsf{1fe}}$ into $\mathsf{E}^*_{\mathsf{1fe}}$ described in Figure 7 when the challenger generates a target ciphertext. In this hybrid game, we set $r'_{i^*} := \mathsf{F}_{K'}(i^*)$, and $\mathsf{CT}_{i^*} := \mathsf{1FE.Enc}(\widehat{\mathsf{MPK}}(i^*), x^*_0; r'_{i^*})$. Thus, the behavior of $\mathsf{E}_{\mathsf{1fe}}$ is the same as that of $\mathsf{E}^*_{\mathsf{1fe}}$, and so is its size since we pad circuit $\mathsf{E}_{\mathsf{1fe}}$ to have the same size as $\mathsf{E}^*_{\mathsf{1fe}}$. Then, we can use the indistinguishability guarantee of $\mathsf{sxi}\mathcal{O}$ and it holds that $\mathsf{Hyb}^{i^*}_1 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_2$.

Setup Circuit $\mathsf{S}^*_{\mathsf{1fe}}[K\{i^*\}, \mathsf{MPK}_{i^*}](i)$

Hardwired: puncturable PRF key $K\{i^*\}$, MPK_{i^*}, and index i^*.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad}_{\mathsf{S}} := \mathsf{pad}_{\mathsf{S}}(\lambda, n, s, q)$, which is determined in analysis.

1. If $i = i^*$, output MPK_{i^*}.
2. Else if compute $r_i \leftarrow \mathsf{F}_{K\{i^*\}}(i)$.
3. Compute $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_i)$ and output MPK_i.

Figure 6: Circuit $\mathsf{S}^*_{\mathsf{1fe}}[K\{i^*\}, \mathsf{MPK}_{i^*}]$.

Encryption Circuit $\mathsf{E}^*_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K'\{i^*\}, x^*_0, x^*_1, \mathsf{CT}_{i^*}](i)$

Hardwired: master public key $\widehat{\mathsf{MPK}}$, puncturable PRF key $K'\{i^*\}$, and index i^*.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad}_{\mathsf{E}} := \mathsf{pad}_{\mathsf{E}}(\lambda, n, s, q)$, which is determined in analysis.

1. Compute $\mathsf{MPK}_i \leftarrow \widehat{\mathsf{MPK}}(i)$.
2. If $i = i^*$, output CT_{i^*}.
3. Else if compute $r'_i \leftarrow \mathsf{F}_{K'}(i)$.

   For $i > i^*$**:** Output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MPK}_i, x^*_0; r'_i)$.

   For $i < i^*$**:** Output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MPK}_i, x^*_1; r'_i)$.

Figure 7: Circuit $\mathsf{E}^*_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K'\{i^*\}, x^*_0, x^*_1, \mathsf{CT}_{i^*}]$.

$\mathsf{Hyb}^{i^*}_3$**:** We change $r_{i^*} = \mathsf{F}_K(i^*)$ and $r'_{i^*} = \mathsf{F}_{K'}(i^*)$ into uniformly random r_{i^*} and r'_{i^*}. Due to the pseudo-randomness at punctured points, it holds that $\mathsf{Hyb}^{i^*}_2 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_3$.

$\mathsf{Hyb}^{i^*}_4$**:** We change CT_{i^*} from $\mathsf{1FE.Enc}(\widehat{\mathsf{MPK}}(i^*), x^*_0)$ to $\mathsf{1FE.Enc}(\widehat{\mathsf{MPK}}(i^*), x^*_1)$. In $\mathsf{Hyb}^{i^*}_3$ and $\mathsf{Hyb}^{i^*}_4$, we do not need randomness to generate MPK_{i^*} and CT_{i^*}. We just use the hardwired CT^*_i and an output of $\widehat{\mathsf{MPK}}$ for input i^*. Note that $\widehat{\mathsf{MPK}}(i^*)$ is the hard-wired MPK_{i^*}. Therefore, $\mathsf{Hyb}^{i^*}_3 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_4$ follows directly from the selective security of the PKFE scheme under the master public key MPK_{i^*}.

$\mathsf{Hyb}^{i^*}_5$**:** We change r^*_i and r'_{i^*} into $r_{i^*} = \mathsf{F}_K(i^*)$ and $r'_{i^*} = \mathsf{F}_{K'}(i^*)$ and un-puncture $K\{i^*\}$ and $K'\{i^*\}$. We can show $\mathsf{Hyb}^{i^*}_4 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_5$ in a reverse manner.

From the definition of $\mathsf{S}^*_{\mathsf{1FE}}$, $\mathsf{E}^*_{\mathsf{1FE}}$, and $\mathsf{Hyb}^{i^*}_1$, the behaviors of $\mathsf{S}^*_{\mathsf{1FE}}$ and $\mathsf{E}^*_{\mathsf{1FE}}$ in $\mathsf{Hyb}^{i^*}_5$ and $\mathsf{Hyb}^{i^*+1}_1$ are the same. Thus, $\mathsf{Hyb}^{i^*}_5 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*+1}_1$ due to $\mathsf{sxi}\mathcal{O}$. It also holds that $\mathsf{Hyb}^{q}_5 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 1)$. This completes the proof of Theorem 3.3. ■

3.3 Collusion-Succinct PKFE from SXIO and Identity-Based Encryption

In this section, we directly construct a weakly collusion-succinct and weakly-selective secure iPKFE scheme from an SXIO and identity-based encryption scheme.

Our construction. The construction of a weakly collusion-succinct and weakly-selective secure q-key PKFE scheme qFE is based on an SXIO, identity-based encryption scheme in the sense of Definition 2.8[12], and garbled circuit, which is implied by a one-way function. Our collusion-succinct PKFE scheme is *weakly-selective* secure (see Definition 2.17) because we use function descriptions as identities and the selective security of identity-based encryption requires adversaries to submit a target identity at the beginning of the game.

Let $\mathsf{IBE} = (\mathsf{IBE.Setup}, \mathsf{IBE.KG}, \mathsf{IBE.Enc}, \mathsf{IBE.Dec})$ be an identity-based encryption scheme whose identity space is $[q] \times [s] \times \{0,1\}$, $\mathsf{GC} = (\mathsf{Grbl}, \mathsf{Eval})$ a garbled circuit, and $(\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Punc})$ a PRF whose domain is $[q] \times [s] \times \{0,1,2\}$. We assume that we can represent every function f by a s bit string $(f[1], \cdots, f[s])$. Let $U(f,x)$ is a universal circuit that computes $f(x)$.

$\mathsf{qFE.Setup}(1^\lambda)$:

- Generate $(\mathsf{MPK}_{\mathsf{ibe}}, \mathsf{MSK}_{\mathsf{ibe}}) \leftarrow \mathsf{IBE.Setup}(1^\lambda)$.
- Set $\mathsf{MPK} := \mathsf{MPK}_{\mathsf{ibe}}$ and $\mathsf{MSK} := \mathsf{MSK}_{\mathsf{ibe}}$ and return $(\mathsf{MPK}, \mathsf{MSK})$.

$\mathsf{qFE.iKG}(\mathsf{MSK}, f, i)$:

- Parse $\mathsf{MSK}_{\mathsf{ibe}} \leftarrow \mathsf{MSK}$ and $(f[1], \cdots, f[s]) := f$.
- For $j \in [s]$, compute $\mathsf{SK}^j \leftarrow \mathsf{IBE.KG}(\mathsf{MSK}_{\mathsf{ibe}}, (i, j, f[j]))$.
- Return $\mathsf{SK}_f := (i, f, \{\mathsf{SK}^j\}_{j\in[s]})$.

$\mathsf{qFE.Enc}(\mathsf{MPK}, x)$:

- Parse $\mathsf{MPK}_{\mathsf{ibe}} \leftarrow \mathsf{MPK}$ and choose $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- Return $\mathsf{CT}_{\mathsf{fe}} := \mathsf{sxi}\mathcal{O}(\mathsf{EL}_{\mathsf{gc}}[\mathsf{MPK}_{\mathsf{ibe}}, K, x])$. $\mathsf{EL}_{\mathsf{gc}}$ is defined in Figure 8.

$\mathsf{qFE.Dec}(\mathsf{SK}_f, \mathsf{CT}_{\mathsf{fe}})$:

- Parse $(i, f, \{\mathsf{SK}^j\}_{j\in[s]}) := \mathsf{SK}_f$.
- Compute $(\widetilde{U}, \{\mathsf{CT}^{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{CT}_{\mathsf{fe}}(i)$.
- For $j \in [s]$, compute $L_j \leftarrow \mathsf{IBE.Dec}(\mathsf{SK}^j, \mathsf{CT}^{j,f[j]})$.
- Return $y := \mathsf{Eval}(\widetilde{U}, \{L_j\}_{j\in[s]})$.

Garbling with encrypted labels circuit $\mathsf{EL}_{\mathsf{gc}}[\mathsf{MPK}_{\mathsf{ibe}}, K, x]$

Hardwired: puncturable PRF key K, public parameter of IBE $\mathsf{MPK}_{\mathsf{ibe}}$, and plaintext x.
Input: index $i \in [q]$.
Padding: circuit is padded to size $\mathsf{pad}_{\mathsf{EL}} := \mathsf{pad}_{\mathsf{EL}}(\lambda, n, s, q)$, which is determined in analysis.

1. Compute $r_{\mathsf{gc}} \leftarrow \mathsf{F}_K(i\|1\|2)$.
2. Compute $(\widetilde{U}, \{L_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x); r_{\mathsf{gc}})$.
3. For $j \in [s]$ and $\alpha \in \{0,1\}$, compute $r_{i\|j\|\alpha} \leftarrow \mathsf{F}_K(i\|j\|\alpha)$ and $\mathsf{CT}^{j,\alpha} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i, j, \alpha), L_{j,\alpha}; r_{i\|j\|\alpha})$.
4. Return $(\widetilde{U}, \{\mathsf{CT}^{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}})$.

Figure 8: The description of $\mathsf{EL}_{\mathsf{gc}}$. In the description, $U(\cdot, x)$ is a universal circuit in which x is hardwired as the second input.

[12]Again, we stress that the size of the encryption circuit is $|\mathcal{ID}|^\alpha \cdot \mathrm{poly}(\lambda, \ell)$ where ℓ is the length of plaintext and $\mathcal{ID}$ is the identity-space and most identity-based encryption schemes based on concrete assumptions have such succinct encryption circuits. In our scheme, $\mathcal{ID}$ is just a polynomial size.

Theorem 3.4. *If there exists δ-selectively-secure succinct identity-based encryption with α-compression (α is a sufficiently small constant) and δ-secure $\widetilde{\gamma}$-compressing SXIO for P/poly for a constant $\widetilde{\gamma}$ such that $0 < \widetilde{\gamma} < 1$ ($\widetilde{\gamma}$ might be close to 1), then there exists weakly collusion-succinct (q, δ)-weakly-selective-secure PKFE with compression factor β, where q is any polynomial of λ and $\widetilde{\gamma} < \beta < 1$.*

Proof of Theorem 3.4. We start with analyzing succinctness then moving on to the security proof.

Padding Parameter. The proof of security relies on the indistinguishability of obfuscated $\mathsf{EL}_{\mathsf{gc}}$ and $\mathsf{EL}^*_{\mathsf{gc}}$ defined in Figures 8 and 9. Accordingly, we set $\mathsf{pad}_{\mathsf{EL}} := \max(|\mathsf{EL}_{\mathsf{gc}}|, |\mathsf{EL}^*_{\mathsf{gc}}|)$. The circuits $\mathsf{EL}_{\mathsf{gc}}$ and $\mathsf{EL}^*_{\mathsf{gc}}$ compute a puncturable PRF over domain $[q]$, IBE ciphertext, and garbled circuit of $U(\cdot, x)$ and may have punctured PRF keys and a hard-wired ciphertext. Note that $|\mathcal{ID}| = 2qs$ Thus, due to the efficiency of IBE, it holds that

$$\mathsf{pad}_{\mathsf{EL}} \leq (2qs)^{\alpha}\mathrm{poly}(\lambda, n) + \mathrm{poly}(\lambda, n, s, \log q) \ .$$

Weak Collusion-Succinctness. The input space for $\mathsf{EL}_{\mathsf{gc}}$ is $[q]$. Note that the size of set S^* in $\mathsf{EL}^*_{\mathsf{gc}}$ is logarithmic in q. Therefore, by the SXIO guarantee, the size of the encryption circuit (dominated by running the obfuscated $\mathsf{EL}_{\mathsf{gc}}$) is bounded by

$$\begin{aligned} q^{\widetilde{\gamma}} \cdot \mathrm{poly}(\lambda, \mathsf{pad}_{\mathsf{EL}}) &< q^{\widetilde{\gamma}+c\alpha} \cdot \mathrm{poly}(\lambda, n, s) \\ &< q^{\beta} \cdot \mathrm{poly}(\lambda, n, s), \end{aligned}$$

where c is some constant if we choose α such that $\widetilde{\gamma} + c\alpha < \beta$ (which is possible since α is sufficiently small constant and c is a constant).

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

Hyb_0**:** The first game is the original weakly-selective security experiment for $b = 0$, $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0)$. In this game, $\mathcal{A}$ first selects the challenge messages (x^*_0, x^*_1) and queries q pairs $(f_1, i_1), \ldots, (f_q, i_q)$ such that $f_\ell(x^*_0) = f_\ell(x^*_1)$ for all $\ell \in [q]$, then obtains an encryption of x^*_0, the master public key, and functional keys $\mathsf{SK}_{f_1}, \ldots, \mathsf{SK}_{f_q}$. (see Definition 2.17 for more details).

$\mathsf{Hyb}^{i^*}_1$**:** We change $\mathsf{EL}_{\mathsf{gc}}$ into $\mathsf{EL}^*_{\mathsf{gc}}$ described in Figure 9. In this hybrid game, we set $r^*_{\mathsf{gc}} = \mathsf{F}_K(i^*\|1\|2)$, $r^*_{i^*\|j\|\alpha} = \mathsf{F}_K(i^*\|j\|\alpha)$, $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0); r^*_{\mathsf{gc}})$, and $\mathsf{CT}^{j,\alpha}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i^*, j, \alpha), L_{j,\alpha}; r^*_{j\|\alpha})$. Hereafter, we use $r^*_{j\|\alpha}$ instead of $r^*_{i^*\|j\|\alpha}$ for ease of notation. Thus, when $i^* = 1$, the behaviors of $\mathsf{EL}_{\mathsf{gc}}$ and $\mathsf{EL}^*_{\mathsf{gc}}$ are the same from the definition of $\mathsf{EL}^*_{\mathsf{gc}}$, and so are their size since we pad circuit $\mathsf{EL}_{\mathsf{gc}}$ to have the same size as $\mathsf{EL}^*_{\mathsf{gc}}$. Then, we can use the indistinguishability guarantee of $\mathsf{sxi}\mathcal{O}$, and it holds that $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^1_1$.

$\mathsf{Hyb}^{i^*}_2$**:** We change $r^*_{\mathsf{gc}} = \mathsf{F}_K(i^*\|1\|2)$ and $r^*_{j\|\alpha} = \mathsf{F}_K(i^*\|j\|\alpha)$ into uniformly random r^*_{gc} and $r^*_{j\|\alpha}$ for all $j \in [s]$ and $\alpha \in \{0, 1\}$. We define $S^* := \left\{i^*\|1\|2, \{i^*\|j\|\alpha\}_{j\in[s],\alpha\in\{0,1\}}\right\}$. Due to the pseudo-randomness at punctured points, it holds that $\mathsf{Hyb}^{i^*}_1 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_2$.

$\mathsf{Hyb}^{i^*}_3$**:** For ease of notation, let $f^* := f_{i^*}$ and $\overline{f}$ be the complement of f, that is, $(\overline{f}[1], \ldots, \overline{f}[s]) := (1 - f[1], \ldots, 1 - f[s])$. Moreover, we omit each randomness for $\mathsf{IBE.Enc}$ since it is uniformly random at this hybrid game. For labels of $(\overline{f^*}[1], \ldots, \overline{f^*}[j])$, we change

- normal ciphertexts $\mathsf{CT}^{j,\overline{f^*}[j]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i^*, j, \overline{f^*}[j]), L_{j,\overline{f^*}[j]})$ into
- junk ciphertexts $\mathsf{CT}^{j,\overline{f^*}[j]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i^*, j, \overline{f^*}[j]), 0^{\ell(\lambda)})$, where ℓ is a polynomial denoting the length of labels output by Grbl.

That is, for identities that $\mathcal{A}$ did *not query*, we do not encrypt corresponding labels. We do not change $\mathsf{CT}^{j,f^*[j]}_{i^*}$. Note that all $f_1, \ldots, f_q$ are known in advance since we consider weakly-selective security. $\mathcal{A}$ is *not* given secret keys of IBE for identity $(i^*, j, \overline{f^*}[j])$, so this change is not detected. We show $\mathsf{Hyb}^{i^*}_2 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_3$ in Lemma 3.5 by using the selective security of IBE.

Garbling with encrypted labels circuit $\mathsf{EL}^*_{\mathsf{gc}}[\mathsf{MPK}_{\mathsf{ibe}}, i^*, K\{S^*\}, x^*_0, x^*_1, \{\mathsf{CT}^{j,\alpha}_{i^*}\}_{j,\alpha}]$

Hardwired: punctured PRF key $K\{S^*\}$ where $S^* := \left\{i^*\|1\|2, \{i^*\|j\|\alpha\}_{j\in[s],\alpha\in\{0,1\}}\right\}$, index i^*, set S^*, public parameter of IBE $\mathsf{MPK}_{\mathsf{ibe}}$, challenge plaintexts x^*_0, x^*_1, $\widetilde{U}^*$, and $\{\mathsf{CT}^{j,\alpha}_{i^*}\}_{j\in[s],\alpha\in\{0,1\}}$.

Input: index $i \in [q]$.

Padding: circuit is padded to size $\mathsf{pad}_{\mathsf{EL}} := \mathsf{pad}_{\mathsf{EL}}(\lambda, n, s, q)$, which is determined in analysis.

1. If $i = i^*$, then output $(\widetilde{U}^*, \{\mathsf{CT}^{j,\alpha}_{i^*}\}_{j\in[s],\alpha\in\{0,1\}})$.
2. Else if compute $r_{\mathsf{gc}} \leftarrow \mathsf{F}(K, i\|1\|2)$.

   For $i > i^*$**:** Compute $(\widetilde{U}, \{L_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0); r_{\mathsf{gc}})$.

   For $i < i^*$**:** Compute $(\widetilde{U}, \{L_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_1); r_{\mathsf{gc}})$.
3. For $j \in [s]$ and $\alpha \in \{0,1\}$, compute $r_{i\|j\|\alpha} \leftarrow \mathsf{F}(K, i\|j\|\alpha)$ and $\mathsf{CT}^{j,\alpha}_i \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i,j,\alpha), L_{j,\alpha}; r_{i\|j\|\alpha})$.
4. Return $(\widetilde{U}, \{\mathsf{CT}^{j,\alpha}_i\}_{j\in[s],\alpha\in\{0,1\}})$.

Figure 9: The description of $\mathsf{EL}^*_{\mathsf{gc}}$. In the description, $U(\cdot, m)$ is a universal circuit in which m is hardwired as the second input.

Lemma 3.5. *It holds that* $\mathsf{Hyb}^{i^*}_2 \stackrel{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_3$ *if* IBE *is selectively secure.*

Proof. First, we define more hybrid games H_{j^*} for $j^* \in [s]$ as follows.

H_{j^*}**:** This is the same as $\mathsf{Hyb}^{i^*}_2$ except that for $j \le j^*$, $\mathsf{CT}^{j,\overline{f}^*[j]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j\|\overline{f}^*[j], 0^\ell)$. Apparently, H_0 and H_s are the same as $\mathsf{Hyb}^{i^*}_2$ and $\mathsf{Hyb}^{i^*}_3$, respectively.

We show that $\mathsf{H}_{j^*-1} \stackrel{\mathsf{c}}{\approx}_\delta \mathsf{H}_{j^*}$ holds for all $j^* \in [s]$. This immediately implies the lemma.

We construct an adversary $\mathcal{B}$ in the selective security game of IBE as follows. To simulate the weakly-selective security of PKFE, $\mathcal{B}$ runs $\mathcal{A}$ of qFE and receives a message pair (x^*_0, x^*_1) and function queries $(f_1, \ldots, f_q)$ with indices. $\mathcal{B}$ simulates the game of qFE as follows.

Setup and Encryption: $\mathcal{B}$ sets $\mathsf{id}^* := i^*\|j^*\|\overline{f}^*[j^*]$ as the target identity to the challenger of IBE. Note that $f^* = f_{i^*}$.

To set challenge messages of IBE, $\mathcal{B}$ computes $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0); r^*_{\mathsf{gc}})$ and sets $m^*_0 := L^*_{j^*,\overline{f}^*[j^*]}$ and $m_1 := 0^{\ell(\lambda)}$. $\mathcal{B}$ sends id^* and (m^*_0, m^*_1) to the challenger of IBE, and receives $\mathsf{MPK}_{\mathsf{ibe}}$ and $\mathsf{CT}^{j^*,\overline{f}^*[j^*]}_{i^*}$ as the master public-key and target ciphertext of IBE. $\mathcal{B}$ sets $\mathsf{MPK} := \mathsf{MPK}_{\mathsf{ibe}}$. To simulate ciphertexts of qFE, $\mathcal{B}$ does the followings.

- For all $j \le j^* - 1$, $\mathcal{B}$ computes $\mathsf{CT}^{j,f^*[j]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j\|f^*[j], L_{j,f^*[j]})$ and $\mathsf{CT}^{j,\overline{f}^*[j]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j\|\overline{f}^*[j^*]), 0^\ell)$.
- For $j = j^*$, $\mathcal{B}$ computes $\mathsf{CT}^{j^*,f^*[j^*]}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j^*\|f^*[j^*], L_{j^*,f^*[j^*]})$.
- For all $j \ge j^* + 1$ and $\alpha \in \{0,1\}$, $\mathcal{B}$ computes $\mathsf{CT}^{j,\alpha}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j\|\alpha), L_{j,\alpha})$.

By using these ciphertexts $\{\mathsf{CT}^{j,\alpha}_{i^*}\}_{j\in[s],\alpha\in\{0,1\}}$, $\mathcal{B}$ construct program $\mathsf{EL}^*_{\mathsf{gc}}$ and sets $\mathsf{CT}^*_{\mathsf{fe}} := \mathsf{sxi}\mathcal{O}(\mathsf{EL}^*_{\mathsf{gc}})$ as the target ciphertext of qFE.

Key Generation: Then, $\mathcal{B}$ queries identities $(i, 1, f_i[1]), \ldots, (i, s, f_i[s])$ for all $i \in [q]$ to the challenger of IBE, receives $\mathsf{SK}^j_i \leftarrow \mathsf{IBE.KG}(\mathsf{MSK}_{\mathsf{ibe}}, i\|j\|f_i[j])$, and sets $\mathsf{SK}_{f_i} := (i, f_i, \{\mathsf{SK}^j_i\}_{j\in[s]})$ for all $i \in [q]$. Note that $(i^*\|j^*\|\overline{f}^*[j^*])$ is not queried.

Now $\mathcal{B}$ sets all values for $\mathcal{A}$ and sends MPK, $\{\mathsf{SK}_{f_i}\}_{i\in[q]}$, and $\mathsf{CT}^*_{\mathsf{fe}}$ to $\mathcal{A}$. If $\mathcal{B}$ is given $\mathsf{CT}^{j^*,\bar{f}^*[j^*]}_{i^*} = \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j^*\|\bar{f}^*[j^*]), L_{j^*,\bar{f}^*[j^*]})$, then $\mathcal{B}$ perfectly simulates H_{j^*-1}. If $\mathcal{B}$ is given $\mathsf{CT}^{j^*,\bar{f}^*[j^*]}_{i^*} = \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^*\|j^*\|\bar{f}^*[j^*], 0^{\ell(\lambda)})$, then $\mathcal{B}$ perfectly simulates H_{j^*}. Therefore, the advantage between H_{j^*-1} and H_{j^*} is bounded by the advantage of IBE and it holds that $\mathsf{H}_{j^*-1} \overset{\mathsf{c}}{\approx}_\delta \mathsf{H}_{j^*}$. This completes the proof of the lemma. ■

$\mathsf{Hyb}^{i^*}_4$**:** We change $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0); r^*_{\mathsf{gc}})$ into simulated garbled circuit $(\widetilde{U}^*, \{L^*_{j,f^*[j]}\}_{j\in[s]}) \leftarrow \mathsf{Sim.GC}(1^\lambda, f^*(x^*_0); r^*_{\mathsf{gc}})$. Now a simulator for $\mathsf{Hyb}^{i^*}_4$ does not have $\{L^*_{j,\bar{f}^*[j]}\}$ since the simulator of GC cannot generate them. However, the simulator for $\mathsf{Hyb}^{i^*}_4$ does not need them since it generates junk ciphertexts for such labels as in $\mathsf{Hyb}^{i^*}_3$. It holds that $\mathsf{Hyb}^{i^*}_3 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_4$ due to the security of the garbled circuit.

$\mathsf{Hyb}^{i^*}_5$**:** We change $(\widetilde{U}^*, \{L^*_{j,f^*[j]}\}_{j\in[s]}) \leftarrow \mathsf{Sim.GC}(1^\lambda, f^*(x^*_0); r^*_{\mathsf{gc}})$ into $(\widetilde{U}^*, \{L^*_{j,f^*[j]}\}_{j\in[s]}) \leftarrow \mathsf{Sim.GC}(1^\lambda, f^*(x^*_1)); r^*_{\mathsf{gc}})$. By the requirement of the security game, $f_i(x^*_0) = f_i(x^*_1)$ holds for all $i \in [q]$. Thus, the distribution of $(\widetilde{U}^*, \{L^*_{j,f^*[j]}\}_{j\in[s]})$ is perfectly the same and it holds that $\mathsf{Hyb}^{i^*}_4 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_5$ due to $\mathsf{sxi}\mathcal{O}$.

$\mathsf{Hyb}^{i^*}_6$**:** We change the simulated garbled circuit, junk ciphertexts, and punctured PRF keys back into the real garbled circuit, normal IBE ciphertexts, and un-punctured PRF keys. In this hybrid game, we set $r^*_{\mathsf{gc}} = \mathsf{F}_K(i^*\|1\|2)$, $r^*_{j\|\alpha} = \mathsf{F}_K(i^*\|j\|\alpha)$, $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_1); r^*_{\mathsf{gc}})$, and $\mathsf{CT}^{j,\alpha}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i^*, j, \alpha), L_{j,\alpha}; r^*_{j\|\alpha})$. We can show $\mathsf{Hyb}^{i^*}_5 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_6$ in a reverse manner.

It holds $\mathsf{Hyb}^{i^*}_6 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*+1}_1$ by the definition of $\mathsf{EL}^*_{\mathsf{gc}}$ and $\mathsf{sxi}\mathcal{O}$. That is, $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0) = \mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^1_1 \overset{\mathsf{c}}{\approx}_\delta \cdots \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^q_6 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 1)$ holds. ■

4 Weakly Succinct FE from Collusion-Succinct FE

In this section, we see a transformation from a q-key weakly collusion-succinct index based functional encryption into a single-key weakly succinct one. Bitansky and Vaikuntanathan have shown such a transformation [BV15, Proposition IV.1]. Ananth, Jain, and Sahai show a transformation from a *collusion-resistant* non-succinct FE into a (collusion-resistant) succinct one [AJS15]. It is easy to verify that the transformation by Ananth *et al.* also works for q-key collusion-succinct functional encryption schemes to achieve single-key weakly succinct ones. The key tool for these transformation is decomposable randomized encoding, which is implied by one-way function (see Definition 2.22).

We stress that the transformation in this section is *not new*. The differences between theirs and ours is that we assume that the underlying weakly collusion-succinct scheme is *weakly-selective secure* and uses an index for functional key generation. To construct IO by using the theorem by Bitansky and Vaikuntanathan [BV15], a single-key *weakly-selective secure* weakly succinct PKFE scheme is sufficient. Adversaries should query a function before it receives the public parameter. Note that if the maximum size of functions in a function family is fixed, the size of a randomized encoding (denoted by μ) of a function is also fixed. Thus, (μ, δ)*-weakly-selective secure* schemes are sufficient for this transformation. We can easily observe this fact. Moreover, the index-based functional key generation is not an issue since our goal is a single-key scheme and μ is fixed in advance. Thus, readers that are familiar with the transformation by Bitansky and Vaikuntanathan [BV15, Proposition IV.1] can skip this section. We write the transformation and a proof for the weakly-selective security for confirmation. Of course, we can obtain a selectively secure scheme by the transformation if we use a selectively secure scheme as the underlying scheme.

Conversion. Our single-key weakly succinct PKFE scheme $\mathsf{sFE} = (\mathsf{sFE.Setup}, \mathsf{sFE.KG}, \mathsf{sFE.Enc}, \mathsf{sFE.Dec})$ for circuits of size at most $s = s(\lambda)$ with $n = n(\lambda)$ bit inputs is based on a q-key weakly collusion-succinct iPKFE scheme $\mathsf{qFE} = (\mathsf{qFE.Setup}, \mathsf{qFE.iKG}, \mathsf{FE.Enc}, \mathsf{qFE.Dec})$ for circuits of size at most s with n bit inputs. Let F, RE, and SKE be a PRF, c-local decomposable randomized encoding, and secret-key encryption scheme, respectively. The construction is essentially the same as the FE scheme from any weakly collusion-succinct FE scheme by Bitansky and Vaikuntanathan [BV15, Proposition IV.1]. In the scheme, we use $\mathsf{F} : \{0,1\}^\lambda \times [c] \to \{0,1\}$.

$\mathsf{sFE.Setup}(1^\lambda)$:

- Generate $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{qFE.Setup}(1^\lambda)$.
- Return $(\mathsf{MPK}, \mathsf{MSK})$.

$\mathsf{sFE.KG}(\mathsf{MSK}, f)$:

- Generate $t \leftarrow \{0,1\}^\lambda$.
- Compute decomposed f, that is, $(\widehat{f}_1, \ldots, \widehat{f}_\mu)$ together with $(S_1, \cdots, S_\mu)$ where $S_i \subseteq [\rho]$ and $|S_i| = c$.
- For $i \in [\mu]$, choose $\mathsf{SK}_i \leftarrow \{0,1\}^\lambda$, generate $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}_i, 0)$, and compute $\mathsf{sk}_{f_i} \leftarrow \mathsf{qFE.iKG}(\mathsf{MSK}, \mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}], i)$. The circuit D_{re} is defined in Figure 10.
- Return $\mathsf{sk}_f \leftarrow (\mathsf{sk}_{f_1}, \cdots, \mathsf{sk}_{f_\mu})$.

$\mathsf{sFE.Enc}(\mathsf{MPK}, x)$:

- Generate $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- Return $\mathsf{CT} \leftarrow \mathsf{qFE.Enc}(\mathsf{MPK}, (0, x, K, \bot))$.

$\mathsf{sFE.Dec}(\mathsf{sk}_f, \mathsf{CT})$:

- Parse $(\mathsf{sk}_{f_1}, \cdots, \mathsf{sk}_{f_\mu}) \leftarrow \mathsf{sk}_f$.
- For $i \in [\mu]$, compute $e_i \leftarrow \mathsf{qFE.Dec}(\mathsf{sk}_{f_i}, \mathsf{CT})$.
- Decode y from $(e_1, \cdots, e_\mu)$.
- Return y.

Decomposable Randomized Encoding Circuit $\mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$

Hardwired: decomposed function $\widehat{f}_i$, set S_i, tag t, and ciphertext $\mathsf{CT}^i_{\mathsf{ske}}$
Input: bit b, message x, PRF key K, and SKE secret key SK_i

1. If $b = 1$, return $e_i \leftarrow \mathsf{SKE.Dec}(\mathsf{SK}_i, \mathsf{CT}^i_{\mathsf{ske}})$.
2. Else for $j \in S_i$, compute $r_j \leftarrow \mathsf{F}_K(t \| j)$, set $r_{S_i} \leftarrow \{r_j\}_{j \in S_i}$ where r_j is j-th bit of r.
3. Return $e_i \leftarrow \widehat{f}_i(x; r_{S_i})$.

Figure 10: Description of D_{re}.

Theorem 4.1. *If there exists weakly collusion-succinct (μ, δ)-weakly-selective-secure PKFE (resp. SKFE) for circuits of size at most $s = s(\lambda)$ with $n = n(\lambda)$ inputs with encryption circuit of size $\mu^\gamma \cdot \mathrm{poly}(\lambda, n, s)$ where $\mu = s \cdot \mathrm{poly}'(\lambda, n)$ and poly and poly' are fixed polynomials, then there exists weakly succinct $(1, \delta)$-weakly-selective-secure PKFE (resp. SKFE) for circuits of size at most $s = s(\lambda)$ with encryption circuit of size $s^\gamma \cdot \mathrm{poly}''(\lambda, n)$ where poly'' is a fixed polynomial.*

We show only the PKFE case. The SKFE case is similarly proven.

Proof of Theorem 4.1. We start with analyzing succinctness then move on to the security proof.

Weak Succinctness. Let $\mathsf{D}_i := \mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$. To issue one key, we need to issue $1 \cdot \mu = s \cdot \mathrm{poly}(\lambda, n)$ keys of qFE since we consider functions of size s. Thus, we choose $s \cdot \mathrm{poly}(\lambda, n)$ as the number of issued keys of PKFE. Note that $\mathrm{poly}(\lambda, n)$ is determined by RE. The size of D_i is $\mathrm{poly}(\lambda, n)$ since $|\widehat{f}_i|$ is independent of $|f|$ by the decomposability of RE and $|\tau|$ and $|\mathsf{CT}^i_{\mathsf{ske}}|$ are bounded by $O(\lambda)$. The size of encryption circuit sFE.Enc is

$$(s \cdot \mathrm{poly}(\lambda, n))^\gamma \cdot \mathrm{poly}(\lambda, n) = s^\gamma \cdot \mathrm{poly}(\lambda, n) \ .$$

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

Hyb_0**:** The first game is the original weakly-selective security experiment for $b = 0$, $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0)$. In this game, $\mathcal{A}$ first selects the challenge messages (x_0^*, x_1^*) and a function f then obtains an encryption of x_0^*, the master public key, and a functional decryption key sk_f.

Hyb_1**:** We change $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}_i, 0)$ into $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{sk}, \widehat{f}_i(x_0^*; r_{S_i}))$. It holds that $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1$ due to the security of SKE.

Hyb_2**:** We change $\mathsf{CT} \leftarrow \mathsf{qFE.Enc}(\mathsf{MPK}, (0, x, K, \bot))$ into $\mathsf{CT} \leftarrow \mathsf{qFE.Enc}(\mathsf{MPK}, (1, x, \bot, \mathsf{sk}))$.

Lemma 4.2. *It holds that* $\mathsf{Hyb}_1 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_2$ *if* qFE *is a* (q, δ)*-weakly-selective-secure PKFE.*

Proof of lemma. We construct an adversary $\mathcal{B}$ of qFE. First, $\mathcal{A}$ sends messages (x_0^*, x_1^*) and a function f to the challenger of sFE. $\mathcal{B}$ generates $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and chooses random t, generates $\mathsf{sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$, computes $(\widehat{f}_1, \ldots, \widehat{f}_\mu)$ from f together with $(S_1, \ldots, S_\mu)$, generates $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{sk}, \widehat{f}_i(x_0^*; r_{S_i}))$, and construct $\mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$ for all $i \in [\mu]$. Note that μ is fixed when we design sFE for circuits of size at most s. However, the master public-key of qFE is not set yet. Thus, we can use the weakly-selective security of the μ-key scheme. $\mathcal{B}$ sends messages $((0, x_0^*, K, \bot), (1, x_0^*, \bot, \mathsf{sk}))$ as challenge messages and functions $\mathsf{D}_i := \mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$ to the challenger of qFE and receives MPK, CT^*, and $\{\mathsf{sk}_{\mathsf{D}_i}\}_{i\in[\mu]}$. $\mathcal{B}$ passes MPK, CT^*, and $\{\mathsf{sk}_{\mathsf{D}_i}\}_{i\in[\mu]}$ as the master public-key, target ciphertext, and functional key for f to $\mathcal{A}$. This perfectly simulates Hyb_1 if CT^* is an encryption of $(0, x_0^*, K, \bot)$ and Hyb_2 if CT^* is an encryption of $(1, x_1^*, \bot, \mathsf{sk})$. Thus, the lemma follows. ■

Hyb_3**:** We change $r_j \leftarrow \mathsf{F}_K(t\|j)$ into $r_j \leftarrow \{0,1\}$ for all $j \in [\rho]$. It holds that $\mathsf{Hyb}_2 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_3$ due to the pseudo-randomness of F.

Hyb_4**:** We change $e_i \leftarrow \widehat{f}_i(x_0^*; r_{S_i})$ into $e_i \leftarrow \widehat{f}_i(x_1^*; r_{S_i})$. It holds that $\mathsf{Hyb}_3 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_4$ due to the security of the decomposable randomized encoding and the condition $f(x_0^*) = f(x_1^*)$ for sFE. In fact, we intermediately use the output of the simulator of RE.

This completes the proof of Theorem 4.1. ■

5 Putting It Altogether: Single-Key Weakly Succinct PKFE and IO

Before we summarize our theorems, we introduce a few known theorems regarding SKFE and SXIO. Brakerski, Komargodski, and Segev [BKS16] and Bitansky *et al.* [BNPW16a, BNPW16b] prove the following theorems.

Theorem 5.1 ([BKS16, BNPW16a]). *If there exists a non-succinct collusion-resistant SKFE for* $\mathsf{P/poly}$*, then there exists a* γ*-compressing SXIO for* $\mathsf{P/poly}$ *where* γ *is an arbitrary constant such that* $0 < \gamma < 1$*. (*γ *could be sufficiently small)*

Theorem 5.2 ([BNPW16b]). *If there exists a single-key weakly succinct SKFE for* $\mathsf{P/poly}$*, then there exists a* $\widetilde{\gamma}$*-compressing SXIO for* $\mathsf{P/poly}$ *where* $\widetilde{\gamma}$ *is a constant such that* $1/2 \leq \widetilde{\gamma} < 1$*.*

In this theorem, *weakly*-selective security of a SKFE scheme (see Definition 2.14) is sufficient for the transformation though Bitansky *et al.* do not explicitly point out it.

Theorem 5.3 ([GS16, LM16]). *If there exists a* $(1, \delta)$*-selectively secure and weakly succinct PKFE scheme for* $\mathsf{P/poly}$*, then there exists a* (poly, δ)*-selectively secure and succinct PKFE scheme for* $\mathsf{P/poly}$*.*

5.1 Main Theorems

We summarize our theorems. By Theorem 3.1, 3.2, and 4.1, we obtain the following theorem.

Theorem 5.4. *If there exists one-way function and* $\widetilde{\gamma}$*-compressing SXIO for* $\mathsf{P/poly}$ *for a constant* $\widetilde{\gamma}$ *such that* $0 < \widetilde{\gamma} < 1$ *(*$\widetilde{\gamma}$ *might be close to* 1*), there exists a single-key selective-message message private and weakly succinct SKFE for* $\mathsf{P/poly}$*.*

By Theorem 3.1, 3.3, 4.1, and 5.1, we obtain the following theorem since Theorem 3.3 requires a sufficiently small compression factor.

Theorem 5.5. *If there exists a plain public-key encryption scheme and collusion-resistant and non-succinct SKFE for* P/poly*, then there exists a single-key selectively secure and weakly succinct PKFE for* P/poly.

Combined with Theorem 2.19, IO is obtained from a sub-exponentially secure plain public-key encryption and sub-exponentially secure collusion-resistant (non-succinct) SKFE for P/poly. This theorem has already been proved by Bitansky *et al.* [BNPW16a], but our construction and proof are significantly simpler than theirs as we show in Section 3.2. Moreover, our construction avoids $2^{O(d)}$ security loss where d is the depth of circuits. That is, we obtain the following corollary by using Theorem 5.3.

Corollary 5.6. *If there exists a plain public-key encryption and collusion-resistant and non-succinct SKFE scheme for* P/poly*, then there exists a collusion-resistant and succinct PKFE scheme for* P/poly. This transformation incurs only polynomial security loss.

By Theorem 3.4, 4.1, and 5.2, we obtain the following theorem since Theorem 3.4 just requires that $\widetilde{\gamma}$ is slightly smaller than 1 (no need to be sufficiently small).

Theorem 5.7. *If there exists identity-based encryption and single-key weakly succinct SKFE for* P/poly*, then there exists a single-key weakly-selective secure weakly succinct PKFE for* P/poly.

Combined with Theorem 2.19, IO is obtained if we assume sub-exponential security of identity-based encryption and single-key weakly succinct SKFE.

Theorem 5.8. *If there exits a sub-exponentially secure plain public-key encryption and sub-exponentially secure single-key weakly succinct SKFE for* P/poly*, then there exists IO.*

T Figure 12 illustrates our theorems.

5.2 By-products of Theorem 5.4

Before we show our corollaries, we introduce a few known facts.

Theorem 5.9 ([BNPW16a]). *If there exits a constant-arity MIFE for* P/poly*, then there exits a* γ*-compressing SXIO for* P/poly.

Theorem 5.10 ([BKS16]). *If there exits a collusion-resistant non-succinct SKFE for* P/poly*, then there exits a constant-arity MIFE for* P/poly.

Theorem 5.11 ([ACJ16]). *A single-key weakly succinct SKFE for* P/poly *implies output-compact updatable randomized encoding with an unbounded number of updates.*

Theorem 5.12 ([ACJ16]). *Output-compact updatable randomized encoding with the unbounded number of updates implies a* $\widetilde{\gamma}$*-compressing SXIO for* P/poly.

Note that Ananth *et al.* prove Theorem 5.12 for a $\widetilde{\gamma}$-compressing XIO, but it is easy to observe that their construction of XIO is actually a $\widetilde{\gamma}$-compressing SXIO.

The following theorem is proved in a concurrent work.

Theorem 5.13 ([Ano17]). *If there exists a single-key selectively secure and weakly succinct SKFE for* P/poly*, then there exists a collusion-resistant SKFE for* P/poly.

Corollaries of Theorem 5.4. By Theorem 5.2 and 5.4, we can obtain the following corollaries.

Corollary 5.14. *A single-key weakly succinct SKFE for* P/poly *is equivalent to one-way function and* $\widetilde{\gamma}$*-compressing SXIO for* P/poly *such that* $0 < \widetilde{\gamma} < 1$ *(*$\widetilde{\gamma}$ *might be close to* 1*).*

Corollary 5.15. *If there exists a single-key* weakly *selective-message message private and weakly succinct SKFE scheme for* P/poly*, there exists a single-key selective-message message private and weakly succinct SKFE scheme for* P/poly.

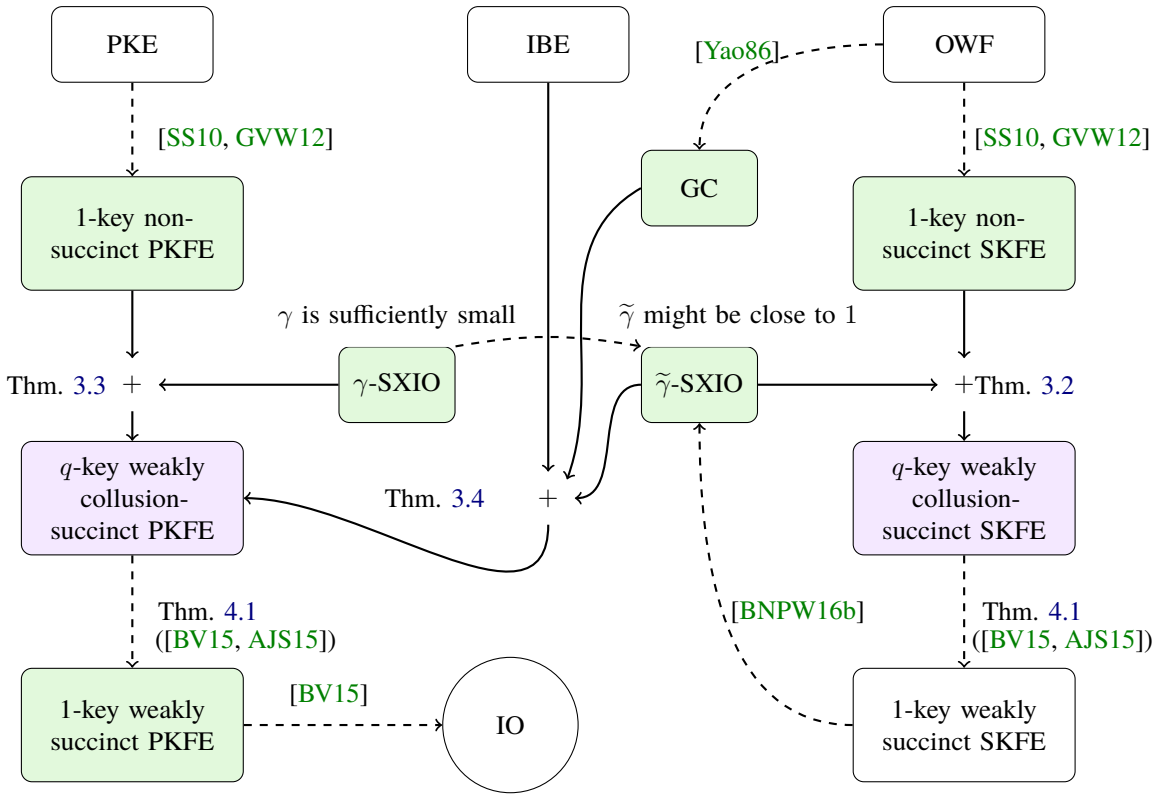

Figure 11: Illustration of our theorems. Dashed lines denote known facts or trivial implications. White boxes denote our ingredients or goal. Purple boxes denote our key schemes. Green boxes denote our intermediate tools. Primitives in rounded boxes should be sub-exponentially-secure to arrive at IO. γ-SXIO denotes SXIO with compression factor γ, which is *sufficiently small* constant of less than 1. $\widetilde{\gamma}$-SXIO denotes SXIO with compression factor $\widetilde{\gamma}$, which is *arbitrary* constant of less than 1. We ignore puncturable PRF and decomposable RE in this figure since they are implied by OWF .

By Theorem 5.1, 5.13, and 5.4, we can obtain the following corollary.

Corollary 5.16. *If there exists one-way function and $\widetilde{\gamma}$-compressing SXIO for $\mathsf{P/poly}$ for a constant $\widetilde{\gamma}$ such that $0 < \widetilde{\gamma} < 1$ ($\widetilde{\gamma}$ might be close to 1), then there exists a γ-compressing SXIO for $\mathsf{P/poly}$ for an arbitrarily small constant γ such that $0 < \gamma < 1$.*

By Theorem 5.9, 5.10, 5.13, and 5.4, we can obtain the following corollary.

Corollary 5.17. *A constant-arity MIFE for $\mathsf{P/poly}$ is equivalent to a $\widetilde{\gamma}$-compressing SXIO such that $0 < \widetilde{\gamma} < 1$.*

By Theorem 5.9, 5.10, 5.13, and 5.4, we can obtain the following corollary.

Corollary 5.18. *A constant-arity MIFE for $\mathsf{P/poly}$ is equivalent to one-way function and $\widetilde{\gamma}$-SXIO such that $0 < \widetilde{\gamma} < 1$.*

By Theorem 5.11, 5.12, and 5.4, we can obtain the following corollary.

Corollary 5.19. *A single-key weakly succinct SKFE for $\mathsf{P/poly}$ is equivalent to one-way function and output-compact updatable randomized encoding with an unbounded number of updates.*

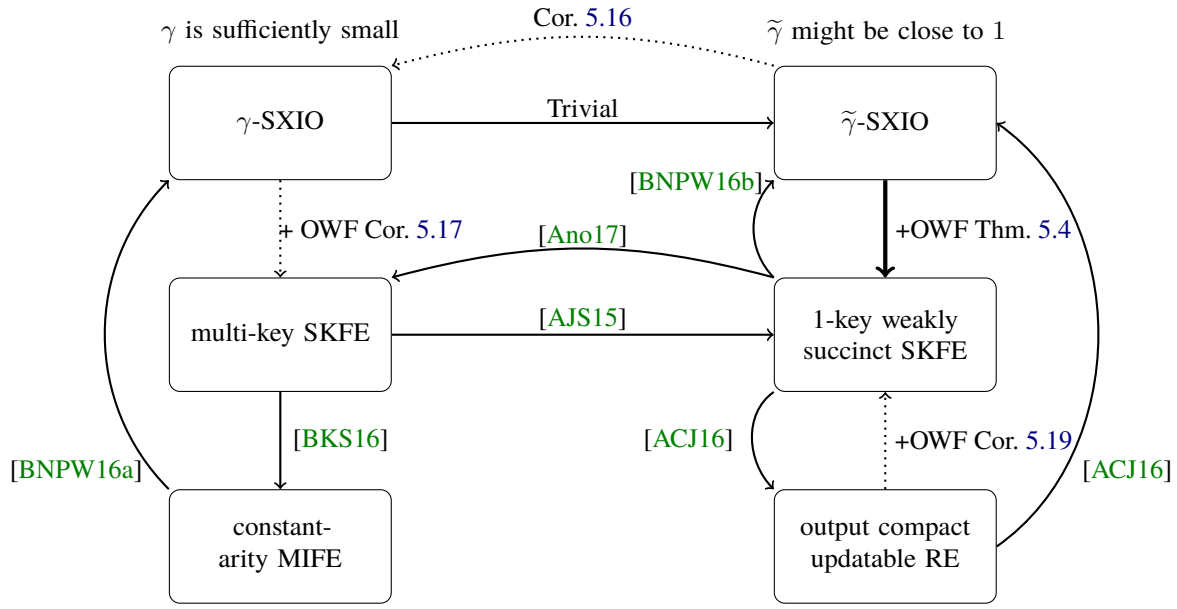

Figure 12: Illustration of our corollaries. The thick line denotes our first main theorem. Solid lines denote known implications. Dashed lines denote our corollaries. γ-SXIO denotes SXIO with compression factor γ, which is *sufficiently small* constant of less than 1. $\widetilde{\gamma}$-SXIO denotes SXIO with compression factor $\widetilde{\gamma}$, which is *arbitrary* constant of less than 1.

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In Dodis and Nielsen [DN15], pages 528–556. (Cited on page 1.)

[ACJ16] Prabhanjan Ananth, Aloni Cohen, and Abhishek Jain. Cryptography with updates. Cryptology ePrint Archive, Report 2016/934, 2016. http://eprint.iacr.org/2016/934. (Cited on page 4, 26, 28.)

[ADGM16] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. *IACR Cryptology ePrint Archive*, 2016:1003, 2016. (Cited on page 1.)

[AGIS14] Prabhanjan Vijendra Ananth, Divya Gupta, Yuval Ishai, and Amit Sahai. Optimizing obfuscation: Avoiding Barrington's theorem. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *ACM CCS 14*, pages 646–658. ACM Press, November 2014. (Cited on page 1.)

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006. (Cited on page 2, 13.)

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Gennaro and Robshaw [GR15], pages 308–326. (Cited on page 1, 6, 10.)

[AJS15] Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation from functional encryption for simple functions. Cryptology ePrint Archive, Report 2015/730, 2015. http://eprint.iacr.org/2015/730. (Cited on page 3, 6, 23, 27, 28.)

[Ano17] Anonymous. From single-key to collusion-resistant secret-key functional encryption by leveraging succinctness. *Unpublished manuscript*, 2017. (Cited on page 3, 4, 26, 28.)

[AS15] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Guruswami [Gur15], pages 191–209. (Cited on page 4.)

[AS16] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. *IACR Cryptology ePrint Archive*, 2016:1097, 2016. (Cited on page 1.)

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Heidelberg, March 2014. (Cited on page 7, 8.)

[BGK+14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In Nguyen and Oswald [NO14], pages 221–238. (Cited on page 1.)

[BKS16] Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In Fischlin and Coron [FC16b], pages 852–880. (Cited on page 4, 25, 26, 28.)

[BMSZ16] Saikrishna Badrinarayanan, Eric Miles, Amit Sahai, and Mark Zhandry. Post-zeroizing obfuscation: New mathematical tools, and the case of evasive circuits. In Fischlin and Coron [FC16b], pages 764–791. (Cited on page 1.)

[BNPW16a] Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From cryptomania to obfustopia through secret-key functional encryption. In Hirt and Smith [HS16], pages 391–418. (Cited on page 1, 2, 3, 4, 5, 9, 14, 25, 26, 28.)

[BNPW16b] Nir Bitansky, Ryo Nishimaki, Alain Passelègue, and Daniel Wichs. From cryptomania to obfustopia through secret-key functional encryption. Cryptology ePrint Archive, Report 2016/558, 2016. http://eprint.iacr.org/2016/558. (Cited on page 4, 25, 27, 28.)

[BR14] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 1–25. Springer, Heidelberg, February 2014. (Cited on page 1.)

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In Dodis and Nielsen [DN15], pages 306–324. (Cited on page 10.)

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Guruswami [Gur15], pages 171–190. (Cited on page 1, 2, 6, 12, 23, 27.)

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Heidelberg, December 2013. (Cited on page 7, 8.)

[CFL+16] Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, Brice Minaud, and Hansol Ryu. Cryptanalysis of the new CLT multilinear map over the integers. In Fischlin and Coron [FC16a], pages 509–536. (Cited on page 1.)

[CGH+15] Jean-Sébastien Coron, Craig Gentry, Shai Halevi, Tancrède Lepoint, Hemanta K. Maji, Eric Miles, Mariana Raykova, Amit Sahai, and Mehdi Tibouchi. Zeroizing without low-level zeroes: New MMAP attacks and their limitations. In Gennaro and Robshaw [GR15], pages 247–266. (Cited on page 1.)

[CGH16] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. Cryptology ePrint Archive, Report 2016/998, 2016. http://eprint.iacr.org/2016/998. (Cited on page 1.)

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 3–12. Springer, Heidelberg, April 2015. (Cited on page 1.)

[CHN+16] Aloni Cohen, Justin Holmgren, Ryo Nishimaki, Vinod Vaikuntanathan, and Daniel Wichs. Watermarking cryptographic capabilities. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 1115–1127. ACM Press, June 2016. (Cited on page 1.)

[CLLT16] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. *IACR Cryptology ePrint Archive*, 2016:1011, 2016. (Cited on page 1.)

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 476–493. Springer, Heidelberg, August 2013. (Cited on page 1.)

[DN15] Yevgeniy Dodis and Jesper Buus Nielsen, editors. *TCC 2015, Part II*, volume 9015 of *LNCS*. Springer, Heidelberg, March 2015. (Cited on page 28, 29, 30.)

[FC16a] Marc Fischlin and Jean-Sébastien Coron, editors. *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*. Springer, Heidelberg, May 2016. (Cited on page 29, 30, 31.)

[FC16b] Marc Fischlin and Jean-Sébastien Coron, editors. *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*. Springer, Heidelberg, May 2016. (Cited on page 29.)

[FRS16] Rex Fernando, Peter M. R. Rasmussen, and Amit Sahai. Preventing clt zeroizing attacks on obfuscation. *IACR Cryptology ePrint Archive*, 2016:1070, 2016. (Cited on page 1.)

[GGG+14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Nguyen and Oswald [NO14], pages 578–602. (Cited on page 4.)

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 1–17. Springer, Heidelberg, May 2013. (Cited on page 1.)

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013. (Cited on page 1.)

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In Dodis and Nielsen [DN15], pages 498–527. (Cited on page 1.)

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984. (Cited on page 7, 8.)

[GMM+16] Sanjam Garg, Eric Miles, Pratyay Mukherjee, Amit Sahai, Akshayaram Srinivasan, and Mark Zhandry. Secure obfuscation in a weak multilinear map model. In Hirt and Smith [HS16], pages 241–268. (Cited on page 1.)

[GR15] Rosario Gennaro and Matthew J. B. Robshaw, editors. *CRYPTO 2015, Part I*, volume 9215 of *LNCS*. Springer, Heidelberg, August 2015. (Cited on page 28, 29.)

[GS16] Sanjam Garg and Akshayaram Srinivasan. Single-key to multi-key functional encryption with polynomial loss. In Hirt and Smith [HS16], pages 419–442. (Cited on page 2, 6, 11, 25.)

[Gur15] Venkatesan Guruswami, editor. *56th FOCS*. IEEE Computer Society Press, October 2015. (Cited on page 28, 29.)

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012. (Cited on page 4, 6, 14, 27.)

[HJ16] Yupu Hu and Huiwen Jia. Cryptanalysis of GGH map. In Fischlin and Coron [FC16a], pages 537–565. (Cited on page 1.)

[HS16] Martin Hirt and Adam D. Smith, editors. *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*, volume 9986 of *Lecture Notes in Computer Science*, 2016. (Cited on page 29, 30, 31.)

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000. (Cited on page 2.)

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 13*, pages 669–684. ACM Press, November 2013. (Cited on page 7, 8.)

[KS17] Ilan Komargodski and Gil Segev. From minicrypt to obfustopia via private-key functional encryption. *IACR Cryptology ePrint Archive*, 2017:080, 2017. (Cited on page 3, 6.)

[Lin16a] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In Fischlin and Coron [FC16a], pages 28–57. (Cited on page 1.)

[Lin16b] Huijia Lin. Indistinguishability obfuscation from DDH on 5-linear maps and locality-5 PRGs. *IACR Cryptology ePrint Archive*, 2016:1096, 2016. (Cited on page 1.)

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In Hirt and Smith [HS16], pages 443–468. (Cited on page 2, 6, 10, 25.)

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part II*, volume 9615 of *LNCS*, pages 447–462. Springer, Heidelberg, March 2016. (Cited on page 1.)

[LT17] Huijia Lin and Stefano Tessaro. Indistinguishability obfuscation from bilinear maps and block-wise local PRGs. *IACR Cryptology ePrint Archive*, 2017:250, 2017. (Cited on page 6.)

[LV16] Huijia Lin and Vinod Vaikuntanathan. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. Cryptology ePrint Archive, Report 2016/795, 2016. http://eprint.iacr.org/2016/795. (Cited on page 1.)

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over GGH13. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 629–658. Springer, Heidelberg, August 2016. (Cited on page 1.)

[NO14] Phong Q. Nguyen and Elisabeth Oswald, editors. *EUROCRYPT 2014*, volume 8441 of *LNCS*. Springer, Heidelberg, May 2014. (Cited on page 29, 30.)

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 500–517. Springer, Heidelberg, August 2014. (Cited on page 1.)

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM CCS 10*, pages 463–472. ACM Press, October 2010. (Cited on page 4, 5, 6, 27.)

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014. (Cited on page 1, 5, 7.)

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th FOCS*, pages 162–167. IEEE Computer Society Press, October 1986. (Cited on page 2, 13, 27.)

[Zim15] Joe Zimmerman. How to obfuscate programs directly. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 439–467. Springer, Heidelberg, April 2015. (Cited on page 1.)