Simple Generic Constructions of Succinct Functional Encryption

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

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

Abstract

We propose simple generic constructions of succinct functional encryption. Our key tool is exponentially-efficient indistinguishability obfuscator (XIO), which is the same as indisitnguishability obfuscator (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.

Our schemes do not rely on any number theoretic or lattice assumptions such as decisional Diffie-Hellman and learning with errors assumpotions. Moreover, all security reductions incur only polynomial security loss. Known constructions of weakly succinct SKFE or PKFE from XIO with polynomial security loss rely on number theoretic or lattice assumptions.

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 new 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.

Keywords: Functional Encryption, Succinctness, Indistinguishability Obfuscation.

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

Contents

1 Introduction

1.1 Background

In cryptography, it is one of major research topics to construct more complex cryptographic primitives from simpler ones in a *generic* way. Here, "generic" means that we use only general cryptographic tools such as one-way function and public-key encryption. For such a generic construction, we do not use any specific or concrete algebraic assumptions such as the factoring, decisional Diffie-Hellman (DDH), learning with errors (LWE) assumptions. Generic constructions are useful in cryptography because they do not rely on any specific structure of underlying primitives. It means that even if a specific number theoretic assumption is broken, say the DDH, a generic construction based on public-key encryption is still secure since there are many instantiations of public-key encryption from other assumptions. Moreover, generic constructions are useful to deeply understand the nature of cryptographic primitives.

Many generic constructions have been proposed. For example, one-way functions imply secure digital signature [NY89, Rom90], pseudo-random function (PRF) [GGM86], and many other primitives. However, we understand little of how to construct functional encryption [BSW11, O'N10] in a generic way despite its usefulness as explained below.

Functional encryption is a generalization of public-key encryption and 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. Functional encryption is a versatile cryptographic primitive since it enables us to achieve not only fine-grained access control systems over encrypted data but also indistinguishability obfuscation (IO) [BGI+12, GGH+13b, AJ15, BV15].

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 softwares 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]. We basically focus on functional encryption and IO for all circuits in this study.

Many concrete functional encryption and IO constructions have been proposed since the celebrating invention of a candidate graded encoding system by Garg, Gentry, Halevi [GGH13a]. However, regarding designing secure functional encryption and IO, we are still at the "embryonic" stage[1]. A few candidates of graded encoding schemes have been proposed [GGH13a, CLT13, GGH15]. However, basically speaking, all are attacked, and most applications (including functional encryption) that use graded encoding schemes are also insecure [CHL+15, CGH+15, CFL+16, HJ16, MSZ16, ADGM17, CLLT17, CGH17]. As an exception, a few IO constructions are still standing [GMM+16, FRS16][2].

The purpose of this study is that we shed new light on how to achieve functional encryption and IO.

The number of functional keys and the size of encryption circuit. In fact, the hardness of constructing functional encryption depends on certain features of functional encryption such as the number of issuable functional keys and ciphertexts and the size of encryption circuit.

We say "single-key" if only one functional key can be issued. We also say q-key or bounded collusion-resistant when a-priori bounded q functional keys can be issued. If q is an a-priori unbounded polynomial, then we say "collusion-resistant". It is known that a single-key secret-key and public-key functional encryption (SKFE and PKFE) are constructed from standard one-way function and public-key encryption, respectively [SS10]. It is also known that a bounded collusion-resistant PKFE (resp. SKFE) is constructed from public-key encryption (resp. one-way function) and pseudo-random generator computed by polynomial degree circuits [GVW12]. However, it is not known whether collusion-resistant functional encryption is constructed without expensive cryptographic tools such as graded encoding systems [GGH13a, CLT13, GGH15] or IO [GGH13a].

It is also known that we can construct collusion-resistant PKFE from single-key weakly succinct PKFE [GS16, LM16]. The notion of succinctness for functional encryption schemes [AJ15, BV15][3] means the size of encryption circuit is independent of the function-size. Weak succinctness means the size of the encryption circuit is

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

$s^\gamma \cdot \mathrm{poly}(\lambda, n)$ 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 $0 < \gamma < 1$. The results of Garg and Srinivasan [GS16] and Li and Micciancio [LM16] mean that we can arbitrarily increase the number of issuable functional keys by using succinctness.

The succinctness of functional encryption is also key feature to achieve IO. Anath and Jain [AJ15] and Bitansky and Vaikuntanathan [BV15] show that a sub-exponentially secure single-key weakly succinct PKFE implies IO. However, it is not known whether (single-key) weakly succinct functional encryption is constructed without graded encoding systems or IO.

Moreover, succinct SKFE and PKFE are constructed from collusion-resistant SKFE and PKFE, respectively [AJS15].

These facts indicate that it is a challenging task to achieve either collusion-resistance or succinctness.

Running time of obfuscator. Not only the encryption-time of functional encryption but also the size of obfuscated circuits and the running time of obfuscators are also important measures.

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 LWE problem is hard, then there exists single-key weakly succinct PKFE (and IO if sub-exponential security is additionally assumed).

Bitansky, Nishimaki, Passelègue, and Wichs [BNPW16a] extend the notion of XIO and define strong XIO (SXIO). If the running time of the obfuscator is $\mathrm{poly}(\lambda, |C|) \cdot 2^{\gamma n}$, then we say it is SXIO. Bitansky *et al.* show that sub-exponentially secure SXIO and public-key encryption imply IO. In addition, they prove that single-key weakly succinct PKFE is constructed from SXIO, public-key encryption, and weak PRF in NC^1, which is implied by the DDH [NR04] or LWE assumptions [BPR12].

Thus, (S)XIO is useful enough to achieve weakly succinct functional encryption and IO. In this study, we discuss more applications of SXIO to functional encryption. In particular, we discuss significantly simple generic constructions of weakly succinct functional encryption by using SXIO.

From SKFE to PKFE. Bitansky *et al.* [BNPW16a] also prove that SXIO is constructed from collusion-resistant SKFE. Thus, we can construct weakly succinct PKFE from a weaker primitive than PKFE by the results of Lin *et al.* and Bitansky *et al.*, though it is not known whether we can construct collusion-resistant SKFE from standard cryptographic primitives.

The works of Lin *et al.* and Bitansky *et al.* are advancements on how to construct succinct PKFE from weaker primitives. However, they still use the DDH or LWE assumptions to achieve weakly succinct PKFE *with polynomial security loss*. Thus, it is not known whether we can construct weakly succinct PKFE with polynomial security loss from SKFE and public-key encryption in a generic way.

1.2 Our Contributions

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

Main theorem 1 (informal): A single-key weakly succinct PKFE is implied by public-key encryption and SXIO with a *sufficiently small* compression factor.

Main theorem 2 (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.

Main theorem 3 (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.

We highlight that all these new theorems incur *only polynomial* security loss and *do not rely on any specific number theoretic or lattice assumption*. These are advantages over the constructions of Lin *et al.* and Bitansky *et al.* [LPST16, BNPW16a]. We explain details of our results below.

Implication of first and second theorems. There are transformations from a single-key weakly succinct PKFE scheme to a collusion-resistant one with polynomial security loss [GS16, LM16]. Thus, by combining the first or second theorems with the transformation, we obtain two collusion-resistant PKFE schemes *with polynomial security loss*. 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 [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 [BNPW16b]. Note that we can also obtain IO constructions from the same building blocks if we assume that they are sub-exponentially secure by using the result of Ananth and Jain[AJ15] or Bitansky and Vaikuntanathan [BV15].

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, our second result indicates that 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 single-key weakly succinct (or collusion-resistant) PKFE and IO.

Advantages over previous constructions. We look closer at previous works for comparison.

Lin *et al.* [LPST16]: They construct single-key weakly succinct PKFE from XIO and single-key succinct PKFE for *Boolean* circuits. It is known that a single key succinct PKFE for Boolean circuits is constructed from the LWE assumption [GKP+13].

Both their construction and ours are generic constructions using (S)XIO. However, their construction additionally needs single-key succinct PKFE for Boolean circuits. We have only one instantiation of such PKFE based on the LWE assumption while our additional primitives (i.e., public-key encryption and identity-based encryption) can be instantiated based on wide range of assumptions. This is the advantage of our construction over that of Lin *et al.*

Bitansky *et al.* [BNPW16a]: They construct single-key weakly succinct PKFE from SXIO and public-key encryption with $2^{O(d)}$ security loss where d is the depth of a circuit. They introduce decomposable garbled circuit, which is an extension of Yao's garbled circuit [Yao86], to achieve succinctness [BNPW16a]. Decomposable garbled circuit is implied by one-way function. However, it has two disadvantages. One is that it incurs the $2^{O(d)}$ security loss. The other is that its security proof is complex.

When we construct single-key weakly succinct (or collusion-resistant) PKFE only with *polynomial security loss*, the exponential security loss in the depth of circuits is a big issue. Thus, Bitansky *et al.* need weak PRF in NC^1 to achieve single-key weakly succinct (or collusion-resistant) PKFE with polynomial security loss due to the $2^{O(d)}$ security loss [BNPW16a, Section 5.3][4]. If our goal is constructing IO, then the $2^{O(d)}$ security loss is not an issue in the sense that we need sub-exponential security of PKFE to achieve IO [BV15, AJ15], and we can cancel the $2^{O(d)}$ security loss by complexity leveraging.

Decomposable garbled circuit is a useful tool for Bitansky *et al.*'s construction. However, it is not easy to understand the security proof. Our unified design strategy significantly simplifies a construction of single-key weakly succinct PKFE based on SXIO. In fact, our constructions use decomposable *randomized encoding* [IK00, AIK06], but decomposable randomized encoding is a simple tool and *does not incur* $2^{O(d)}$ *security loss*.[5]

Using identity-based encryption. We show that we can relax the requirements on SKFE to achieve PKFE and IO if we are allowed to use identity-based encryption.

Our construction of PKFE using identity-based encryption needs SXIO with compression factor slightly smaller than 1 that is implied by single-key (wealy) succinct SKFE while the constructions using public-key encryption need SXIO with sufficiently small compression factor that is implied by collusion-resistant SKFE. It is not known whether single-key (weakly) succinct SKFE implies collusion-resistant SKFE 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

[4] They use a bootstrapping technique by Ananth *et al.* [ABSV15], which transforms functional encryption for NC^1 into one for $\mathsf{P/poly}$.

[5] See Remark 2.8 in Section 2 for more details on the difference between decomposable garbled circuit and decomposable randomized encoding.

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.

Readers who are familiar with the construction of Bitansky *et al.* might think the second theorem is easily obtained from the result of Bitansky *et al.*, 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 then transform it into a single-key weakly succinct PKFE scheme for non-Boolean functions by using SXIO again. Therefore, 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*, and due to this nested use, it still needs SXIO with sufficiently small compression factor.

Thus, the advantages of our single-key weakly succinct PKFE schemes over Bitansky *et al.*'s construction are as follows:

- Our single-key weakly succinct PKFE scheme does not incur $2^{O(d)}$ security loss thus does not need weak PRF in NC^1 (implied by the DDH or LWE assumptions) to support all circuits.
- Our PKFE schemes and their proofs are much simpler.
- We can use single-key weakly succinct SKFE instead of collusion-resistant SKFE (if we use identity-based encryption instead of public-key encryption).

Komargodski and Segev [KS17]: Komargodski and Segev construct IO for *circuits with inputs of poly-logarithmic length and sub-polynomial size* from a quasi-polynomially secure and collusion-resistant SKFE scheme for P/poly. They also construct PKFE for *circuits with inputs of poly-logarithmic length and sub-polynomial size* from a quasi-polynomially secure and collusion-resistant SKFE scheme for P/poly and *sub-exponentially secure* one-way function. Their reduction incurs super-polynomial security loss. Thus, the advantages of our single-key weakly succinct PKFE schemes and IO over Komargodski and Segev's construction are as follows:

- Our PKFE schemes support all circuits. (When constructing IO by combining previous results [AJ15, BV15], the construction also support all circuits.)
- We can use single-key weakly succinct SKFE instead of collusion-resistant SKFE (if we use identity-based encryption)
- Our PKFE schemes are with polynomial security loss and do not need sub-exponentially secure one-way function (though we additionally use a public-key primitive).

We summarize differences between these previous constructions of single-key weakly succinct (or collusion-resistant) PKFE schemes and ours in Table 1.

	ingredients for 1-key weakly succinct (or collusion-resistant) PKFE		supported circuits
[LPST16]	1-key weakly succinct SKFE,	LWE	P/poly
[BNPW16a]	collusion-resistant SKFE, PKE,	dGC, PRF in NC^1 (DDH or LWE)	P/poly
[KS17]	collusion-resistant SKFE,	sub-exponentially secure OWF	$\mathcal{C}^{\text{sub-poly}}_{\log\text{-input}}$
1st thm.	collusion-resistant SKFE, PKE,	dRE	P/poly
2nd thm.	1-key weakly succinct SKFE, IBE,	GC, dRE	P/poly

Table 1: Comparison with previous constructions. OWF, PKE, IBE, GC, dGC, and dRE denote one-way function, public-key encryption, identity-based encryption, garbled circuit, decomposable garbled circuit, and decomposable randomized encoding, respectively. Underlines denote disadvantages. In "supported circuit" column, $\mathcal{C}^{\text{sub-poly}}_{\log\text{-input}}$ means circuits with inputs of poly-logarithmic length and sub-polynomial size.

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

Implication of third theorem. We can obtain interesting by-products from the third 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 slightly smaller than 1 [BNPW16b].

By-product 2: We show that the existence of output-compact updatable randomized encoding with unbounded number of updates [ACJ17] 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 [ACJ17]. 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 [ACJ17] with one-way function.

1.3 Overview of Our Construction Technique

Our core schemes are q-key weakly collusion-succinct functional encryption schemes for a-priori fixed polynomial q 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 2.20 for more details on succinctness. It is known that weakly collusion-succinct functional encryption is transformed into weakly-succinct one [BV15, AJS15].

We explain our ideas to achieve q-key weakly collusion-succinct functional encryption schemes below.

Our main idea in one sentence. We compress parallelized encryption circuits of a non-succinct scheme based on standard cryptographic primitives 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.[7] 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].

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[8]. 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, the time needed to generate the ciphertext $\mathsf{sxi}\mathcal{O}(\mathsf{E}'[K, x])$ is $\mathrm{poly}(\lambda, |x|, |f|) \cdot q^{\gamma}$ from the efficiency guarantee of SXIO. 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].

Extension to public-key setting. 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 to generate not only a ciphertext but also *a master public-key* to prevent the size of a master public-key linearly depending on q. 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. We give the

[7] In fact, the functional key generation algorithm takes an additional input called index and is stateful. We ignore this issue here. However, in fact, this issue does not matter at all. See Remark 2.16 in Section 2 regarding this issue.

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

simplified description of this setup circuit (denoted by S) below for clarity. For the formal description of S, see Figure 4 in Section 3.2.

// Description of (simplified) S	// Description of (simplified) E''
Hard-Coded Constants: K.	**Hard-Coded Constants**: MPK, x.
Input: $i \in [q]$	**Input:** $i \in [q]$
1. Compute $r^i_{\mathsf{Setup}} \leftarrow \mathsf{F}_K(i)$.	1. Parse $\mathsf{sxi}\mathcal{O}(\mathsf{S}) \leftarrow \mathsf{MPK}$.
2. Compute $(\mathsf{MPK}_i, \mathsf{MSK}_i) \leftarrow \mathsf{Setup}(1^\lambda; r^i_{\mathsf{Setup}})$.	2. Compute $\mathsf{MPK}_i \leftarrow \mathsf{sxi}\mathcal{O}(\mathsf{S})(i)$
3. Return MPK_i.	3. Return $\mathsf{CT}_i \leftarrow \mathsf{Enc}(\mathsf{MPK}_i, x)$.

If we do not use $\mathsf{sxi}\mathcal{O}(\mathsf{S})$ as the master public key, we must use $\{\mathsf{MPK}_i\}_{i\in[q]}$ as the master public-key and embed them in a *public* encryption circuit E'' since we cannot make PRF key K public. This leads to linear dependence on q of the encryption time.

Encryption circuit E'' is almost the same as E' in the SKFE construction except that $\mathsf{MPK} = \mathsf{sxi}\mathcal{O}(\mathsf{S})$ is hardwired to generate a master public-key in an on-line manner. Similarly to the SKFE construction, a ciphertext is $\mathsf{sxi}\mathcal{O}(\mathsf{E}'')$. This incurs two applications of SXIO in a nested manner (i.e., we obfuscate a circuit in which another obfuscated circuit is hard-wired). Although the input space of E'' is $[q]$ and the size of the encryption circuit of the single-key scheme is independent of q, the size of $\mathsf{sxi}\mathcal{O}(\mathsf{E}'')$ polynomially depends on $\mathsf{sxi}\mathcal{O}(\mathsf{S})$. Thus, a better compression factor of SXIO for S is required to ensure the weak collusion-succinctness of the resulting scheme. Such better SXIO is implied by *collusion-resistant* (non-succinct) SKFE [BNPW16a]. See Section 3.2 for details of the efficiency analysis.

Using power of identity-based encryption. To overcome the nested applications of SXIO, we directly construct a q-key weakly collusion-succinct PKFE from SXIO, identity-based encryption, and garbled circuit. 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 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, $\{\mathsf{pk}^j_0, \mathsf{pk}^j_1\}_{j\in[s]}$.

Functional Key: A functional key for f consists of s secret-keys of PKE, $\{\mathsf{sk}^j_{f_j}\}_{j\in[s]}$ where $f = f_1 \dots f_s$ and f_j is a single bit for every $j \in [s]$.

Encryption: A ciphertext of a plaintext x consists of a garbled circuit of U_x and encryptions of $2s$ labels of the garbled circuit under pk^j_b for all $j \in [s]$ and $b \in \{0,1\}$.

Decryption: We obtain labels corresponding to f by using $\{\mathsf{sk}^j_{f_j}\}_{j\in[s]}$ and evaluate the garbled U_x with those labels.

We can replace PKE with an identity-based encryption scheme IBE by using identities in $[s] \times \{0,1\}$. That is, $\{\mathsf{pk}^j_0, \mathsf{pk}^j_1\}_{j\in[s]}$ 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 and plaintext x are 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. Identities consists of $(i, j, f_j) \in [q] \times [s] \times \{0,1\}$ for every $j \in [s]$. 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 $\mathsf{sxi}\mathcal{O}(\widetilde{\mathsf{E}})$ on the input i, 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.[9] In addition, the input space of $\widetilde{\mathsf{E}}$ is just $[q]$ and the garbled circuit part of $\widetilde{\mathsf{E}}$ is independent of q.

[9]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 2.11.

Therefore, the time needed to generate a ciphertext $\mathsf{sxi}\mathcal{O}(\widetilde{\mathsf{E}})$ is sub-linear in q from the efficiency property of SXIO. Thus, the scheme is weakly collusion-succinct.

In fact, this PKFE construction is similar to that of Bitansky *et al.* [BNPW16a], but we do not need decomposable garbled circuit because our goal is achieving weak 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.

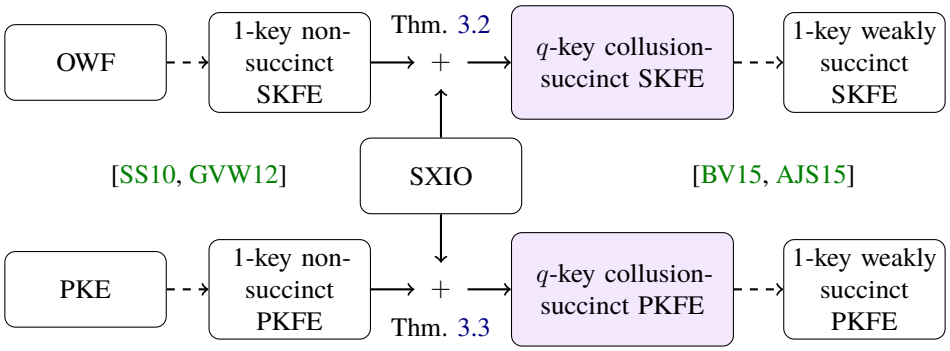

Figure 1: Illustration of our first and third theorems. Dashed lines denote known constructions. Purple boxes denote our core schemes. We ignore puncturable PRF in this figure. It is implied by one-way function.

Uniting pieces. 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, via our weakly collusion-succinct PKFE (resp. SKFE), we can obtain single-key weakly succinct PKFE (resp. SKFE) based on SXIO and standard cryptographic primitives. Figure 1 illustrates our first and third informal theorems.

Concurrent and independent work. Lin and Tessaro [LT17] prove 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.

Their construction 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.

We observe that collusion-resistant SKFE for P/poly is constructed from one-way function and IO for $\omega(\log \lambda)$-bit-input circuits by the construction similar to that of Lin and Tessaro. On the other hand, 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.[10] Thus, our assumptions are milder than theirs to construct collusion-resistant PKFE for P/poly (or single-key weakly succinct PKFE for P/poly).

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 schemes 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.

2 Preliminaries

We now define some notations and cryptographic primitives.

[10] Komargodski and Segev [KS17] show that IO for $O(\mathrm{poly}(\log \lambda))$-bit-input and sub-polynomial size circuits is constructed from collusion-resistant SKFE. However, the construction incurs quasi-polynomial security loss. In addition, it is not clear whether their IO is sufficient for the construction of Lin and Tessaro since it supports only circuits of sub-polynomial size.

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 a 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. PPT stands for probabilistic polynomial time.

- 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 $f(\lambda) \leq \mathsf{negl}(\lambda)$ 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 PPT 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 $\{\mathsf{F}_K : \{0,1\}^{\ell_1} \to \{0,1\}^{\ell_2} \mid K \in \{0,1\}^\lambda\}$ be a family of polynomially computable functions, where ℓ_1 and ℓ_2 are some polynomials of λ. We say that F is a pseudo-random function (PRF) family if for any PPT distinguisher $\mathcal{A}$, it holds 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 F is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any PPT $\mathcal{A}$ 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 $\mathsf{PPRF} = (\mathsf{PRF.Gen}, \mathsf{F}, \mathsf{Punc})$ 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 polynomial-size set $S \subseteq \mathsf{D}$ with $S = \{x_1, \ldots, x_{k(\lambda)}\}$ and any PPT distinguisher $\mathcal{A}$, it holds 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 PPRF is δ-secure, for some concrete negligible function $\delta(\cdot)$, if for any PPT $\mathcal{A}$ the above indistinguishability gap is smaller than $\delta(\lambda)^{\Omega(1)}$.

The Goldwasser-Goldreich-Micali tree-based construction of PRFs [GGM86] 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 ([GGM86, 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 (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)* $\{L_{j,\alpha}\}_{j\in[n],\alpha\in\{0,1\}}$.

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

A garbling scheme is required to satisfy the following properties.

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

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* $(\widetilde{C}, \{L_{j,\alpha}\}_{j\in[n],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, C)$ *and returns* $(\widetilde{C}, \{L_{j,x_j}\}_{j\in[n]})$ *to* $\mathcal{A}$*. Otherwise, the challenger returns* $(\widetilde{C}, \{L_{x_j}\}_{j\in[n]}) \leftarrow \mathsf{GC.Sim}(1^\lambda, 1^{|C|}, C(x))$.
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* $\mathsf{GC.Sim}$*, for any PPT* $\mathcal{A}$*, it holds that*

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

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

Theorem 2.5 ([Yao86]). *If there exists one-way function, there exists a secure garbling scheme for all poly-size circuits.*

Definition 2.6 (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 an input* $\widehat{f}(x; r)$ *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* $\mathsf{RE.Sim}$ *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* $\mathsf{RE.Sim}$*, for any PPT $\mathcal{A}$, it holds 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 PPT $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Theorem 2.7 ([Yao86, AIK06]). *If there exists one-way function, there exists a semantically secure decomposable randomized encoding for all poly-size circuits.*

Remark 2.8 (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 an exponential security loss in the depth of circuits 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].

Definition 2.9 (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.*

CPA-security *We define the experiment $\mathsf{Expt}^{\mathsf{ske}}_{\mathcal{A}}(1^\lambda, b)$ between an adversary $\mathcal{A}$ and a challenger as follows. 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. *$\mathcal{A}$ may make polynomially many encryption queries adaptively. If $\mathcal{A}$ 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 CPA-secure if, for any PPT adversary $\mathcal{A}$, it holds 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 PPT $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Definition 2.10 (Plain Public-key Encryption). *Let $\mathcal{M}$ be a message space. A public-key encryption scheme for $\mathcal{M}$ is a tuple of algorithms* $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *where:*

- $\mathsf{KG}(1^\lambda)$ *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 $\bot$.*

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

CPA-security *We define the experiment $\mathsf{Expt}^{\mathsf{pke}}_{\mathcal{A}}(1^\lambda, b)$ between an adversary $\mathcal{A}$ and challenger as follows.*

1. *The challenger runs* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KG}(1^\lambda)$*, 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 CPA-secure if, for any PPT adversary* $\mathcal{A}$*, it holds 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 PPT* $\mathcal{A}$ *the above advantage is smaller than* $\delta(\lambda)^{\Omega(1)}$*.*

Definition 2.11 (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* $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *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 length* ℓ *is at most* $|\mathcal{ID}|^\alpha \mathrm{poly}(\lambda, \ell)$ *where* α *is a constant such that* $0 < \alpha < 1$*.*

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 defined by Bitansky *et al.* [BNPW16a] which is a weaker variant of standard selective-security in the sense that the definition requires an adversary to declare challenge messages along with the challenge identity at the beginning of the security game.

Definition 2.12 (Selectively-Secure Identity-Based Encryption). *A tuple of algorithms* $\mathsf{IBE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *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 IBE *is selectively-secure if, for any PPT* $\mathcal{A}$*, it holds 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 PPT* $\mathcal{A}$ *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 (SKFE)

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. See Remark 2.16 for details.

Definition 2.13 (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, $\mathcal{I} := [q_k(\lambda)]$ an index space where q_k is a fixed polynomial, 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* CT.
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{CT})$ *takes as input sk_f for $f \in \mathcal{F}$ and* CT *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 2.14 (Selective-Message Message Privacy). *Let* SKFE *be an iSKFE 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 an a-priori unbounded 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 function queries at most $|\mathcal{I}| = q_k$ times. For the ℓ-th key query $f_\ell \in \mathcal{F}$ from $\mathcal{A}$, the challenger generates $sk_{f_\ell} \leftarrow \mathsf{iKG}(\mathsf{MSK}, f_\ell, \ell)$ and returns sk_{f_ℓ} to $\mathcal{A}$.*
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-selective-message message private (or selectively secure for short) if for any PPT $\mathcal{A}$, it holds 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, δ)-selective-message message private, for some concrete negligible function $\delta(\cdot)$, if for any PPT $\mathcal{A}$ the above advantage is smaller than $\delta(\lambda)^{\Omega(1)}$.*

Remark 2.15 (Regarding the number of key queries). 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.

Remark 2.16 (Regarding an index for algorithm iKG). Our definitions of functional encryptions slightly deviates from the standard ones (e.g., 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 single-key PKFE. For a single-key scheme, $|\mathcal{I}| = 1$ and we do not need such an index. Index based bounded collusion-resistant functional encryption schemes are just intermediate tools in this study. In fact, such an index have been introduced by Li and Micciancio in the context of PKFE [LM16].[11]

Public-Key Functional Encryption (PKFE)

We next review the definition of PKFE. Similarly to SKFE, we introduce the index based variant of definition here.

Definition 2.17 (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, $\mathcal{I} := [q_k(\lambda)]$ an index space where q_k is a fixed polynomial, 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* CT.
- $\mathsf{Dec}(\mathsf{sk}_f, \mathsf{CT})$ *takes as input sk_f for $f \in \mathcal{F}$ and c, and outputs $y \in \mathcal{Y}$, or $\perp$.*

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

Definition 2.18 (Selectively-Security). *We say that a tuple of algorithms* $\mathsf{PKFE} = (\mathsf{Setup}, \mathsf{iKG}, \mathsf{Enc}, \mathsf{Dec})$ *is a* 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 a 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 function queries at most $|\mathcal{I}| = q_k$ times, where it sends a function $f_\ell \in \mathcal{F}$ to the challenger. The challenger checks that $f_\ell(x^*_0) = f_\ell(x^*_1)$. If the check fails, then the challenger returns $\perp$. Otherwise, the challenger responds with $\mathsf{sk}_{f_\ell} \leftarrow \mathsf{iKG}(\mathsf{MSK}, f_\ell, \ell)$ for the ℓ-th query f_ℓ.*
4. *$\mathcal{A}$ outputs a guess b' for b.*
5. *The experiment outputs 1 if $b = b'$; otherwise 0.*

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

We also introduce an weaker variant of selective-security.

[11] The security definition of Li and Micciancio for index based functional encryption and ours is slightly different. Their definition allows an adversary to use indices for key generation in an arbitrary order. On the other hand, our definition does not allow it. The difference comes from the fact that their goal is constructing collusion-resistant functional encryption while our goal is constructing single-key functional encryption. By restricting an adversary to use indices successively from one, we can describe security proofs more simply.

Definition 2.19 (Weakly-Selective Security [GS16]). *We say that a tuple of algorithms* $\mathsf{PKFE} = (\mathsf{Setup}, \mathsf{iKG}, \mathsf{Enc}, \mathsf{Dec})$ *is an* 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* $x_0^*, x_1^* \in \mathcal{M}$ *and functions* $(f_1, \ldots, f_{q_k}) \in \mathcal{F}^{q_k}$ *to the challenger, where* q_k *is a polynomial of* λ *such that* $q_k \leq |\mathcal{I}|$.
2. *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* $\mathsf{sk}_{f_\ell} \leftarrow \mathsf{Key}(\mathsf{MSK}, f_\ell, \ell)$ *for all* $\ell \in [q_k]$. *The challenger gives* $(\mathsf{MPK}, \mathsf{CT}^*, \mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_{q_k}})$ *to* $\mathcal{A}$.
3. $\mathcal{A}$ *outputs a guess* b' *for* b.
4. *The experiment outputs* 1 *if* $b = b'$ *and* $f_\ell(x_0^*) = f_\ell(x_1^*)$ *for all* $\ell \in [q_k]$; *otherwise* $\perp$.

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

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

Definition 2.20 (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 the upper bound on the circuit size of functions in* $\mathcal{F}_\lambda$,
- $d(\lambda) = \max_{f \in \mathcal{F}_\lambda} \mathsf{depth}(f)$ *be the upper 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 (that is, the size of the index space of the scheme),* 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. We recall that single-key iPKFE is also single-key PKFE, and vice versa.

Theorem 2.21 ([BV15]). *If there exists a single-key sub-exponentially weakly-selective secure weakly succinct PKFE scheme for* $\mathsf{P/poly}$, *then there exists an indistinguishability obfuscator for* $\mathsf{P/poly}$.

2.4 Indistinguishability Obfuscation

Definition 2.22 (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 PPT distinguisher* D *and 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* $|C_0| = |C_1|$, *it holds that*

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

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

2.5 Strong Exponentially-Efficient Indistinguishability Obfuscation

Definition 2.23 (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 $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if it satisfies the functionality and indistinguishability in Definition 2.22 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 iSKFE and iPKFE 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][12]). *If there exists a δ-secure one-way function, then there exists a $(1,\delta)$-selectively-secure and non-succinct SKFE scheme for 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 P/poly.*

See Appendix A for more details of single-key non-succinct schemes.

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 2.20.

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

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

Our Construction. The construction of an iSKFE scheme qFE whose index space is $[q]$ from a single-key SKFE and SXIO is as follows, where q is an a-priori fixed polynomial of λ. 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, where $\widetilde{\gamma}$ is a constant such that $0 < \widetilde{\gamma} < 1$.

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

- 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 the circuit $\widehat{\mathsf{CT}}$ on input i, that is $\mathsf{CT}_i \leftarrow \widehat{\mathsf{CT}}(i)$.
- 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 keys K, K', and 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 $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x; r'_i)$.

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 P/poly and δ-secure $\widetilde{\gamma}$-compressing SXIO for 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 P/poly with compression factor γ' such that $0 < \widetilde{\gamma} < \gamma' < 1$, where q is an a-priori fixed 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. j is a variable that takes a value in $[q_m]$, where q_m is the number of message pairs an adversary queries. Note that the value of j does not affect the size of $\mathsf{E}^{(j)}$. We need to set $\mathsf{pad} := \max(|\mathsf{E}_{\mathsf{1fe}}|, |\mathsf{E}^{(j)}|)$.

Let, n and s be the size of x and f, respectively. The circuits $\mathsf{E}_{\mathsf{1fe}}$ and $\mathsf{E}^{(j)}$ compute a puncturable PRF over the domain $[q]$, an SKFE master secret key, and may have punctured PRF keys and a hardwired ciphertext. Note that the size of instances of $\mathsf{1FE}$ 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 efficiency guarantee of SXIO, the size of the encryption circuit $\mathsf{qFE.Enc}$ (dominated by generating the obfuscated $\mathsf{E}_{\mathsf{1fe}}$) is

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

where $\widetilde{\gamma}$ and γ' are constants such that $0 < \widetilde{\gamma} < \gamma' < 1$.

Security Proof. Let us assume that the underlying primitives are δ-secure. Let $\mathcal{A}$ be an adversary attacking the selective security of qFE. We define a sequence of hybrid games.

Hyb_0**:** The first game is the original selective security experiment for $b = 0$, that is $\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0)$. 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 $x_0^{(1)}, \ldots, x_0^{(q_m)}$. After that, it also adaptively makes q function queries $\{f_i\}_{i \in [q]}$ 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 2.13 for more details).

$\mathsf{Hyb}_1^{i^*}$**:** Let $i^* \in [q]$. For all $j \in [q_m]$, we generate the challenge ciphertext as obfuscated $\mathsf{E}^{(j)}$ described in Figure 3. In this hybrid game, we set $r_{i^*} \leftarrow \mathsf{F}_K(i^*)$, $K\{i^*\} \leftarrow \mathsf{Punc}(K, i^*)$, $r_{i^*}^{(j)} \leftarrow \mathsf{F}_{K^{(j)}}(i^*)$, $K^{(j)}\{i^*\} \leftarrow \mathsf{Punc}(K^{(j)}, i^*)$, $\mathsf{MSK}_{i^*} \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_{i^*})$, and $\mathsf{CT}_{i^*}^{(j)} \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_{i^*}, x_0^{(j)}; r_{i^*}^{(j)})$ for every $j \in [q_m]$, where $K^{(j)} \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ is randomness for the j-th target ciphertext.

When $i^* = 1$, for all $j \in [q_m]$, the behaviors of $\mathsf{E}_{\mathsf{1fe}}[K, K', x_0^{(j)}]$ and $\mathsf{E}^{(j)}$ are the same since the hard-wired ciphertexts $\mathsf{CT}_1^{(j)}$ in $\mathsf{E}^{(j)}$ is the same as the output of $\mathsf{E}_{\mathsf{1fe}}[K, K', x_0^{(j)}]$ on an input 1. 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.)

[12]More precisely, Gorbunov *et al.* prove that we can construct *adaptively* secure schemes, in which adversaries are allowed to declare a target message pair after the function query phase. However, selective security is sufficient for our purpose.

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 $K\{i^*\}$, $K^{(j)}\{i^*\}$, messages $x_0^{(j)}$, $x_1^{(j)}$, and ciphertext $\mathsf{CT}_{i^*}^{(j)}$.
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. If $i > i^*$, output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x_0^{(j)}; r_i^{(j)})$.
5. If $i < i^*$, output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MSK}_i, x_1^{(j)}; r_i^{(j)})$.

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

$\mathsf{Hyb}_2^{i^*}$**:** 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 of puncturable PRF, it holds that $\mathsf{Hyb}_1^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_2^{i^*}$ for every $i^* \in [q]$.

$\mathsf{Hyb}_3^{i^*}$**:** 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 $\mathsf{CT}_{i^*}^{(j)}$ as the hardwired ciphertext for $\mathsf{E}^{(j)}$. Therefore, for every $i^* \in [q]$, $\mathsf{Hyb}_2^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_3^{i^*}$ follows 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 $r_{i^*}^{(j)} = \mathsf{F}_{K^{(j)}}(i^*)$ for every $j \in [q_m]$. We can show that $\mathsf{Hyb}_3^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_4^{i^*}$ for every $i^* \in [q]$ based on the pseudo-randomness at punctured point of puncturable PRF.

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 for every $i^* \in [q-1]$. Thus, $\mathsf{Hyb}_4^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1^{i^*+1}$ holds for every $i^* \in [q-1]$ due to the indistinguishability of $\mathsf{sxi}\mathcal{O}$. It also holds that $\mathsf{Hyb}_4^{q} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}_{\mathcal{A}}^{\mathsf{sel}}(1^\lambda, 1)$. (Again, we use the security of $\mathsf{sxi}\mathcal{O}$ q_m times to show these indistinguishability.)

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 iPKFE scheme from an SXIO and PKE scheme.

Overview and proof strategy. Before we proceed to details, we give a main idea for our iPKFE scheme.

Analogously to SKFE setting in Section 3.1, to achieve collusion-succinctness, we consider to set a ciphertext as a circuit obfuscated by SXIO that can generate q ciphertexts of a single-key non-succinct scheme. We need to maintain q encryption keys succinctly. In the SKFE setting, we maintain q master secret-keys as one puncturable PRF key. However, we cannot directly use this solution in the PKFE setting. If we do so in the PKFE setting, since the puncturable PRF key should be the master secret-key, an encryptor cannot use it. Thus, we need some mechanism that makes all master public-keys of single-key non-succinct schemes available to an encryptor maintaining them succinctly.

To generate a *succinct* master public-key, we generate a setup circuit (denoted by $\mathsf{S}_{\mathsf{1fe}}$ in our scheme) that outputs i-th master public-key of a single-key non-succinct scheme corresponding to an input i, and obfuscate the circuit by SXIO as explained in Section 1.3. An encryptor embeds $\mathsf{MPK} := \mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}})$ into an encryption circuit and outputs an obfuscation of this encryption circuit as a ciphertext. This encryption circuit is hardwired a plaintext x and can output ciphertexts under all q master public-keys like the encryption circuit in Section 3.1.

Our solution means that we must obfuscate a circuit in which an obfuscated circuit is hardwired (nested applications of SXIO). The nested application still increases the size of a ciphertext. However, if the compression factor of SXIO for $\mathsf{S}_{\mathsf{1fe}}$ is sufficiently small, we can achieve weak collusion-succinctness.

In the security proof, we use the security of a single-key non-succinct scheme to change a ciphertext of x_0 under each master public-key into that of x_1 via the punctured programming approach as the SKFE case. However, in the reduction to the single-key security, a target master public-key should be given from the security experiment. This means that we must embed the target master public-key into the setup circuit instead of generating it in an on-line manner. Thus, we must apply the punctured programming technique to the setup circuit too before the reduction to the single-key security. This is what the first hybrid step in the security proof does. The rest of the proof is almost the same as that of our iSKFE scheme.

Our construction. The construction of an iPKFE scheme qFE whose index space is $[q]$ from an SXIO and public-key encryption scheme is as follows, where q is a fixed polynomial of λ. 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 $\mathsf{S}_{\mathsf{1fe}}[K]$ 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 $\widehat{\mathsf{sk}}_f \leftarrow (i, \mathsf{sk}_f^i)$.

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

- 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 the circuit $\widehat{\mathsf{CT}}$ on input i, that is $\mathsf{CT}_i \leftarrow \widehat{\mathsf{CT}}(i)$.
- 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 $\mathsf{S}_{\mathsf{1fe}}[K]$.

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

Hardwired: circuit $\widehat{\mathsf{MPK}}$, puncturable PRF key K', and message x.
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 the circuit $\widehat{\mathsf{MPK}}$ on input i, that is $\mathsf{MPK}_i \leftarrow \widehat{\mathsf{MPK}}(i)$.
2. Compute $r'_i \leftarrow \mathsf{F}_{K'}(i)$ and output $\mathsf{CT}_i \leftarrow \mathsf{1FE.Enc}(\mathsf{MPK}_i, x; r'_i)$.

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

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 weakly collusion-succinct PKFE for P/poly with compression factor β, where q is an a-priori fixed polynomial of λ, and β is a constant such that $0 < \beta < 1$ specified later.*

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

Padding Parameter. The proof of security relies on the indistinguishability of obfuscated $\mathsf{S}_{\mathsf{1fe}}$ and $\mathsf{S}^*_{\mathsf{1fe}}$ defined in Figures 4 and 5, and that of obfuscated $\mathsf{E}_{\mathsf{1fe}}$ and $\mathsf{E}^*_{\mathsf{1fe}}$ defined in Figure 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 $[q]$ and a key pair of $\mathsf{1FE}$, and may have punctured PRF keys and a master public key hardwired. The circuits $\mathsf{E}_{\mathsf{1fe}}$ and $\mathsf{E}^*_{\mathsf{1fe}}$ run the circuit $\widehat{\mathsf{MPK}}$ and compute a puncturable PRF over domain $[q]$ and a ciphertext of $\mathsf{1FE}$, and may have punctured PRF keys and a hard-wired ciphertext. Note that the size of instances of $\mathsf{1FE}$ 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. To clearly analyze the size of $\mathsf{qFE.Enc}$, we suppose that SXIO used to obfuscate $\mathsf{S}_{\mathsf{1fe}}$ and that used to obfuscate $\mathsf{E}_{\mathsf{1fe}}$ are different.

Let γ' be the compression factor of the SXIO for $\mathsf{S}_{\mathsf{1fe}}$. The input space for $\mathsf{S}_{\mathsf{1fe}}$ is $[q]$. Therefore, by the efficiency guarantee of SXIO, we have

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

Let γ be the compression factor of the SXIO for $\mathsf{E}_{\mathsf{1fe}}$. The input space of $\mathsf{E}_{\mathsf{1fe}}$ is also $[q]$. The size of the encryption circuit $\mathsf{qFE.Enc}$ (dominated by generating 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, by setting γ' as $\gamma' < \frac{1-\gamma}{c}$, we can ensure that $\beta := \gamma + c\gamma' < 1$, that is qFE is weakly collusion-succinct.

Security Proof. Let us assume that the underlying primitives are δ-secure. Let $\mathcal{A}$ be an adversary attacking the selective security of qFE. We define a sequence of hybrid games.

Hyb_0**:** The first game is the original selective security experiment for $b = 0$, that is $\mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 0)$. $\mathcal{A}$ first selects the challenge messages (x^*_0, x^*_1) and receives the master public key $\widehat{\mathsf{MPK}} := \mathsf{sxi}\mathcal{O}(\mathsf{S}_{\mathsf{1fe}}[K])$ and target ciphertext $\mathsf{sxi}\mathcal{O}(\mathsf{E}_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K', x^*_0])$. Next, $\mathcal{A}$ adaptively makes q function queries $f_1, \ldots, f_q$ such that $f_i(x^*_0) = f_i(x^*_1)$ for all $i \in [q]$ and receives functional keys $\widehat{\mathsf{sk}}_{f_1}, \ldots, \widehat{\mathsf{sk}}_{f_q}$. (see Definition 2.17 for more details).

$\mathsf{Hyb}_1^{i^*}$**:** Let $i^* \in [q]$. We generate $\widehat{\mathsf{MPK}}$ as obfuscated $\mathsf{S}^*_{\mathsf{1fe}}$ described in Figure 6. In this hybrid game, we set $r_{i^*} \leftarrow \mathsf{F}_K(i^*)$, $K\{i^*\} \leftarrow \mathsf{Punc}(K, i^*)$ and $(\mathsf{MPK}_{i^*}, \mathsf{MSK}_{i^*}) \leftarrow \mathsf{1FE.Setup}(1^\lambda; r_{i^*})$.

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 in $\mathsf{S}^*_{\mathsf{1fe}}$ is the same as the output of $\mathsf{S}_{\mathsf{1fe}}$ on the input 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 $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1^1$.

$\mathsf{Hyb}_2^{i^*}$**:** The challenge ciphertext is generated by obfuscating $\mathsf{E}^*_{\mathsf{1fe}}$ described in Figure 7. In this hybrid game, we set $r'_{i^*} \leftarrow \mathsf{F}_{K'}(i^*)$, $K'\{i^*\} \leftarrow \mathsf{Punc}(K', i^*)$, $\mathsf{CT}_{i^*} \leftarrow \mathsf{1FE.Enc}(\mathsf{MPK}_{i^*}, x_0^*; r'_{i^*})$, and $\mathsf{MPK}_{i^*} \leftarrow \widehat{\mathsf{MPK}}(i^*)$.

When $i^* = 1$, the behavior of $\mathsf{E}^*_{\mathsf{1fe}}$ is the same as that of $\mathsf{E}_{\mathsf{1fe}}$ since the hard-wired CT_1 in $\mathsf{E}^*_{\mathsf{1fe}}$ is the same as the output of $\mathsf{E}_{\mathsf{1fe}}$ on the input 1. Moreover, both circuits have the same size by padding $\mathsf{pad}_{\mathsf{E}}$. Then, we can use the indistinguishability guarantee of $\mathsf{sxi}\mathcal{O}$ and it holds that $\mathsf{Hyb}_1^1 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_2^1$.

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

Hardwired: puncturable PRF key $K\{i^*\}$ and 1FE master public-key MPK_{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 $\mathsf{S}^*_{\mathsf{1fe}}[K\{i^*\}, \mathsf{MPK}_{i^*}]$. The description depends on i^*, but we use the notion $\mathsf{S}^*_{\mathsf{1fe}}$ instead of $\mathsf{S}^{i^*}_{\mathsf{1fe}}$ for simpler notations.

Encryption Circuit $\mathsf{E}^*_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K'\{i^*\}, x_0^*, x_1^*, \mathsf{CT}_{i^*}](i)$

Hardwired: master public key $\widehat{\mathsf{MPK}}$ (that is an obfuscated circuit), puncturable PRF key $K'\{i^*\}$, messages x_0^*, x_1^*, and ciphertext CT_{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. If $i = i^*$, output CT_{i^*}.
2. Else if compute $r'_i \leftarrow \mathsf{F}_{K'}(i)$ and the circuit $\widehat{\mathsf{MPK}}$ on input i, that is $\mathsf{MPK}_i \leftarrow \widehat{\mathsf{MPK}}(i)$.
   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 $\mathsf{E}^*_{\mathsf{1fe}}[\widehat{\mathsf{MPK}}, K'\{i^*\}, x_0^*, x_1^*, \mathsf{CT}_{i^*}]$. The description depends on i^*, but we use the notion $\mathsf{E}^*_{\mathsf{1fe}}$ instead of $\mathsf{E}^{i^*}_{\mathsf{1fe}}$ for simpler notations.

$\mathsf{Hyb}_3^{i^*}$**:** 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 of puncturable PRF, it holds that $\mathsf{Hyb}_2^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_3^{i^*}$ for every $i^* \in [q]$.

$\mathsf{Hyb}_4^{i^*}$**:** We change CT_{i^*} from $\mathsf{1FE.Enc}(\mathsf{MPK}_{i^*}, x_0^*)$ to $\mathsf{1FE.Enc}(\mathsf{MPK}_{i^*}, x_1^*)$. In $\mathsf{Hyb}_3^{i^*}$ and $\mathsf{Hyb}_4^{i^*}$, we do not need randomness to generate MPK_{i^*} and CT_{i^*}. We just hardwire MPK_{i^*} and CT_i^* into $\mathsf{S}^*_{\mathsf{1fe}}$ and $\mathsf{E}^*_{\mathsf{1fe}}$, respectively. Therefore, for every $i^* \in [q]$, $\mathsf{Hyb}_3^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_4^{i^*}$ follows from the selective security of 1FE under the master public key MPK_{i^*}.

$\mathsf{Hyb}_5^{i^*}$**:** We change r_i^* and r'_{i^*} into $r_{i^*} = \mathsf{F}_K(i^*)$ and $r'_{i^*} = \mathsf{F}_{K'}(i^*)$. We can show $\mathsf{Hyb}_4^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_5^{i^*}$ for every $i^* \in [q]$ based on the pseudo-randomness at punctured point of puncturable PRF.

From the definition of $\mathsf{S}^*_{\mathsf{1FE}}$, $\mathsf{E}^*_{\mathsf{1FE}}$, and $\mathsf{Hyb}_1^{i^*}$, the behaviors of $\mathsf{S}^*_{\mathsf{1FE}}$ and $\mathsf{E}^*_{\mathsf{1FE}}$ in $\mathsf{Hyb}_5^{i^*}$ and $\mathsf{Hyb}_1^{i^*+1}$ are the same. Thus, $\mathsf{Hyb}_5^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1^{i^*+1}$ holds for every $i^* \in [q-1]$ due to the security guarantee of $\mathsf{sxi}\mathcal{O}$. In addition, for $i^* \geq 2$, the behavior of $\mathsf{E}^*_{\mathsf{1fe}}$ does not change between $\mathsf{Hyb}_1^{i^*}$ and $\mathsf{Hyb}_2^{i^*}$. Thus, $\mathsf{Hyb}_1^{i^*} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_2^{i^*}$ holds for every $i^* \in \{2, \cdots, q\}$ due to the security guarantee of $\mathsf{sxi}\mathcal{O}$. It also holds that $\mathsf{Hyb}_5^{q} \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}^{\mathsf{sel}}_{\mathcal{A}}(1^\lambda, 1)$ based on the security guarantee of $\mathsf{sxi}\mathcal{O}$.

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 iPKFE scheme qFE for any fixed polynomial q of λ is based on an SXIO, identity-based encryption scheme in the sense of Definition 2.11[13], and garbled circuit which is implied by a one-way function. Our collusion-succinct iPKFE scheme is *weakly-selective* secure (see Definition 2.19) because we use function descriptions as identities of identity-based encryption, and the selective security of identity-based encryption requires adversaries to submit a target identity at the beginning of the game.

We assume that we can represent every function f by a s bit string $(f[1], \cdots, f[s])$. 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 $[q] \times [s] \times \{0,1,2\}$.

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

- 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 every $j \in [s]$, 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 $\mathsf{CT}_{\mathsf{fe}} \leftarrow \mathsf{sxi}\mathcal{O}(\mathsf{EL}_{\mathsf{gc}}[\mathsf{MPK}_{\mathsf{ibe}}, K, x])$. $\mathsf{EL}_{\mathsf{gc}}$ is defined in Figure 8.

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

- Parse $(i, f, \{\mathsf{SK}^j\}_{j\in[s]}) \leftarrow \mathsf{sk}_f$.
- Compute the circuit $\mathsf{CT}_{\mathsf{fe}}$ on input i, that is $(\widetilde{U}, \{\mathsf{CT}^{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{CT}_{\mathsf{fe}}(i)$.
- For every $j \in [s]$, compute $L_j \leftarrow \mathsf{IBE.Dec}(\mathsf{SK}^j, \mathsf{CT}^{j,f[j]})$.
- Return $y \leftarrow \mathsf{Eval}(\widetilde{U}, \{L_j\}_{j\in[s]})$.

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 $\mathsf{P/poly}$ 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 iPKFE with compression factor β, where q is an a-priori fixed polynomial of λ and β is a constant such that $\widetilde{\gamma} < \beta < 1$ specified later.*

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

[13]Again, we stress that the size of the encryption circuit of an identity-based encryption scheme is $|\mathcal{ID}|^\alpha \cdot \mathrm{poly}(\lambda, \ell)$ where ℓ is the length of plaintext, $\mathcal{ID}$ is the identity-space, and α is a constant such that $0 < \alpha < 1$. Most identity-based encryption schemes based on concrete assumptions have such succinct encryption circuits. In our scheme, $\mathcal{ID}$ is just a polynomial size.

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

Hardwired: public parameter of IBE $\mathsf{MPK}_{\mathsf{ibe}}$, puncturable PRF key K, 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 every $j \in [s]$ 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 $\mathsf{EL}_{\mathsf{gc}}$. $U(\cdot, x)$ is a universal circuit in which x is hardwired as the second input.

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, respectively. 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 $[q]$, $2s$ IBE ciphertexts, and garbled circuit of $U(\cdot, x)$, and may have punctured PRF keys and a hard-wired ciphertext. Note that the size of set S^* of punctured points of PRF in $\mathsf{EL}^*_{\mathsf{gc}}$ is logarithmic in q. Note also that $|\mathcal{ID}| = 2qs$. Thus, due to the efficiency of IBE, it holds that

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

where α is a constant such that $0 < \alpha < 1$.

Weak Collusion-Succinctness. The input space for $\mathsf{EL}_{\mathsf{gc}}$ is $[q]$. Thus, by the efficiency guarantee of SXIO, the size of the encryption circuit $\mathsf{qFE.Enc}$ (dominated by generating an obfuscated $\mathsf{EL}_{\mathsf{gc}}$) is

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

where $\widetilde{\gamma}$ is a constant such that $0 < \widetilde{\gamma} < 1$ and c is some constant.

By using an identity-based encryption scheme whose compression factor α satisfies $\alpha < \frac{1-\widetilde{\gamma}}{c}$, we ensure that $\beta := \widetilde{\gamma} + c\alpha < 1$, that is qFE is weakly collusion succinct.

Security Proof. Let us assume that the underlying primitives are δ-secure. Let $\mathcal{A}$ be an adversary attacking weakly-selective security of qFE. We define a sequence of hybrid games.

Hyb_0**:** The first game is the original weakly-selective security experiment for $b = 0$, that is $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0)$. In this game, $\mathcal{A}$ first selects the challenge messages (x^*_0, x^*_1) and queries q functions $f_1, \ldots, f_q$ such that $f_i(x^*_0) = f_i(x^*_1)$ for all $i \in [q]$. Then $\mathcal{A}$ obtains an encryption of x^*_0, the master public key, and functional keys $\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_q}$. (see Definition 2.19 for more details).

$\mathsf{Hyb}^{i^*}_1$**:** Let $i^* \in [q]$. The challenge ciphertext is generated by obfuscating $\mathsf{EL}^*_{\mathsf{gc}}$ described in Figure 9. In this hybrid game, we set $r^*_{\mathsf{gc}} \leftarrow \mathsf{F}_K(i^*\|1\|2)$, $r^*_{i^*\|j\|\alpha} \leftarrow \mathsf{F}_K(i^*\|j\|\alpha)$ for all $j \in [s]$ and $\alpha \in \{0,1\}$, $K\{S^*\} \leftarrow \mathsf{Punc}(K, S^*)$ where $S^* := \left\{i^*\|1\|2, \{i^*\|j\|\alpha\}_{j\in[s],\alpha\in\{0,1\}}\right\}$, $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0); r^*_{\mathsf{gc}})$, and $\mathsf{CT}^{j,\alpha}_{i^*} \leftarrow \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, (i^*, j, \alpha), L_{j,\alpha}; r^*_{i^*\|j\|\alpha})$ for all $j \in [s]$ and $\alpha \in \{0,1\}$. Hereafter, we use $r^*_{j\|\alpha}$ instead of $r^*_{i^*\|j\|\alpha}$ for ease of notation.

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 $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^1_1$.

Garbling with encrypted labels circuit $\mathsf{EL}^*_{\mathsf{gc}}[\mathsf{MPK}_{\mathsf{ibe}}, K\{S^*\}, x^*_0, x^*_1, (\widetilde{U}^*, \{\mathsf{CT}^{j,\alpha}_{i^*}\}_{j\in[s],\alpha\in\{0,1\}})]$

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

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 $r_{\mathsf{gc}} \leftarrow \mathsf{F}(K, i\|1\|2)$.

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

   If $i < i^*$, 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 every $j \in [s]$ 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 $\mathsf{EL}^*_{\mathsf{gc}}$. The description depends on i^*, but we use the notion $\mathsf{EL}^*_{\mathsf{gc}}$ instead of $\mathsf{EL}^{i^*}_{\mathsf{gc}}$ for simpler notations. $U(\cdot, x)$ is a universal circuit in which x is hardwired as the second input.

$\mathsf{Hyb}^{i^*}_2$**:** 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 $\alpha \in \{0,1\}$. Due to the pseudo-randomness at punctured points of puncturable PRF, it holds that $\mathsf{Hyb}^{i^*}_1 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_2$ for every $i^* \in [q]$.

$\mathsf{Hyb}^{i^*}_3$**:** 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 every $j \in [s]$, 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 which *does not correspond* to the i^*-th function queried by $\mathcal{A}$, we do not encrypt labels of garbled circuit. We do not change $\mathsf{CT}^{j,\overline{f^*}[j]}_{i^*}$ for all $j \in [s]$. 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 it is hard for $\mathcal{A}$ to detect this change. We show $\mathsf{Hyb}^{i^*}_2 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_3$ more formally in Lemma 3.5 by using the selective security of IBE.

Lemma 3.5. *It holds that* $\mathsf{Hyb}^{i^*}_2 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_3$ *for all* $i^* \in [q]$ *if* IBE *is selectively secure.*

Proof. First, we define more hybrid games H_{j^*} for $j^* \in \{0, \cdots, s\}$ as follows.

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

We show that $\mathsf{H}_{j^*-1} \overset{\mathsf{c}}{\approx}_\delta \mathsf{H}_{j^*}$ 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 game of iPKFE, $\mathcal{B}$ runs $\mathcal{A}$ attacking qFE and receives a message pair (x^*_0, x^*_1) and function queries $f_1, \cdots, f_q$. $\mathcal{B}$ 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 $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0))$ 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 $j \leq j^* - 1$, $\mathcal{B}$ 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 $j \geq j^* + 1$ 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 $\mathcal{B}$ does not have to query the challenge identity $(i^* \| j^* \| \overline{f}^*[j^*])$.

Now $\mathcal{B}$ sets all values for $\mathcal{A}$ and sends MPK, $\mathsf{CT}^*_{\mathsf{fe}}$, and $\{\mathsf{SK}_{f_i}\}_{i\in[q]}$ to $\mathcal{A}$. If $\mathcal{B}$ is given $\mathsf{CT}^{j^*,\overline{f}^*[j^*]}_{i^*} = \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^* \| j^* \| \overline{f}^*[j^*]), L_{j^*,\overline{f}^*[j^*]})$, then $\mathcal{B}$ perfectly simulates H_{j^*-1}. If $\mathcal{B}$ is given $\mathsf{CT}^{j^*,\overline{f}^*[j^*]}_{i^*} = \mathsf{IBE.Enc}(\mathsf{MPK}_{\mathsf{ibe}}, i^* \| j^* \| \overline{f}^*[j^*], 0^{\ell(\lambda)})$, then $\mathcal{B}$ perfectly simulates H_{j^*}. Therefore, the advantage of $\mathcal{A}$ between H_{j^*-1} and H_{j^*} is bounded by that of $\mathcal{B}$ attacking 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 $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_0))$ into a simulated output $(\widetilde{U}^*, \{L^*_{j,f^*[j]}\}_{j\in[s]}) \leftarrow \mathsf{Sim.GC}(1^\lambda, y^*))$ where $y^* := f^*(x^*_0) = f^*(x^*_1)$. By the requirement of the game, $f^*(x^*_0) = f^*(x^*_1)$ holds. In this game, $\{L^*_{j,\overline{f}^*[j]}\}_{j\in[s]}$ are not generated since the simulator of GC does not generate them. This is not a problem since for such labels, junk ciphertexts are generated as in $\mathsf{Hyb}^{i^*}_3$. It holds that $\mathsf{Hyb}^{i^*}_3 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_4$ for every $i^* \in [q]$ due to the security of the garbled circuit.

$\mathsf{Hyb}^{i^*}_5$**:** We change the simulated garbled circuit, junk ciphertexts, and punctured PRF keys hardwired into $\mathsf{EL}^*_{\mathsf{gc}}$ back into the real garbled circuit, normal IBE ciphertexts, and un-punctured PRF keys. In this hybrid game, we set $r^*_{\mathsf{gc}} = \mathsf{F}_K(i^* \| 1 \| 2)$, $r^*_{j\|\alpha} = \mathsf{F}_K(i^* \| j \| \alpha)$ for all $j \in [s]$ and $\alpha \in \{0,1\}$, $(\widetilde{U}^*, \{L^*_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, x^*_1); r^*_{\mathsf{gc}})$, 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 $\mathsf{Hyb}^{i^*}_4 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*}_5$ for every $i^* \in [q]$ in a reverse manner.

It holds $\mathsf{Hyb}^{i^*}_5 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^{i^*+1}_1$ for every $i^* \in [q-1]$ by the definition of $\mathsf{EL}^*_{\mathsf{gc}}$ and $\mathsf{sxi}\mathcal{O}$. That is, $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 0) = \mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^1_1 \overset{\mathsf{c}}{\approx}_\delta \cdots \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}^q_5 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 1)$ holds. ■

4 Weak Succinctness from Collusion-Succinctness

In this section, we see a transformation from a q-key weakly collusion-succinct index based functional encryption into a single-key weakly succinct functional encryption. Bitansky and Vaikuntanathan have shown such a transformation [BV15, Proposition IV.1].[14] The key tool for the transformation is decomposable randomized encoding, which is implied by one-way function (see Definition 2.6).

We stress that the transformation in this section is *not new*. The differences between the construction of Bitansky and Vaikuntanathan and ours is that we assume that the underlying weakly collusion-succinct scheme is

[14] Ananth, Jain, and Sahai show a transformation from a *collusion-resistant* non-succinct functional encryption 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.

weakly-selective secure and uses an index for functional key generation. The resulting weakly succinct scheme of our transformation is also weakly-selective secure. Note that the resulting scheme does not need any index for key generation since it is a single-key scheme.

It is known that single-key weakly-selective secure weakly succinct PKFE can be transformed into collusion-resistant and succinct PKFE [GS16]. Moreover, to construct IO by using the theorem by Bitansky and Vaikuntanathan [BV15], a single-key *weakly-selective secure* weakly succinct PKFE scheme is sufficient.

Note that if the maximum size of functions in a function family is fixed, the number of decomposed randomized encodings (denoted by μ) of a function is also fixed. Thus, (μ, δ)-weakly-selective secure (i.e., bounded collusion-resistant) schemes are sufficient for this transformation.

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

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 CPA-secure secret-key encryption scheme, respectively. In the scheme, we use $\mathsf{F} : \{0,1\}^\lambda \times [\rho] \to \{0,1\}$ (note that $\{0,1\}^\lambda \times [\rho]$ is the domain).

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

- 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 $(S_1, \ldots, S_\mu)$ where $S_i \subseteq [\rho]$ and $|S_i| = c$.
- Choose SKE secret key $\mathsf{SK} \leftarrow \{0,1\}^\lambda$. For all $i \in [\mu]$, generate $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}, 0)$, 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 $\mathsf{sk}_f \leftarrow (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_\mu})$.

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

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

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 $\mu = s \cdot \mathrm{poly}_{\mathsf{RE}}(\lambda, n)$ and $\mathrm{poly}_{\mathsf{RE}}$ is a fixed polynomial determined by RE, 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 γ' is a fixed constant such that $\gamma < \gamma' < 1$.*

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

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

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

Figure 10: Description of D_{re}.

Weak Succinctness. To issue one key, we need to issue $1 \cdot \mu = s \cdot \mathrm{poly}_{\mathsf{RE}}(\lambda, n)$ keys of qFE since we consider functions of size s. Thus, we choose $\mu = s \cdot \mathrm{poly}_{\mathsf{RE}}(\lambda, n)$ as the number of issued keys of qFE.

Let $\mathsf{D}_i := \mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$. D_i includes a decryption of SKE, PRF evaluation on the domain $\{0,1\}^\lambda \times [\mu]$, and evaluation of decomposed randomized encoding $\widehat{f}_i$. $|\widehat{f}_i|$ is independent of $|f|$ by the decomposability of RE and $|t|$ and $|\mathsf{CT}^i_{\mathsf{ske}}|$ are bounded by $O(\lambda)$. Moreover, the PRF evaluation is done in time $\mathrm{poly}(\lambda, \log s)$. Thus, the size of D_i is $\mathrm{poly}(\lambda, n, \log s)$. Therefore, the size of encryption circuit sFE.Enc is

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

where γ' is a fixed constant such that $\gamma < \gamma' < 1$.

Security Proof. Let us assume that the underlying primitives are δ-secure. Let $\mathcal{A}$ be an adversary attacking the weakly-selective security of sFE. 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 $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}, 0)$ into $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}, \widehat{f}_i(x^*_0; r_{S_i}))$ for all $i \in [\mu]$. It holds that $\mathsf{Hyb}_0 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_1$ due to the CPA-security of SKE.

Hyb_2**:** We change $\mathsf{CT} \leftarrow \mathsf{qFE.Enc}(\mathsf{MPK}, (0, x^*_0, K, \bot))$ into $\mathsf{CT} \leftarrow \mathsf{qFE.Enc}(\mathsf{MPK}, (1, \bot, \bot, \mathsf{SK}))$.

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 and an secret-key encryption key $\mathsf{SK} \leftarrow \{0,1\}^\lambda$, computes $(\widehat{f}_1, \ldots, \widehat{f}_\mu)$ from f together with $(S_1, \ldots, S_\mu)$, and generates $\mathsf{CT}^i_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(\mathsf{SK}, \widehat{f}_i(x^*_0; r_{S_i}))$ and $\mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$ for all $i \in [\mu]$. Then, $\mathcal{B}$ sends messages $((0, x^*_0, K, \bot), (1, \bot, \bot, \mathsf{SK}))$ as challenge messages and functions $\mathsf{D}_i := \mathsf{D}_{\mathsf{re}}[\widehat{f}_i, S_i, t, \mathsf{CT}^i_{\mathsf{ske}}]$ for all $i \in [\mu]$ 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 $(1, \bot, \bot, \mathsf{SK})$. 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 $e_i \leftarrow \widehat{f}_i(x^*_1; r_{S_i})$ for all $i \in [\mu]$. 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.

Hyb_5**:** This is the same as $\mathsf{Expt}^{\mathsf{sel}^*}_{\mathcal{A}}(1^\lambda, 1)$. We can show $\mathsf{Hyb}_4 \overset{\mathsf{c}}{\approx}_\delta \mathsf{Hyb}_5$ in a reverse manner.

This completes the proof of Theorem 4.1. ■

5 Putting It Altogether

Before summarizing our results, we introduce the following theorems regarding SKFE and SXIO obtained by the results of Brakerski, Komargodski, and Segev [BKS16] and Bitansky *et al.* [BNPW16a, BNPW16b].

Theorem 5.1 ([BKS16, BNPW16a]). *If there exists* (poly, δ)*-selective-message message private and non-succinct SKFE for* $\mathsf{P/poly}$*, then there exists* δ*-secure and* 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* $(1, \delta)$*-selective-message message private and weakly succinct SKFE for* $\mathsf{P/poly}$*, then there exists* δ*-secure and* $\widetilde{\gamma}$*-compressing SXIO for* $\mathsf{P/poly}$ *where* $\widetilde{\gamma}$ *is a constant such that* $1/2 \leq \widetilde{\gamma} < 1$*.*

We also introduce the following result shown by Garg and Srinivasan [GS16] stating that we can transform single-key PKFE into collusion-resistant one strengthening selective security and succinctness.

Theorem 5.3 ([GS16]). *If there exists a* $(1, \delta)$*-weakly-selective 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 Transformation from SKFE to PKFE

By Theorems 3.1, 3.3, 4.1 and 5.1, we obtain the following theorem. We note that Theorem 3.3 requires a sufficiently small compression factor for SXIO.

Theorem 5.4. *If there exists* δ*-secure plain public-key encryption and* (poly, δ)*-selective-message message private and non-succinct SKFE for* $\mathsf{P/poly}$*, then there exists* $(1, \delta)$*-selectively secure and weakly succinct PKFE for* $\mathsf{P/poly}$*.*

From this theorem and Theorem 5.3, we obtain the following corollary stating that collusion-resistant PKFE is constructed from collusion-resistant SKFE if we additionally assume public-key encryption.

Corollary 5.5. *If there exists* δ*-secure plain public-key encryption and* (poly, δ)*-selective-message message private and non-succinct SKFE for* $\mathsf{P/poly}$*, then there exists* (poly, δ)*-selectively secure and succinct PKFE for* $\mathsf{P/poly}$*.*

We stress that the transformations above incur only *polynomial security loss*. Bitansky *et al.* [BNPW16a] show a construction of PKFE based on public-key encryption and collusion-resistant SKFE. However, their construction incurs exponential security loss in the depth of circuits, and thus their construction leads to only PKFE for NC^1 if we allow only polynomial security loss. In order to construct PKFE scheme for $\mathsf{P/poly}$ with polynomial security loss through the construction of Bitansky *et al.*, we additionally need to assume weak PRF computable in NC^1 to bootstrap the circuit class of PKFE. Moreover, we believe that our construction and proof are significantly simpler than theirs as we show in Section 3.2.

We next see that single-key weakly-succinct SKFE is also powerful enough to yield PKFE if we additionally assume identity-based encryption. By Theorems 3.4, 4.1 and 5.2, we obtain the following theorem since Theorem 3.4 just requires that the compression factor of SXIO $\widetilde{\gamma}$ is slightly smaller than 1 (no need to be sufficiently small).

Theorem 5.6. *If there exists* δ*-secure identity-based encryption and* $(1, \delta)$*-selective-message message private and weakly succinct SKFE for* $\mathsf{P/poly}$*, then there exists* $(1, \delta)$*-weakly-selective secure and weakly succinct PKFE for* $\mathsf{P/poly}$*.*

By combining this theorem with Theorem 5.3, we obtain the following corollary stating that we can construct collusion-resistant PKFE from single-key weakly succinct SKFE if we additionally assume identity-based encryption.

Corollary 5.7. *If there exits* δ*-selectively-secure identity-based encryption and* $(1, \delta)$*-selectively-secure weakly succinct SKFE for* $\mathsf{P/poly}$*, then there exists* (poly, δ)*-selectively secure and succinct PKFE for* $\mathsf{P/poly}$*.*

Moreover, by combining Theorem 5.6 with Theorem 2.21, IO is obtained if we assume sub-exponential security of single-key weakly succinct SKFE and identity-based encryption.

Corollary 5.8. *If there exits sub-exponentially selectively secure identity-based encryption and sub-exponentially selectively secure single-key weakly succinct SKFE for* $\mathsf{P/poly}$*, then there exists IO for* $\mathsf{P/poly}$*.*

The combination of the result of Lin *et al.* [LPST16] and that of Bitansky *et al.* [BNPW16b] imply that single-key weakly-succinct SKFE is strong enough to yield IO under the learning with errors (LWE) assumption. Regarding this implication, Corollary 5.8 shows that we can replace the LWE assumption with general primitive, that is identity-based encryption. We believe that this fact is important to understand the nature of IO.

Figure 11 illustrates our results stated above.

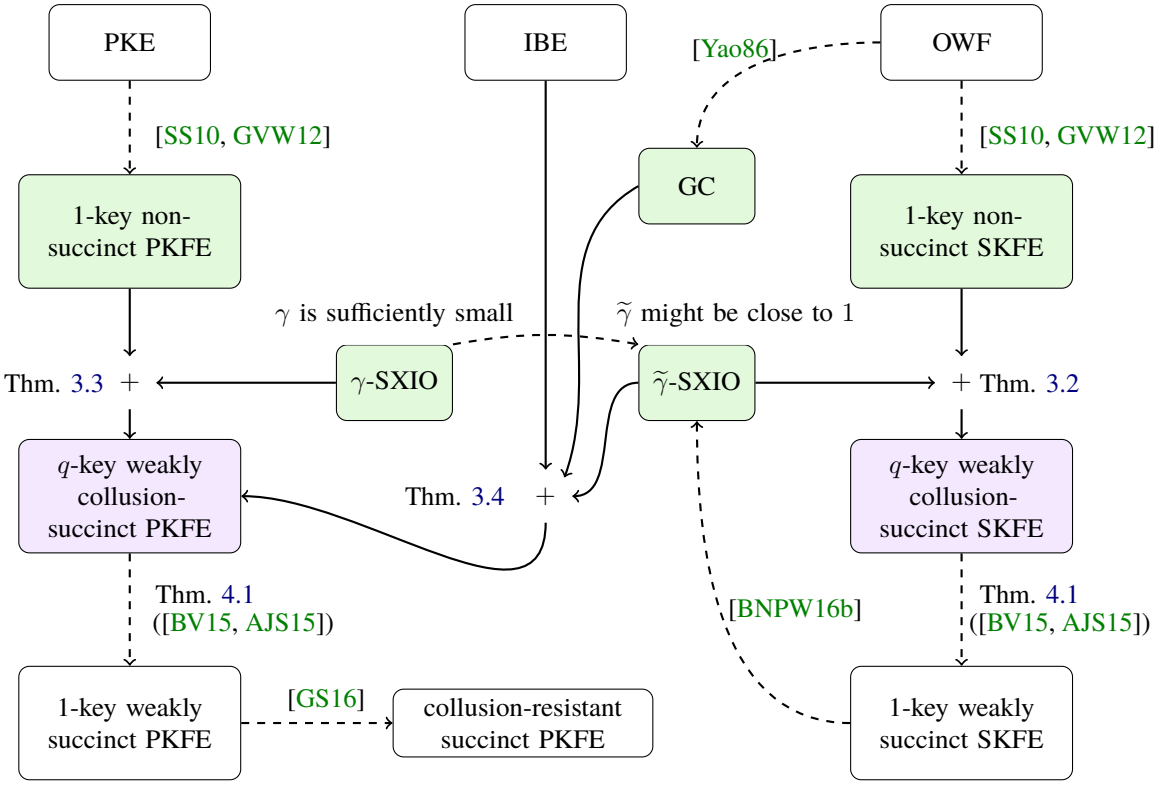

Figure 11: Illustration of our theorems. Dashed lines denote known facts or trivial implications. White boxes denote our ingredients or goal. Purple boxes denote our key schemes. Green boxes denote our intermediate tools. γ-SXIO denotes SXIO with compression factor γ, which is *sufficiently small* constant 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. They are implied by one-way function.

5.2 Equivalence of SKFE, SXIO, and Updatable Randomized Encoding

By Theorems 3.1, 3.2 and 4.1, we obtain the following theorem.

Theorem 5.9. *If there exists δ-secure one-way function and δ-secure 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 $(1, \delta)$-selective-message message private and weakly succinct SKFE for $\mathsf{P/poly}$.*

By combining this theorem and Theorem 5.2, we obtain the following corollary stating that the existence of single-key weakly-succinct SKFE is equivalent to those of SXIO and one-way function.

Corollary 5.10. *A single-key weakly succinct SKFE for $\mathsf{P/poly}$ is equivalent to one-way function and $\widetilde{\gamma}$-compressing SXIO for $\mathsf{P/poly}$ such that $0 < \widetilde{\gamma} < 1$ ($\widetilde{\gamma}$ might be close to 1).*

We can also obtain equivalence of these primitives and updatable randomized encoding. We introduce the following results related to updatable randomized encoding shown by Ananth *et al.* [ACJ17].

Theorem 5.11 ([ACJ17]). *A single-key weakly succinct SKFE for $\mathsf{P/poly}$ implies output-compact updatable randomized encoding with an unbounded number of updates.*

Theorem 5.12 ([ACJ17]). *Output-compact updatable randomized encoding with an unbounded number of updates implies a $\widetilde{\gamma}$-compressing SXIO for $\mathsf{P/poly}$ where $\frac{1}{2} \leq \widetilde{\gamma} < 1$.*

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 can be extended to $\widetilde{\gamma}$-compressing SXIO.

By Theorems 5.9, 5.11 and 5.12, we can obtain the following corollary.

Corollary 5.13. *A single-key weakly succinct SKFE for* P/poly *is equivalent to one-way function and output-compact updatable randomized encoding with an unbounded number of updates.*

Ananth *et al.* show that single-key weakly-succinct SKFE is equivalent to the combination of updatable randomized encoding and the LWE assumption. Regarding the result, Corollary 5.13 shows that the LWE assumption is replaced with weaker and general assumption, that is one-way function.

References

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 657–677. Springer, Heidelberg, August 2015. (Cited on page 3.)

[ACJ17] Prabhanjan Ananth, Aloni Cohen, and Abhishek Jain. Cryptography with updates. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, pages 445–472, 2017. (Cited on page 5, 28.)

[ADGM17] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. In *44rd International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland (to appear)*, 2017. (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 3, 10.)

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Gennaro and Robshaw [GR15], pages 308–326. (Cited on page 1, 2, 3, 4, 13.)

[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 2, 3, 5, 7, 24, 28.)

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012. (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 8.)

[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 27.)

[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 2, 3, 4, 6, 7, 10, 11, 27.)

[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 3, 5, 27, 28.)

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012. (Cited on page 2.)

[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 12, 13.)

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, volume 6597 of *LNCS*, pages 253–273. Springer, Heidelberg, March 2011. (Cited on page 1.)

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In Guruswami [Gur15], pages 171–190. (Cited on page 1, 2, 3, 4, 5, 7, 14, 24, 25, 28.)

[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 8.)

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996. (Cited on page 34.)

[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.)

[CGH17] Yilei Chen, Craig Gentry, and Shai Halevi. Cryptanalyses of candidate branching program obfuscators. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, pages 278–307, 2017. (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.)

[CLLT17] Jean-Sébastien Coron, Moon Sung Lee, Tancrède Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over CLT13. In *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, pages 41–58, 2017. (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 30, 31.)

[FC16a] Marc Fischlin and Jean-Sébastien Coron, editors. *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*. Springer, Heidelberg, May 2016. (Cited on page 31, 32.)

[FC16b] Marc Fischlin and Jean-Sébastien Coron, editors. *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*. Springer, Heidelberg, May 2016. (Cited on page 30.)

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

[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.)

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986. (Cited on page 1, 8.)

[GKP+13] Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 555–564. ACM Press, June 2013. (Cited on page 3.)

[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 30, 31.)

[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 1, 2, 3, 14, 25, 27, 28.)

[Gur15] Venkatesan Guruswami, editor. *56th FOCS*. IEEE Computer Society Press, October 2015. (Cited on page 30.)

[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 1, 5, 7, 15, 28, 33, 34.)

[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 30, 32.)

[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 3.)

[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 8.)

[KS17] Ilan Komargodski and Gil Segev. From minicrypt to obfustopia via private-key functional encryption. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 122–151, 2017. (Cited on page 4, 7.)

[LM16] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In Hirt and Smith [HS16], pages 443–468. (Cited on page 1, 2, 3, 13.)

[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 2, 3, 4, 27.)

[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 7.)

[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.)

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *Journal of the ACM*, 51(2):231–262, 2004. (Cited on page 2.)

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In STOC 1989 [STO89], pages 33–43. (Cited on page 1.)

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. http://eprint.iacr.org/2010/556. (Cited on page 1.)

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In STOC 1990 [STO90], pages 387–394. (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 1, 5, 6, 7, 28, 33.)

[STO89] *21st ACM STOC*. ACM Press, May 1989. (Cited on page 33.)

[STO90] *22nd ACM STOC*. ACM Press, May 1990. (Cited on page 33.)

[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, 8.)

[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 3, 9, 10, 28.)

A Single-Key Non-Succinct Functional Encryption

Our construction of weakly-succinct PKFE (resp. SKFE) uses a single-key non-succinct PKFE (resp. SKFE) scheme as a building block. As observed by Sahai and Seyalioglu [SS10] and later extended by Gorbunov *et al.* [GVW12], single-key non-succinct functional encryption scheme can be constructed based on standard assumptions such as public-key encryption and one-way function.

For self containment, we show the construction of single-key non-succinct PKFE scheme based on public-key encryption scheme. More specifically, the construction is based on garbling scheme and public-key encryption.

Let $\mathsf{GC} = (\mathsf{Grbl}, \mathsf{Eval})$ be a garbling scheme, and $\mathsf{PKE} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme. Using GC and PKE, we construct a single-key PKFE scheme $\mathsf{OneKey} = (\mathsf{1Key.Setup}, \mathsf{1Key.KG}, \mathsf{1Key.Enc}, \mathsf{1Key.Dec})$ as follows. Below, we assume that we can represent every function f by an n-bit string $(f[1], \cdots, f[s])$.

Construction. The scheme consists of the following algorithms.

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

- Generate $(\mathsf{pk}_{j,\alpha}, \mathsf{sk}_{j,\alpha}) \leftarrow \mathsf{KG}(1^\lambda)$ for every $j \in [s]$ and $\alpha \in \{0,1\}$.
- Return $\mathsf{MPK} \leftarrow \{\mathsf{pk}_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}$ and $\mathsf{MSK} \leftarrow \{\mathsf{sk}_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}$.

$\mathsf{1Key.KG}(\mathsf{MSK}, f)$:

- Parse $\{\mathsf{sk}_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}} \leftarrow \mathsf{MSK}$ and $(f[1], \cdots, f[s]) \leftarrow f$.
- Return $\mathsf{sk}_f \leftarrow (f, \{\mathsf{sk}_{j,f[j]}\}_{j\in[s]})$.

$\mathsf{1Key.Enc}(\mathsf{MPK}, m)$:

- Parse $\{\mathsf{pk}_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}} \leftarrow \mathsf{MPK}$.
- Compute $(\widetilde{U}, \{L_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{Grbl}(1^\lambda, U(\cdot, m))$.
- For every $j \in [s]$ and $\alpha \in \{0,1\}$, compute $c_{j,\alpha} \leftarrow \mathsf{Enc}(\mathsf{pk}_{j,\alpha}, L_{j,\alpha})$.
- Return $\mathsf{CT} \leftarrow (\widetilde{U}, \{c_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}})$.

$\mathsf{1Key.Dec}(\mathsf{sk}_f, \mathsf{CT})$:

- Parse $(f, \{\mathsf{sk}_j\}_{j\in[s]}) \leftarrow \mathsf{sk}_f$ and $(\widetilde{U}, \{c_{j,\alpha}\}_{j\in[s],\alpha\in\{0,1\}}) \leftarrow \mathsf{CT}$.
- For every $j \in [s]$, compute $L_j \leftarrow \mathsf{Dec}(\mathsf{sk}_{j,f[j]}, c_{j,f[j]})$.
- Return $y \leftarrow \mathsf{Eval}(\widetilde{U}, \{L_j\}_{j\in[s]})$.

OneKey is single-key PKFE scheme that satisfies weakly-selective security if GC is secure and PKE is CPA-secure. The construction is non-succinct since the encryption algorithm of OneKey encrypts a universal circuit whose size is at least linear in the size of functions.

We can analogously construct single-key non-succinct SKFE scheme based on garbling scheme and secret-key encryption.

Gorbunov *et al.* [GVW12] later showed how to extend this construction to adaptively secure one using a technique of non-committing encryption [CFGN96]. This is done by only using public-key encryption (or one-way function) if we focus on single-key schemes. [15] Thus, we need only public-key encryption or one-way function to obtain single-key adaptively secure schemes for our building blocks.

[15] If we want to achieve bounded collusion-resistant schemes, we additionally need a pseudo-random generator that is computed by polynomial degree circuits, which is implied by number theoretic or lattice assumptions [GVW12].