From Weakly Selective to Selective Security in Compact Functional Encryption, Revisited

Linfeng Zhou*

Cheetah Mobile Inc.

Abstract

We provide a generic transformation from weakly selective secure FE to selective secure FE through an approach called *hybrid functional key generation*. Furthermore, our transformation preserves the compactness of the FE scheme. Additionally, we note that this transformation is much simpler than the prior work [GS16]. We consider the simplicity of the construction in this work as a positive feature and the hybrid functional key generation approach as a new method that can be applied in functional encryption schemes. Furthermore, we try to weaken the input FE scheme of our transformation to be a non-compact one instead of a fully-compact one, by additionally assuming the hardness of LWE (or Ring-LWE) assumption. We achieve this result by utilizing the FE scheme for bounded collusions with *decomposable and succinct ciphertext property*, which can be solely based on the LWE (or Ring-LWE) assumption. Finally we present the implications of our result, which improves previous results, in building general-purpose indistinguishability obfuscator from (well-expressed) functional encryption.

1 Introduction

Indistinguishability obfuscation ($i\mathcal{O}$), first defined in the seminal work of Barak et al. [BGI+01] and further investigated in [GR07], is currently an extraordinarily powerful object on the cryptographic landscape. Since Garg et al. [GGH+13] put forward a plausible candidate obfuscation algorithm, $i\mathcal{O}$ has been successfully used to solve a wide range of open problems (e.g., [GGH+13, SW14, CLP15, BGJ+16]), to achieve new cryptographic goals (e.g., [KLW15, BGL+15, CHJV15]), and even to imply notions outside of cryptography (e.g., [BPR15, GPS16])

However, the problem of building an indistinguishability obfuscator with a solid proof of security still remains uncertain. The multilinear-map problems [GGH+13, CLT13, CHL+15, CLT15] underlying most known candidate $i\mathcal{O}$ constructions [GGH+13, BR14, BGK+14, AB15, PST14, GMS16, MSZ16] have recently been subject to attacks [CHL+15, GHMS14, ADGM16, CLLT16, CGH16], and basing $i\mathcal{O}$ on a solid, well-understood standard complexity assumption, has rapidly emerged as perhaps one of the most pressing open problems in theoretical cryptography.

Bitansky and Vaikuntanathan [BV15] and Ananth and Jain [AJ15] opened another door towards building $i\mathcal{O}$ from standard assumptions, these two independent works showed that $i\mathcal{O}$ can be built from any public key functional encryption scheme satisfying certain compactness requirements, but with sub-exponential security loss. While general constructions of compact functional encryption (for arbitrary functions) are only known using $i\mathcal{O}$ [GGH+13, AS16a], functional encryption is typically considered a weaker primitive than general-purpose $i\mathcal{O}$.

[*] daniel.linfeng.zhou@gmail.com

We recall that a (public key) functional encryption scheme [BSW12, O'N10, Wat13, AGVW13] is an (public-key) encryption scheme that allows for the creation of functional secret keys SK_f corresponding to functions f, such that when such a functional secret key SK_f is applied to an encryption of message m, one could decrypt it only yielding $f(m)$, but nothing else more about the message m.

Properties of interest in the study of functional encryption generally lie in the following three phases [1]:

- Security: The security of FE scheme can be captured by an indistinguishability-based security game [2] between a challenger and an adversary. Ideally, we want the FE scheme to achieve *adaptive security*, which guarantees security that both functional secret keys and messages can be adaptively chosen at any point in time. To ease the security notion people often consider security under a weaker notion of *selective security* where the adversary is forced to commit the challenge messages before seeing the public key. Furthermore, it is possible to further weaken the security notion of FE to *weakly selective security*, where the adversary must commit not only to the challenge messages, but also to all the functional queries before seeing the public key.
- Compactness: which captures the time (or circuit) complexity of the encryption algorithm. This is the central notion of efficiency of FE. Ideally, we want to achieve the strongest efficiency notion of FE, which is called *full compactness*. A FE scheme is said to be *fully-compact* if the size of the encryption circuit is some polynomial in the size of the message to be encrypted and the security parameter, but independent of the circuit size of the functions in the corresponding function family. A FE scheme has *non-compact* ciphertext if the size of the encryption circuit grows with the maximum circuit size of functions. A relaxed property lying between fully-compact FE and non-compact FE that has been considered in literature is called *weakly compact*. A FE scheme has *weakly-compact* ciphertexts if the size of the encryption circuit grows sublinearly with the maximum circuit size of functions.
- Collusion-resistance: The number of functional secret key queries that can be released is also an essential parameter considered in the FE scheme. More specifically, FE schemes can be parameterized based on whether the adversary obtains a-priori bounded or unbounded number of functional secret key queries. In this work, we mainly consider FE for *single-key* query (i.e., single-key FE and FE for *multi-key* queries (i.e., multi-key FE.

There are a lot of works considering these properties described above. Sahai and Seyalioglu [SS10] and Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [GKP$^+$13] provided FE scheme supporting all of P/poly circuits based on standard assumptions. However, these constructions only support single-key query. Garg et al. [GGH$^+$13] construct a multi-key FE scheme for P/poly circuits. Nevertheless, their construction is based on $i\mathcal{O}$. Ananth et al. [ABSV15] show a generic transformation from selective security to adaptive security in functional encryption schemes. However, their transformation is not compact-preserving, which means that the output FE of their transformation is always non-compact. Goyal et al. [GKW16] give a generic transformation from selective security to semi-adaptive security, which is a security notion of FE schemes lying between the selective security and the adaptive security in FE schemes. We remark that their transformation also works for the attribute-based encryption scheme, which is a restricted version of the functional encryption scheme. Recently in the concurrent works of Garg and Srinivasan [GS16], and Li and Micciancio [LM16a], they show how to transform any single-key FE to multi-key FE through two different approaches respectively. Moreover, the transformation of Garg and Srinivasan can be viewed as a transformation from weakly selective to selective security in FE schemes. However, the input FE scheme of their transformation must be weakly compact.

In this work, we ask the following question:

[1] Also, the class of functions supported by the FE scheme is also an important property, but for simplicity here we will focus on schemes for which the class of functions can be a class of polynomial sized circuits.

[2] The security of FE scheme can be captured by a simulation-based security game as well, but in this work we only consider its indistinguishability-based security

Can we reduce the general-purpose indistinguishability obfuscator to the weakest variant of functional encryption (i.e., single-key, weakly selective secure, non-compact FE*)?*

We answer this question affirmatively through the contributions described in the next section[3].

2 Our Contributions

In this work we show a *simple* generic transformation from any weakly-selective functional encryption scheme for single-key query to a selectively-secure one for single-key query, without relying on any additional assumptions. Our transformation applies equally to public-key schemes and to private-key ones [4], where the resulting selective scheme inherits the public key or private-key flavor of the underlying scheme. In particular, we note that this transformation preserves the compactness of FE scheme. Namely, if the input FE scheme is compact, the output FE scheme is still compact. The following theorem informally summarizes our main contribution.

Theorem 2.1 (Informal)**.** *Given any public-key (resp. private-key) weakly selective secure, fully-compact functional-encryption scheme for the class of all polynomial-size circuits, there exists a public-key (resp. private-key) selective secure, fully-compact functional encryption scheme.*

We remark that, for the specific purpose of transforming weakly selective secure FE scheme to a selective secure one, the transformation we provide is a *simpler* one than the prior work [GS16]. We achieve this transformation through the novel technique called *hybrid functional key generation*, which is opposed to the technique called hybrid functional encryption utilized in the work of Ananth et al.[ABSV15].

Furthermore, we try to weaken the input FE scheme of our transformation by taking advantage of additional assumptions (i.e., LWE or Ring-LWE), and thus we obtain the following result.

Theorem 2.2 (Informal)**.** *Given any public-key (resp. private-key) weakly-selective,* ***non-compact*** *functional encryption scheme for the class of all polynomial size circuits for single-key query, there exists a selectively-secure public-key (resp. private-key)* ***compact*** *functional encryption scheme with similar properties, additionally assuming the hardness of LWE assumption.*

We achieve the above result by utilizing the FE scheme for bounded collusions with *decomposable and compact ciphertext property.* We remark that this kind of FE schemes was recently achieved by Agrawal and Rosen [AR16], based solely on the LWE (or Ring-LWE) assumption.

We view the significance of our result in two dimensions. First of all, combining our work with the transformation (from single-key to multi-key FE scheme) proposed in the recent work of Li and Micciancio [LM16b], we could obtain a multi-key, selective, compact FE which has shorter public keys when compared to the work of Garg and Srinivasan [GS16]. Also, if we just want the selective scheme to be single-key secure then the weakly selective scheme can even be non-compact, whereas the input FE scheme of the transformation proposed by Garg and Srinivasan must be weakly-compact. We consider the simplicity of the construction in this work as a positive feature and the hybrid functional key generation approach as a new method that can be applied in functional encryption schemes. Secondly, we investigate the implications of our result in building general-purpose indistinguishability obfuscator and stronger variant of FE scheme, and we obtain the following corollaries.

[3] Recently these works [Lin16a, Lin16a, Lin16b, AS16b] show how to construct general-purpose $i\mathcal{O}$ from degree-5 multilinear maps. While their focus is on building $i\mathcal{O}$ from the minimal assumptions over multilinear maps, we focus on building $i\mathcal{O}$ from functional encryption.

[4] Since the page limitation, we refer the reader to our transformation under the private-key setting in the fully version of this work.

Corollary 2.1 (This work + [AJ15, BV15], Informal)**.** *Assuming the existence of one-way functions and the hardness of LWE assumption, there exists a general-purpose indistinguishability obfuscator given any single-key, weakly-selective, non-compact* FE *scheme with sub-exponential security.*

Corollary 2.2 (This work + [GS16, AS16a, HJO$^+$15], Informal)**.** *Assuming the existence of one-way functions and the hardness of LWE assumption, any multi-key, adaptively-secure, width-compact*[5] FE *scheme can be polynomially reduced to a single-key, weakly selective secure, non-compact* FE *scheme.*

The corollary 2.1 answers the question we proposed in the last section affirmatively. In particular, it improves the previous result that the general-purpose indistinguishability obfuscator is implied by any single-key, weakly-selective, *weakly-compact* FE scheme with sub-exponential security. In contrast, our result shows that general-purpose indistinguishability obfuscator is implied by any single-key, weakly-selective, *non-compact* FE scheme with sub-exponential security, additionally assuming the hardness of LWE (or Ring-LWE) assumption.

3 Our Techniques

We give an overview of our intuition and techniques utilized in this work.

From Weakly Selective Secure to Selective Secure FE**.** Before illustrating our intuition of the transformation, let us first show the gap between the weakly selective security game and the selective security game by describing a reduction. The reduction could internally execute some adversaries to break the underlying selective secure FE scheme, while simulating the role of the challenger of the selective secure FE scheme. At the very beginning, the adversary first submits a pair of messages, and then the reduction which simulates the role of the challenger returns back an functional encryption of the message corresponding to a random bit $b \in \{0, 1\}$ flipped by the challenger of the underlying weakly selective secure FE scheme. Recall that the message challenge phase of weakly selective security game is the same as the one of selective security game, except that the adversary must submit a function query (note that we only consider FE supporting single-key query) along with the pair of challenge messages together. We notice that this difference does not effect the challenger of the weakly selective game to compute the functional encryption over the message corresponding to the random bit $b \in \{0, 1\}$. Nevertheless, the bad news is from the message challenge phase. In the weakly selective security game, the adversary cannot generate any functional secret key since the reduction does not receive any function query. Therefore, the key query phase between the reduction and the adversary (of the selective secure FE) will not be opened. Thus the intractable point to construct the reduction is how to submit the function query while not receiving any function query from the adversary (of the selective secure FE).

To solve this problem, our idea is to deploy two functional secret keys rather than only one functional secret key. Namely the final functional secret key is comprised of two functional secret keys in order to separate the key generation step such that the reduction could submit the function query to the challenger of weakly selective security game without the information of the function query from the adversary, and then the reduction could receive back the challenge ciphertext and in turn go through the following steps.

Hybrid Key Generation. Taking this idea root in mind, one may ask how does the reduction generate a function query without any information of the function query from the adversary? That is, the function query generated by the reduction should be independent of the function query from the adversary. To handle this we propose a novel approach called *hybrid functional key generation*. The intuition of this approach is from the observation that messages and functions enjoy the same level of privacy in FE scheme. Indeed, [BS15] shows this through transforming private-key FE schemes into function private FE schemes. Therefore, after applying the [BS15] transformation, we can switch the roles of functions and messages. Our technique is

[5] We refer the reader to [GS16, HJO$^+$15] for the formal definition of width-compact FE schemes.

basically achieved by applying hybrid functional encryption and dual-system encryption over the functional secret key generation algorithm.

Ananth et al. [ABSV15] have shown the power of hybrid functional encryption and dual-system encryption techniques in transforming selective security to adaptive security in functional encryption schemes. The idea in hybrid functional encryption is to combine two encryption schemes. An “external" scheme (i.e., key encapsulation mechanism) and an “internal" scheme (i.e., data encapsulation mechanism). In order to encrypt a message in the hybrid scheme, a fresh key is generated for the internal schemes, and is used to encrypt the message. Then the key itself is encrypted using the external scheme. The final hybrid ciphertext contains two ciphertexts: $\mathsf{CT}_0 = \mathsf{Enc}_{\mathsf{int},\mathsf{k}}(m)$ and $\mathsf{CT}_1 = \mathsf{Enc}_{\mathsf{ext}}(k)$ (all external ciphertexts use the same key). To decrypt, one first decrypts the external ciphertext, retrieves k and applies it to the internal ciphertext. The hybrid functional encryption method in functional encryption scheme relates to the dual-system encryption technique because the two ciphertexts $\mathsf{CT}_0 = \mathsf{Enc}_{\mathsf{int},\mathsf{k}}(m)$ and $\mathsf{CT}_1 = \mathsf{Enc}_{\mathsf{ext}}(\mathsf{k})$ control the dual-system encryption externally and internally. At first glance the execution of the functional encryption algorithm and the functional secret key generation algorithm can be done interchangeably since messages and functions enjoy the same level of privacy in FE schemes. Namely the functional encryption algorithm can be viewed as a kind of functional secret key generation algorithm for the message m using the master public key (or master secret key in private-key functional encryption schemes), and the functional secret key generation algorithm can be viewed as a kind of functional encryption algorithm for the function f using the master secret key.

Our idea in hybrid functional key generation is to combine two key generation schemes in a similar internal-external manner. In order to generate a functional secret key in the hybrid scheme, a fresh key k is generated for the internal scheme, and is used to generate the functional secret key. Then the key k itself is hardwired into a circuit G. We denote the hardwired circuit by G_{k}. Then a functional secret key corresponding to the circuit G is generated using the external scheme. The final hybrid functional key generation contains the two functional secret keys $(\mathsf{KG}_{\mathsf{ext}}(G_{\mathsf{k}}), \mathsf{KG}_{\mathsf{int},\mathsf{k}}(f))$ (all external functional secret keys use the same key). To decrypt, one first decrypts the ciphertext using the external functional secret key, retrieves a ciphertext CT (i.e., the soldier hidden in the Trojan Horse) and applies the internal functional secret key to decrypt it.

From Non-compact FE to Compact FE. While the hybrid functional key generation method successfully transforming any weakly selective secure FE to selective secure FE, we observe that this transformation is compact-preserving. Namely, if the input FE scheme is fully-compact, then the output FE scheme is also fully-compact. To weaken the input FE scheme of our transformation to be a non-compact scheme instead of a fully-compact one, we employ the recent result of Agrawal and Rosen [AR16]. Recall that they showed how to construct an improved FE scheme for bounded collusions, assuming the hardness of LWE (or Ring-LWE) assumption, where it enjoys a special property called *decomposable and compact ciphertext.* More specifically, the ciphertext CT of their FE scheme, which encrypts a message $m = (m_1, \cdots, m_\ell)$ of length $\ell = \ell(\lambda)$, can be decomposed as the following form:

$$\mathsf{CT} = \left(\underbrace{\mathsf{CT}_{\mathsf{data}} = (\mathsf{CT}_1, \cdots, \mathsf{CT}_\ell)}_{\text{compact data-dependent component}} \;\middle|\; \underbrace{\mathsf{CT}_{\mathsf{indpt}}}_{\text{non-compact data-independent component}} \right)$$
$$\mathsf{CT}_i = \mathsf{FE.Enc}(\mathsf{MPK}, m_i, r; \hat{r}_i) \quad \forall i \in [\ell] \qquad \mathsf{CT}_{\mathsf{indpt}} = \mathsf{FE.Enc}(\mathsf{MPK}, r; \hat{r})$$

where r is the shared randomness of each component. Additionally each component may utilize independent randomness $\hat{r}_1, \cdots, \hat{r}_\ell$ and $\hat{r}$ respectively. We call $\mathsf{CT}_{\mathsf{data}}$ as the compact data-dependent component, and it can be further decomposed as $\mathsf{CT}_{\mathsf{data}} = (\mathsf{CT}_1, \cdots, \mathsf{CT}_\ell)$, where each CT_i is compact (i.e., the size of each CT_i is polynomial only in the security parameter and the message length but not related to the circuit size and the number of functional queries.). In contrast, we call $\mathsf{CT}_{\mathsf{indpt}}$ as the non-compact data-independent component.

Since a (selectively-secure) private-key FE scheme for bounded collusions is one of the building blocks of our transformation, we can naturally embed such FE scheme (with decomposable and compact ciphertext property) into it. More specifically, to generate the functional secret key we let the circuit G_{k} to output the compact data-dependent component $\mathsf{CT}_{\mathsf{data}} = \{\mathsf{CT}_i\}_{i\in[\ell]}$ and the non-compact data-independent component $\mathsf{CT}_{\mathsf{indpt}}$ as an additional element of the functional secret key. Recall that in our transformation the compactness of the ciphertext is only associated with the circuit size of G_{k}, therefore the result scheme of our transformation is fully-compact even though the input scheme is non-compact. We remark that [AJS15] also provided a similar transformation from non-compact FE to fully-compact FE. However, we utilize the decomposable and compact ciphertext property of FE schemes for bounded collusions, instead of the decomposable randomized encoding schemes.

Our Construction in a Nutshell. We give a brief description of our construction. It first sets up the master key pair $(\mathsf{MPK}, \mathsf{MSK})$ with respect to the underlying weakly selective secure FE scheme. To generate functional secret keys, the key generation algorithm constructs the trapdoor circuit G as follows: the circuit G, which is hardwired with a master secret key that is newly generated with respect to a selectively-secure one-ciphertext FE scheme, a pseudorandom ciphertext C_E and a random tag τ, takes as input the message m, a PRF key K_p, a symmetric key K_E and a bit β and it outputs the result in two threads. If $\beta = 1$, it outputs the symmetric decryption of C_E using the symmetric key K_E, otherwise it outputs an encryption over the message m using the master secret key hardwired inside of the circuit G. Note that this encryption is derandomized using the PRF key K_p. Finally the key generation algorithm outputs a pair of functional secret keys $(\mathsf{SK}_f, \mathsf{SK}_G)$ as the functional secret key. The ciphertext of our construction is an encryption of the tuple $(m, K_p, 0^\lambda, 0)$, where K_p is a newly sampled PRF key, using the underlying weakly selective secure FE scheme. To decrypt, one can decrypt the ciphertext using the functional secret key SK_G to release the internal ciphertext and then to decrypt the internal ciphertext using the functional secret key SK_f.

4 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. For a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. We denote by $y \leftarrow f(x)$ the process of sampling a value y from the distribution $f(x)$ given a randomized function $f \in \mathcal{F}$ and an input $x \in \mathcal{X}$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for any polynomial $\mathsf{poly}(\cdot)$ we have $\mathsf{negl}(\lambda) < 1/\mathsf{poly}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

4.1 Public-Key Functional Encryption

A public-key functional encryption scheme PKFE over a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a tuple $(\mathsf{PKFE.Setup}, \mathsf{PKFE.KG}, \mathsf{PKFE.Enc}, \mathsf{PKFE.Dec})$ of PPT algorithms with the following properties.

- $\mathsf{PKFE.Setup}(1^\lambda)$: The setup algorithm takes as input the unary representation of the security parameter, and outputs a master public key MPK and a master secret key MSK.
- $\mathsf{PKFE.KG}(\mathsf{MSK}, f)$: The key generation algorithm takes as input a secret key MSK and a function $f \in \mathcal{F}_\lambda$ and outputs a functional secret key SK_f.
- $\mathsf{PKFE.Enc}(\mathsf{MPK}, m)$: The encryption algorithm takes as input a master public key MPK and a message $m \in \mathcal{M}_\lambda$, and outputs a ciphertext CT.
- $\mathsf{PKFE.Dec}(\mathsf{SK}_f, \mathsf{CT})$: The decryption algorithm takes as input a functional secret key SK_f and a ciphertext CT, and outputs $m \in \mathcal{M}_\lambda \cup \{\bot\}$

We say a public-key functional encryption scheme is defined for a complexity class $\mathcal{C}$ if it supports all the functions that can be implemented in $\mathcal{C}$.

Correctness. We require that there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all sufficiently large $\lambda \in \mathbb{N}$, for every message $m \in \mathcal{M}_\lambda$, and for every function $f \in \mathcal{F}_\lambda$ we have

$$\Pr\left[\mathsf{PKFE.Dec}(\mathsf{PKFE.KG}(\mathsf{MSK}, f), \mathsf{PKFE.Enc}(\mathsf{MPK}, m)) = f(m)\right] \geq 1 - \mathsf{negl}(\lambda)$$

where $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{PKFE.Setup}(1^\lambda)$, and the probability is taken over the random choices of all algorithms.

Security. We consider the standard selective and adaptive indistinguishability-based notions for functional encryption. Intuitively, these notions ask that encryptions of any two messages, m_0 and m_1, should be computationally indistinguishable given access to functional secret keys for any function f such that $f(m_0) = f(m_1)$. In the case of selective security, adversaries are required to specify the two messages in advance (i.e., before interacting with the system). In the case of adaptive security, adversaries are allowed to specify the two messages even after obtaining the master public key and functional secret keys.

Remark. Our notions of security consider a single challenge, and in the public-key setting these are known to be equivalent to their multi-challenge variants via a standard hybrid argument.

Definition 4.1 (Weakly Selective Security)**.** A public-key functional encryption scheme PKFE over a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ is *weak selective secure* if for any PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathbf{Adv}^{\mathrm{wSel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, 1) = 1]\right| \leq \mathsf{negl}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for each $b \in \{0, 1\}$ the experiment $\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, b)$, modeled as a game between the adversary $\mathcal{A}$ and a challenger, is defined as follows:

1. **Challenge Phase**: The adversary $\mathcal{A}$ outputs two messages (m_0, m_1) such that $|m_0| = |m_1|$ and a set of functions $f_1, \cdots, f_q \in \mathcal{F}$ to the challenger. The parameter q and the size of message vectors are apriori-unbounded.
2. The challenger samples $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{PKFE.Setup}(1^\lambda)$ and generates the challenger ciphertext $\mathsf{CT} \leftarrow \mathsf{PKFE.Enc}(\mathsf{MPK}, m_b)$. The challenger also computes $\mathsf{SK}_{f,i} \leftarrow \mathsf{PKFE.KG}(\mathsf{MSK}, f_i)$ for all $i \in [q]$. It then sends $(\mathsf{MPK}, \mathsf{CT}), \{\mathsf{SK}_{f,i}\}_{i \in [q]}$ to the adversary $\mathcal{A}$.
3. If $\mathcal{A}$ makes a query f_j for some $j \in [q]$ to functional secret key generation oracle such that $f_j(m_0) \neq f_j(m_1)$, the output of the experiment is $\perp$. Otherwise the output is b' which is the output of $\mathcal{A}$

Remark. We say that the functional encryption scheme PKFE is *single-key*, *weakly selective secure* if the adversary $\mathcal{A}$ in $\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, b)$ is allowed to obtain the functional secret key for a single function f.

Definition 4.2 (Selective Security)**.** A public-key functional encryption scheme PKFE over a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ is *selectively secure* if for any PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathbf{Adv}^{\mathrm{Sel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, 1) = 1]\right| \leq \mathsf{negl}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for each $b \in \{0, 1\}$ the experiment $\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{pkfe},\mathcal{A}}(\lambda, b)$, modeled as a game between the adversary $\mathcal{A}$ and a challenger, is defined as follows:

1. **Setup Phase**: The challenger samples $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{PKFE.Setup}(1^\lambda)$.

2. **Challenge Phase**: The adversary submits a pair of message (m_0, m_1), and the challenger replies with MPK and $\mathsf{CT} \leftarrow \mathsf{PKFE.Enc}(\mathsf{MPK}, m_b)$, where b is a random coin flipped by the challenger.
3. **Query Phase**: The adversary adaptively queries the challenger with any function $f \in \mathcal{F}_\lambda$ such that $f(m_0) = f(m_1)$. For each such query, the challenger replies with $\mathsf{SK}_f \leftarrow \mathsf{PKFE.KG}(\mathsf{MSK}, f)$.
4. **Output Phase**: The adversary outputs a bit b' which is defined as the output of the experiment.

Efficiency. We now define the efficiency requirements of a PKFE scheme.

Definition 4.3 (Fully Compact)**.** A public-key functional encryption scheme PKFE is said to be *fully compact* if for all security parameter $\lambda \in \mathbb{N}$ and for all message $m \in \{0,1\}^*$ the running time of the encryption algorithm $\mathsf{PKFE.Enc}$ is $\mathsf{poly}(\lambda, |m|)$.

Definition 4.4 (Weakly Compact)**.** A public-key functional encryption scheme PKFE is said to be *weakly compact* if for all security parameter $\lambda \in \mathbb{N}$ and for all message $m \in \{0,1\}^*$ the running time of the encryption algorithm $\mathsf{PKFE.Enc}$ is $s^\gamma \cdot \mathsf{poly}(\lambda, |m|)$, where $\gamma < 1$ is a constant and $s = \max_{f \in \mathcal{F}} |C_f|$, where C_f is a circuit implementing the function f.

A public-key functional encryption scheme is said to be *non-compact* if the running time of the encryption algorithm can depend arbitrarily on the maximum circuit size of the function family.

Definition 4.5 (Bounded Collusions)**.** We say a functional encryption is *q-bounded* if the adversary is given functional secret keys for a-priori bounded number of functions $f_1, \cdots, f_q$, which can be made adaptively.

Definition 4.6 (Decomposable and succinct FE Ciphertext, [AR16])**.** We say a functional encryption has *decomposable and succinct* ciphertext CT if it can be decomposed as $\mathsf{CT} = (\mathsf{CT}_{\mathsf{data}}, \mathsf{CT}_{\mathsf{indpt}})$, where $\mathsf{CT}_{\mathsf{data}}$ is called compact data dependent component and $\mathsf{CT}_{\mathsf{indpt}}$ is called as non-compact data independent component. Furthermore, let $m = m_1, \cdots, m_\ell$ be the message encrypted by the ciphertext CT, the compact data dependent component $\mathsf{CT}_{\mathsf{data}}$ can be decomposed as $(\mathsf{CT}_1, \cdots, \mathsf{CT}_\ell)$, where CT_i depends solely on the bit m_i and each CT_i is compact (i.e., the ciphertext size is a polynomial in the security parameter and the message length). More specifically, each component CT_i and $\mathsf{CT}_{\mathsf{indpt}}$ can be represented as the following form.

$$\begin{aligned} \mathsf{CT}_i &= \mathsf{FE.Enc}(\mathsf{MPK}, m_i, r, \hat{r}_i) \qquad \forall i \in [\ell] \\ \mathsf{CT}_{\mathsf{indpt}} &= \mathsf{FE.Enc}(\mathsf{MPK}, r, \hat{r}) \end{aligned}$$

where r is a common randomness used by all components by the encryption algorithm. Apart from the common randomness r, each CT_i may additionally make use of independent randomness $\hat{r}_i$. We note that such a FE scheme with bounded collusions can be solely based on the LWE (or Ring-LWE) assumption [AR16].

4.2 Pseudorandom functions

We rely on the following standard notion of a pseudorandom function family [GGM86], asking that a pseudorandom function be computationally indistinguishable from a truly random function via oracle access.

Definition 4.7 (pseudorandom function)**.** A family $\mathcal{F} = \{\mathsf{PRF}_K : \{0,1\}^{n(\lambda)} \rightarrow \{0,1\}^{m(\lambda)} : K \in \mathcal{K}\}$ of efficiently-computable functions is *pseudorandom* if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\left| \Pr_{K \xleftarrow{\$} \mathcal{K}} \left[\mathcal{A}^{\mathsf{PRF}_K(\cdot)}(1^\lambda) = 1 \right] - \Pr_{\mathsf{R} \xleftarrow{\$} U} \left[\mathcal{A}^{\mathsf{R}(\cdot)}(1^\lambda) = 1 \right] \right| \leq \mathsf{negl}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where U is the set of all functions from $\{0,1\}^{n(\lambda)}$ to $\{0,1\}^{m(\lambda)}$.

4.3 Symmetric Encryption with pseudorandom ciphertexts

A symmetric encryption scheme consists of a tuple of PPT algorithms $(\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$.

- The algorithm $\mathsf{SKE.Setup}$ takes as input a security parameter λ in unary and outputs a key K_E.
- The encryption algorithm $\mathsf{SKE.Enc}$ takes as input a symmetric key K_E and a message m and outputs a ciphertext $\mathsf{SKE.CT}$.
- The decryption algorithm $\mathsf{SKE.Dec}$ takes as input a symmetric key K_E and a ciphertext $\mathsf{SKE.CT}$ and outputs the message m.

In this work, we require a symmetric encryption scheme SKE where the ciphertexts produced by $\mathsf{SKE.Enc}$ are pseudorandom strings. Let $\mathsf{OEnc}_K(\cdot)$ denote the (randomized) oracle that takes as input a message m, chooses a random string r and outputs $\mathsf{SKE.Enc}(K_E, m; r)$. Let $\mathsf{R}_{\ell(\lambda)}(\cdot)$ denote the (randomized) oracle that takes as input a message m and outputs a uniformly random string of length $\ell(\lambda)$ where $\ell(\lambda)$ is the length of the ciphertexts. More formally, we require that for every PPT adversary $\mathcal{A}$ the following advantage is negligible in λ:

$$\mathbf{Adv}_{\mathsf{SKE},\mathcal{A}}(\lambda) = \left| \Pr\left[\mathcal{A}^{\mathsf{OEnc}_{K_E}(\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{R}_{\ell(\lambda)}(\cdot)}(1^\lambda) = 1\right]\right|$$

where the probability is taken over the choice of $K_E \leftarrow \mathsf{SKE.Setup}(1^\lambda)$, and over the internal randomness of the adversary $\mathcal{A}$, the oracle OEnc and $\mathsf{R}_{\ell(\lambda)}$.

We note that such a symmetric encryption scheme with pseudorandom ciphertexts can be constructed from one-way functions, e.g., using weak pseudorandom functions by defining $\mathsf{SKE.Enc}(K_E, m; r) = (r, \mathsf{PRF}_K(r) \oplus m)$.

5 Transformation in the Public Key Setting

In this section we describe the transformation from $\{1, \mathsf{wSel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme to $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme. We first list the building blocks used in the transformation. We denote by our resulting scheme as $\mathsf{pSel} = (\mathsf{pSel.Setup}, \mathsf{pSel.KG}, \mathsf{pSel.Enc}, \mathsf{pSel.Dec})$.

- A fully compact, single-key public-key functional encryption $\mathsf{wSel} = (\mathsf{wSel.Setup}, \mathsf{wSel.KG}, \mathsf{wSel.Enc}, \mathsf{wSel.Dec})$. We require this scheme is weakly selective secure.
- A private-key functional encryption $\mathsf{sSel} = (\mathsf{sSel.Setup}, \mathsf{sSel.KG}, \mathsf{sSel.Enc}, \mathsf{sSel.Dec})$ for single message and many functions. We require this scheme is selectively secure. [6]
- A symmetric encryption scheme with pseudorandom ciphertext $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$.
- A pseudorandom function PRF.

5.1 Construction

We construct the scheme $\mathsf{pSel} = (\mathsf{pSel.Setup}, \mathsf{pSel.KG}, \mathsf{pSel.Enc}, \mathsf{pSel.Dec})$ as follows.

[6] Such scheme can be obtained from semantically secure encryption schemes. More specifically, Gorbunov, Vaikuntanathan and Wee [GVW12] present an adaptively secure one-time bounded FE scheme, which implies an selectively secure one-time bounded FE scheme. This scheme allows to only generate a key for one function, and to encrypt as many messages as the user wishes. [BS15] shows how to transform private-key FE schemes into function-private FE, where messages and functions enjoy the same level of privacy. Therefore, after applying the [BS15] transformation, we can switch the roles of the functions and messages, and obtain a private-key FE scheme which is selectively secure for a single message and many functions.

- **Setup** $\mathsf{pSel.Setup}(1^\lambda)$: On input a security parameter λ in unary, it executes the algorithm $\mathsf{wSel.Setup}(1^\lambda)$ to obtain the key pair $(\mathsf{MPK}_{\mathrm{wSel}}, \mathsf{MSK}_{\mathrm{wSel}})$. The algorithm outputs the public key $\mathsf{MPK}_{\mathrm{pSel}} = \mathsf{MPK}_{\mathrm{wSel}}$ and the master secret key $\mathsf{MSK}_{\mathrm{pSel}} = \mathsf{MSK}_{\mathrm{wSel}}$.

- **Key Generation** $\mathsf{pSel.KG}(\mathsf{MSK}_{\mathrm{pSel}}, f)$: Takes as input a master secret key $\mathsf{MSK}_{\mathrm{pSel}}$ and a function f, it first executes $\mathsf{sSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$. Then it samples a random ciphertext $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$[7] and a random tag $\tau \leftarrow \{0,1\}^{\ell_2(\lambda)}$. It constructs a circuit $G = G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau]$ as described in the figure 1 and then generates a functional secret key $\mathsf{SK}_G \leftarrow \mathsf{wSel.KG}(G, \mathsf{MSK}_{\mathrm{wSel}})$ and a functional secret key $\mathsf{SK}'_f \leftarrow \mathsf{sSel.KG}(\mathsf{MSK}_{\mathrm{sSel}}, f)$. Finally it outputs $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ as the functional secret key.

$G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$

1. If $\beta = 1$, outputs $\mathsf{SKE.Dec}(K_E, C_E)$.
2. Otherwise outputs $\mathsf{CT}_{\mathrm{sSel}} \leftarrow \mathsf{sSel.Enc}\big((\mathsf{MSK}_{\mathrm{sSel}}, m); \mathsf{PRF}_{K_p}(\tau)\big)$.

Figure 1: The circuit $G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau]$

- **Encryption** $\mathsf{pSel.Enc}(m, \mathsf{MPK}_{\mathrm{pSel}})$: Takes as input the message m and the public key $\mathsf{MPK}_{\mathrm{pSel}}$, which is parsed as $\mathsf{MPK}_{\mathrm{wSel}}$. It samples a PRF key $K_p \leftarrow \mathcal{K}$ and outputs the ciphertext $\mathsf{CT}_{\mathrm{pSel}}$ by executing $\mathsf{wSel.Enc}\big(\mathsf{MPK}_{\mathrm{wSel}}, (m, K_p, 0^\lambda, 0)\big)$.

- **Decryption** $\mathsf{pSel.Dec}(\mathsf{SK}_f, \mathsf{CT}_{\mathrm{pSel}})$: On input a functional secret key $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ and the ciphertext $\mathsf{CT}_{\mathrm{pSel}}$, it computes $\mathsf{CT}_{\mathrm{sSel}} \leftarrow \mathsf{wSel.Dec}(\mathsf{CT}_{\mathrm{pSel}}, \mathsf{SK}_G)$ and outputs $f(m) \leftarrow \mathsf{sSel.Dec}(\mathsf{CT}_{\mathrm{sSel}}, \mathsf{SK}_f)$.

The correctness of the above scheme easily follows from the underlying building blocks, and in the remainder of this section we prove the following theorem:

Theorem 5.1. *Assuming that (1) fully-compact, single-key, public-key functional encryption scheme with weakly selective security, (2) selective secure one-ciphertext private-key functional encryption scheme, (3) symmetric encryption with pseudorandom ciphertext and (4) a pseudorandom function family, then there exists a fully-compact, single-key, public-key functional encryption scheme with selective security.*

Proof. We prove by providing a sequence of hybrid arguments described as below. We refer the reader to the formal proof in the full version of this paper since the page limitation.

For security, we consider a sequence of hybrids to prove the above theorem. For simplicity we only consider the one-ciphertext setting and we remark that it is easily generalized to multi-ciphertext setting. We show that any PPT adversary $\mathcal{A}$ succeeds in the selective security game with only negligible advantage. We denote by $\mathbf{Hyb}_{i.b}$ as the ith hybrid argument for $b \in \{0,1\}$ and $\mathbf{Adv}_{i.b}$ is denoted by the probability that the adversary outputs 1 in the hybrid $\mathbf{Hyb}_{i.b}$.

$\underline{\mathbf{Hyb}_{1.b}}$: This corresponds to the real experiment where the challenger encrypts the message m_b, that is, the ciphertext is $\mathsf{CT}_{\mathrm{pSel}} \leftarrow \mathsf{wSel.Enc}\big(\mathsf{MPK}_{\mathrm{wSel}}, (m_b, K_p, 0^\lambda, 0)\big)$.

$\underline{\mathbf{Hyb}_{2.b}}$: For every functional query f, the challenger replaces C_E with a symmetric encryption $\mathsf{SKE.Enc}(K_E, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}} \leftarrow \mathsf{sSel.Enc}\Big((\mathsf{MSK}^*_{\mathrm{sSel}}, m_b); \mathsf{PRF}_{K_p^*}(\tau)\Big)$ (note that each functional secret key has its own different symmetric ciphertext C_E), and K_p^* is a PRF key sampled from the key space $\mathcal{K}$. The symmetric encryption is computed with respect to K_E^* where K_E^* is the output of $\mathsf{SKE.Setup}(1^\lambda)$ and τ is the random tag associated to the functional secret key of f. The same K_E^* and K_p^* are used while generating all the functional secret

[7] The length of C_E is determined as follows. Denote by ℓ_{sSel} be the length of the ciphertext obtained by encrypting a message of length $|m|$, using $\mathsf{sSel.Enc}$. Further, denote by ℓ_1 to be the length of ciphertext obtained by encrypting a message of length ℓ_{sSel}, using $\mathsf{SKE.Dec}$. We set the length of C_E to be ℓ_1

keys, and K_p^* is used in generating the challenge ciphertext $\mathsf{CT}^*_{\mathrm{pSel}} = \mathsf{wSel.Enc}\left(\mathsf{MPK}^*_{\mathrm{wSel}}, (m, K_p^*, 0^\lambda, 0)\right)$. The rest of hybrid is the same as the previous hybrid $\mathbf{Hyb}_{1.b}$. Note that the symmetric key K_E^* is not used for any purpose other than generating the symmetric ciphertext C_E. Therefore, the pseudorandom ciphertexts property of the symmetric encryption scheme implies that $\mathbf{Hyb}_{2.b}$ and $\mathbf{Hyb}_{1.b}$ are indistinguishable.

Lemma 5.1. *Assuming the pseudorandom ciphertexts property of* SKE*, for each $b \in \{0,1\}$, we have*

$$\left|\boldsymbol{Adv}^{\mathcal{A}}_{1.b} - \boldsymbol{Adv}^{\mathcal{A}}_{2.b}\right| \leq \mathsf{negl}(\lambda)$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of SKE. The reduction internally executes the adversary by simulating the role of the challenger in the selective public-key FE game. It answers both the message and the functional queries made by the adversary as follows.

The adversary commits to a pair of messages (m_0, m_1) which is submitted to the reduction. The reduction first obtain a master secret key $\mathsf{MSK}^*_{\mathrm{sSel}}$ by executing $\mathsf{sSel.Setup}(1^\lambda)$, it then samples the PRF key K_p^* from the key space $\mathcal{K}$. Further, the reduction generates $(\mathsf{MPK}_{\mathrm{wSel}}, \mathsf{MSK}_{\mathrm{wSel}})$ which is the output of $\mathsf{wSel.Setup}(1^\lambda)$ and K_E^* which is the output of $\mathsf{SKE.Setup}(1^\lambda)$. The reduction sends back the challenge ciphertext $\mathsf{CT}^*_{\mathrm{pSel}} \leftarrow \mathsf{wSel.Enc}\left(\mathsf{MPK}_{\mathrm{wSel}}, (m_b, K_p^*, 0^\lambda, 0)\right)$. Now the reduction is ready to handle functional secret key queries from the adversary. When the adversary submits a functional query f, the reduction first picks the tag τ at random. The reduction obtains $\mathsf{CT}_{\mathrm{sSel}}$ by executing $\mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\mathrm{sSel}}, m_b); \mathsf{PRF}_{K_p^*}(\tau)\right)$. It then sends $\mathsf{CT}_{\mathrm{sSel}}$ to the challenger of the symmetric encryption scheme. The challenger returns back with C_E, where C_E is either a uniformly random string or it is an encryption of $\mathsf{CT}_{\mathrm{sSel}}$. Then the reduction generates a functional secret key SK_G by executing $\mathsf{wSel.KG}(G[\mathsf{MSK}^*_{\mathrm{sSel}}, C_E, \tau], \mathsf{MSK}_{\mathrm{wSel}})$ and a functional secret key SK'_f by executing $\mathsf{sSel.KG}(\mathsf{MSK}^*_{\mathrm{sSel}}, f)$, then the reduction denotes the tuple $(\mathsf{SK}'_f, \mathsf{SK}_G)$ by SK_f which is sent to the adversary as the functional secret key. The output of the reduction is the same as the output of the adversary.

If the challenger of the symmetric key encryption scheme sends a uniformly random string back to the reduction every time the reduction makes a query to the challenger then we are in $\mathbf{Hyb}_{1.b}$, otherwise we are in $\mathbf{Hyb}_{2.b}$. Since the adversary can distinguish both the hybrids with non-negligible probability, we have that the reduction breaks the security of the symmetric key encryption scheme with non-negligible probability. From our hypothesis, we have that the reduction breaks the security of the symmetric key encryption scheme with non-negligible probability. This proves the lemma. □

<u>$\mathbf{Hyb}_{3.b}$</u>: This is the same as $\mathbf{Hyb}_{2.b}$, except that the challenge ciphertext will be an encryption of $(m_b, 0, K_E, 1)$ instead of $(m_b, K_p, 0^\lambda, 0)$. Note that the functionality of the functional secret keys generated for the function f is not modified while modifying the challenger ciphertext $\mathsf{CT}_{\mathrm{pSel}}$. Therefore, we prove that the weakly selective security implies that $\mathbf{Hyb}_{3.b}$ is indistinguishable from the hybrid $\mathbf{Hyb}_{2.b}$.

Lemma 5.2. *Assuming the weakly selective security of* wSel*, for each $b \in \{0,1\}$, we have*

$$\left|\boldsymbol{Adv}^{\mathcal{A}}_{2.b} - \boldsymbol{Adv}^{\mathcal{A}}_{3.b}\right| \leq \mathsf{negl}(\lambda)$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of wSel. The reduction internally executes the adversary $\mathcal{A}$ by simulating the role of the challenger of the selective FE scheme. It answers both the message and the functional queries made by the adversary as follows.

The adversary first submits a pair of messages (m_0, m_1) to the reduction. The reduction executes the algorithm $\mathsf{sSel.Setup}(1^\lambda)$ to obtain $\mathsf{MSK}^*_{\mathrm{sSel}}$ and then sample a random tag τ. Then it generates a symmetric key K_E^* and a PRF key K_p^*. The reduction computes $C_E = \mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}}$ is the output of $\mathsf{sSel.Enc}\left(\mathsf{MSK}^*_{\text{sSel}}, m_b; \mathsf{PRF}_{K^*_p}(\tau)\right)$, and then it constructs the circuit $G[\mathsf{MSK}^*_{\text{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$. The reduction submits the pair of messages $\big((m_b, K^*_p, 0^\lambda, 0), (m_b, 0, K^*_E, 1)\big)$ along with the function query $G[\mathsf{MSK}^*_{\text{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$ to the challenger of the weakly-selectively secure FE scheme (Note that the underlying weakly selectively secure FE scheme only supports a single-key query). Then the challenger returns back a challenge ciphertext $\mathsf{CT}^*_{\text{wSel}}$ and the functional secret key SK_G to the reduction. The reduction denote $\mathsf{CT}^*_{\text{wSel}}$ by $\mathsf{CT}^*_{\text{pSel}}$ as the challenge ciphertext and sends it to the adversary. Now the reduction is ready to handle the functional secret key queries from the adversary. In the functional secret key query phase, when the adversary submits a function query f, the reduction generates SK'_f by executing $\mathsf{sSel.KG}(\mathsf{MSK}^*_{\text{sSel}}, f)$ and sends back $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ as the functional secret key to the adversary. Finally the adversary outputs a bit b' to guess b and the output of the reduction is the output of the adversary.

We claim that the reduction is a legal adversary in the weak selective security game of wSel, i.e., for challenge message query $\big(M_0 = (m_b, K^*_p, 0^\lambda, 0), M_1 = (m_b, 0^\lambda, K^*_E, 1)\big)$ and every functional query of the form $G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau]$ made by the reduction, we have that $G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau](M_0) = G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau](M_1)$. $G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau](M_0)$ is the functional secret key which is independent of the function f, with respect to the key $\mathsf{MSK}^*_{\text{sSel}}$ and randomness $\mathsf{PRF}_{K^*_p}(\tau)$. Furthermore, $G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau](M_1)$ is the decryption of C_E which is nothing but the encryption of the input message m_b with respect to key $\mathsf{MSK}^*_{\text{sSel}}$ and randomness $\mathsf{PRF}_{K^*_p}(\tau)$. This proves that the reduction is a legal adversary in the weak selective security game.

In conclusion, if the challenger of the weak selective security game sends back an encryption of $(m_b, K^*_p, 0^\lambda, 0)$ then we are in $\mathbf{Hyb}_{2.b}$, otherwise if the challenger encrypts $(m_b, 0^\lambda, K^*_E, 1)$ then we are in $\mathbf{Hyb}_{3.b}$. By our hypothesis, this means the reduction breaks the security of the weak selective security game with non-negligible probability that contradicts the security wSel. This completes the proof of the lemma. □

<u>$\mathbf{Hyb}_{4.b}$</u>: This is the same as $\mathbf{Hyb}_{3.b}$, except that for every function query f made by the adversary, the challenger generates C_E in all the functional secret keys with $\mathsf{SKE.Enc}(K^*_E, \mathsf{CT}_{\text{sSel}})$, where $\mathsf{CT}_{\text{sSel}}$ is the output of $\mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\text{sSel}}, m_b); R\right)$, where R is picked at random. The rest of the hybrid is the same as the previous hybrid. Note that the PRF key K^*_p is not explicitly needed in the previous hybrid, and therefore the pseudorandomness of $\mathcal{F}$ implies that $\mathbf{Hyb}_{4.b}$ is indistinguishable from $\mathbf{Hyb}_{3.b}$. Finally we have that $\mathbf{Hyb}_{4.0}$ is computationally indistinguishable from $\mathbf{Hyb}_{4.1}$ due to the selective security of the underlying sSel scheme.

Lemma 5.3. *Assuming that $\mathcal{F}$ is a pseudorandom function family, for each $b \in \{0,1\}$, we have*

$$\left|\boldsymbol{Adv}^{\mathcal{A}}_{3.b} - \boldsymbol{Adv}^{\mathcal{A}}_{4.b}\right| \leq \mathsf{negl}(\lambda)$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of $\mathcal{F}$. The reduction will internally execute the adversary by simulating the role of the challenger of the selectively secure FE scheme. It answers both the message and the functional queries made by the adversary as follows.

The message queries are answered as in $\mathbf{Hyb}_{3.b}$ and it answers the functional queries made by the adversary as follows. For every functional query f made by the adversary, the reduction picks τ at random which is then forwarded to the challenger of the PRF security game. In response it receives R^*. The reduction then computes C_E to be $\mathsf{SKE.Enc}(K^*_E, \mathsf{CT}_{\text{sSel}})$, where $\mathsf{CT}_{\text{sSel}} = \mathsf{sSel.Enc}(\mathsf{MSK}^*_{\text{sSel}}, m_b; R^*)$. The reduction then proceeds as in the previous hybrids to compute the functional secret key SK_f which it then sends to the adversary $\mathcal{A}$.

If the challenger of the PRF game sent $R^* = \mathsf{PRF}_{K^*_p}(\tau)$ back to the reduction then we are in $\mathbf{Hyb}_{3.b}$ otherwise if R^* is generated at random by the challenger then we are in $\mathbf{Hyb}_{4.b}$. From our hypothesis this means that the probability that the reduction distinguishes the pseudorandom value from random is non-negligible, contradicting the security of the pseudorandom function family. □

Finally we have that the hybrid $\mathbf{Hyb}_{4.0}$ is computationally indistinguishable from $\mathbf{Hyb}_{4.1}$.

Lemma 5.4. *Assuming the selective security of the scheme* sSel*, we have*

$$\left|\boldsymbol{Adv}^{\mathcal{A}}_{4.0} - \boldsymbol{Adv}^{\mathcal{A}}_{4.1}\right| \leq \mathsf{negl}(\lambda)$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of sSel. The reduction internally executes the adversary by simulating the role of the challenger in the selective public-key FE game. It answers both the message and the functional queries made by the adversary as follows.

The adversary first submits a pair of messages (m_0, m_1) which is in turn submitted to the challenger of selective private-key FE, then the challenger returns back an encryption $\mathsf{CT}_{\mathrm{sSel}}$ and then the reduction computes C_E as $C_E = \mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\mathrm{sSel}})$ where K_E^* is the output of $\mathsf{SKE.Setup}(1^\lambda)$. The reduction first generates $\mathsf{MPK}_{\mathrm{wSel}}$ and the symmetric key K_E^* which is the output of $\mathsf{SKE.Setup}(1^\lambda)$, and then it sends back the challenge ciphertext $\mathsf{CT}^*_{\mathrm{pSel}} = \mathsf{wSel.Enc}(\mathsf{MPK}_{\mathrm{wSel}}, (m_b, 0, K_E^*, 1))$. (Note that the challenger could choose either m_0 or m_1 to encrypt since $\beta = 1$ which means that the random bit b is only related to the message encrypted by the challenger of the selective private-key FE. Furthermore, the reduction could construct any $\mathsf{MSK}_{\mathrm{sSel}}$ to construct the circuit G since it has access to the challenger to help him to encrypt the message.) Now the reduction is ready to interact with the adversary $\mathcal{A}$ in the functional secret key query phase. If the adversary submits a function query f, the reduction in turn submits the function f to the challenger and it sends back a functional secret key SK'_f. Now the reduction generates the functional secret key SK_G by it self and sends back $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ to the adversary as the functional secret key. Finally, the reduction outputs what is output by the adversary.

We claim that the reduction is a legal adversary in the selective game of sSel, i.e., for every challenge message query (m_0, m_1), functional query f, we have that $f(m_0) = f(m_1)$ since each functional query made by the adversary of pSel is the same as each functional query made by the reduction and the adversary of pSel os a legal adversary. This proves that the reduction is a legal adversary in the selective game.

In conclusion, if the challenger sends an encryption of m_0 then we are in $\mathbf{Hyb}_{4.0}$ and if the challenger sends an encryption of m_1 then we are in $\mathbf{Hyb}_{4.1}$. From our hypothesis, this means that the reduction breaks the security of sSel. This proves the lemma. □

For efficiency, we prove that our transformation is compact-preserving. Namely, the resulting scheme is also fully-compact. We note that the encryption algorithm of the resulting scheme is the encryption using algorithm $\mathsf{wSel.Enc}$, therefore the compactness of the resulting scheme only depends on the compactness of the underlying weakly selective secure public-key FE scheme wSel. Therefore, if the scheme wSel is compact, then the resulting scheme pSel is also compact. More specifically, we denote the size of a circuit C in a family of circuits $\{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ as $|C|$, and then we have

$$\begin{aligned} |\mathsf{pSel.Enc}| &= |\mathsf{wSel.Enc}| \\ &= \mathrm{poly}(\lambda, \left|(m, K_p, 0^\lambda, 0)\right|) \\ &= \mathrm{poly}(\lambda, |m|) \end{aligned}$$

which proves that our transformation is compact-preserving. □

6 From Non-compact FE to Compact FE via Decomposable and Compact FE Ciphertext

In this section we weaken the input FE scheme of our transformation to be a non-compact one. Namely, we employ the decomposable and compact FE ciphertext property [AR16] to transform a single-key, weakly selective, non-compact FE scheme to a single-key, selective, compact FE scheme.

6.1 Construction

We construct the compact FE scheme $\mathsf{cSel} = (\mathsf{cSel.Setup}, \mathsf{cSel.KG}, \mathsf{cSel.Enc}, \mathsf{cSel.Dec})$ from a non-compact FE scheme as follows.

- **Setup** $\mathsf{cSel.Setup}(1^\lambda, 1^\ell)$: On input a security parameter λ in unary and the message length ℓ in unary, it executes the algorithm $\mathsf{wSel.Setup}(1^\lambda)$ to obtain the key pair $(\mathsf{MPK}_{\mathrm{wSel}}, \mathsf{MSK}_{\mathrm{wSel}})$. The algorithm outputs the public key $\mathsf{MPK}_{\mathrm{cSel}} = \mathsf{MPK}_{\mathrm{wSel}}$ and the master secret key $\mathsf{MSK}_{\mathrm{cSel}} = \mathsf{MSK}_{\mathrm{wSel}}$.
- **Key Generation** $\mathsf{cSel.KG}(\mathsf{MSK}_{\mathrm{cSel}}, f)$: Takes as input a master secret key $\mathsf{MSK}_{\mathrm{cSel}}$ and a function f, it first executes $\mathsf{sSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$. Then it samples a random ciphertext $C_E = (C_E^1, \cdots, C_E^\ell)$, where each $C_E^i \leftarrow \{0,1\}^{\ell=\ell_1(\lambda)}$[8], tags $\tau, \hat{\tau}_1, \cdots, \hat{\tau}_\ell$ and $\hat{\tau}$ uniformly at random from the space of length $\ell_2(\lambda)$ and a PRF key $K \leftarrow \{0,1\}^\lambda$. It constructs a circuit $G = G[\mathsf{MSK}_{\mathrm{sSel}}, C_E^1, \cdots, C_E^\ell, \tau, \hat{\tau}_1, \cdots, \hat{\tau}_\ell]$ as described in the figure 2 and then generates the following:

$$\begin{aligned}\mathsf{SK}_G &\leftarrow \mathsf{wSel.KG}(G[\mathsf{MSK}_{\mathsf{sSel}}, C_E^1, \cdots, C_E^\ell, \tau, \hat{\tau}_1, \cdots, \hat{\tau}_\ell], \mathsf{MSK}_{\mathrm{wSel}})\\ \mathsf{SK}'_f &\leftarrow \mathsf{sSel.KG}(\mathsf{MSK}_{\mathrm{sSel}}, f)\\ \mathsf{CT}_{\mathsf{indpt}} &\leftarrow \mathsf{sSel.Enc}(\mathsf{MSK}_{\mathsf{sSel}}, \mathsf{PRF}_K(\tau); \mathsf{PRF}_K(\hat{\tau}))\end{aligned}$$

  Finally it outputs $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G, \mathsf{CT}_{\mathsf{indpt}})$ as the functional secret key.

$G[\mathsf{MSK}_{\mathsf{sSel}}, C_E^1, \cdots, C_E^\ell, \tau, \hat{\tau}_1, \cdots, \hat{\tau}_\ell](m, K_p, K_E, \beta)$

1. If $\beta = 1$, outputs $\mathsf{SKE.Dec}(K_E, C_E)$.
2. Otherwise it computes

$$\mathsf{CT}_{\mathrm{sSel}}^i \leftarrow \mathsf{sSel.Enc}\left((\mathsf{MSK}_{\mathrm{sSel}}, m^i, \mathsf{PRF}_{K_p}(\tau)); \mathsf{PRF}_{K_p}(\hat{\tau}_i)\right)$$

   for each $i \in [\ell]$. Finally it outputs $\{\mathsf{CT}_{\mathrm{sSel}}^i\}_{i\in[\ell]}$.

Figure 2: The circuit $G[\mathsf{MSK}_{\mathsf{sSel}}, C_E^1, \cdots, C_E^\ell, \tau, \hat{\tau}_1, \cdots, \hat{\tau}_\ell]$

- **Encryption** $\mathsf{cSel.Enc}(m, \mathsf{MPK}_{\mathrm{cSel}})$: Takes as input the message $m = (m_1, \cdots, m_\ell)$ and the public key $\mathsf{MPK}_{\mathrm{cSel}}$, which is parsed as $\mathsf{MPK}_{\mathrm{wSel}}$. It samples a PRF key $K_p \leftarrow \mathcal{K}$ and it computes ciphertexts $\mathsf{CT}_{\mathrm{cSel}} \leftarrow \mathsf{wSel.Enc}\left(\mathsf{MPK}_{\mathrm{wSel}}, (m, K_p, 0^\lambda, 0)\right)$.
- **Decryption** $\mathsf{cSel.Dec}(\mathsf{SK}_f, \mathsf{CT}_{\mathrm{cSel}})$: On input a functional secret key $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G, \mathsf{CT}_{\mathsf{indpt}})$ and the ciphertext $\mathsf{CT}_{\mathrm{cSel}}$ and then it computes $\{\mathsf{CT}_{\mathrm{sSel}}^i\}_{i\in[\ell]} \leftarrow \mathsf{wSel.Dec}(\mathsf{CT}_{\mathrm{cSel}}, \mathsf{SK}_G)$ and outputs

$$f(m) \leftarrow \mathsf{sSel.Dec}\left((\{\mathsf{CT}_{\mathrm{sSel}}^i\}_{i\in[\ell]}, \mathsf{CT}_{\mathsf{indpt}}), \mathsf{SK}'_f\right)$$

[8] The length of C_E is determined as follows. Denote by ℓ_{sSel} be the length of the ciphertext obtained by encrypting a message of length $|m|$, using $\mathsf{sSel.Enc}$. Further, denote by ℓ_1 to be the length of ciphertext obtained by encrypting a message of length ℓ_{sSel}, using $\mathsf{SKE.Dec}$. We set the length of C_E to be ℓ_1

The correctness of the above scheme easily follows from the underlying building blocks, and in the remainder of this section we prove the following theorem:

Theorem 6.1. *Assuming that (1) non-compact, single-key, public-key functional encryption scheme with weakly selective security, (2) selective secure one-ciphertext private-key functional encryption scheme with decomposable and succinct ciphertext property, (3) symmetric encryption with pseudorandom ciphertext and (4) a pseudorandom function family, then there exists a fully-compact, single-key, public-key functional encryption scheme with selective security.*

Proof. We give a proof sketch by providing a sequence of hybrid arguments.

$\underline{\mathbf{Hyb}_{1.b}}$: This corresponds to the real experiment where the challenger encrypts the message $m_b = (m_b^1, \cdots, m_b^\ell)$, that is, the ciphertext is $\mathsf{CT}_{\mathsf{cSel}} = \{\mathsf{CT}^i_{\mathsf{wSel}}\}_{i\in\ell}$, where $\mathsf{CT}^i_{\mathsf{wSel}} \leftarrow \mathsf{wSel.Enc}(\mathsf{MPK}_{\mathsf{wSel}}, (m_b^i, K_p, 0^\lambda, 0))$ for each $i \in [\ell]$.

$\underline{\mathbf{Hyb}_{2.b.i}}$: This is the same as $\mathbf{Hyb}_{1.b}$ except that for every functional query f, the challenger replaces each C_E^i with a symmetric encryption $\mathsf{SKE.Enc}(K_E, \mathsf{CT}^i_{\mathsf{sSel}})$, where $\mathsf{CT}^i_{\mathsf{sSel}} \leftarrow \mathsf{sSel.Enc}((\mathsf{MSK}^*_{\mathsf{sSel}}, m_b^i, \mathsf{PRF}_{K_p^*}(\tau); \mathsf{PRF}_{K_p^*}(\hat{\tau}_i)))$ where K_p^* is a PRF key sampled from the key space $\mathcal{K}$. The symmetric encryption is computed with respect to K_E^* where $K_E^* \leftarrow \mathsf{SKE.Setup}(1^\lambda)$ and τ and each $\hat{\tau}_i$ is the random tag associated to the functional secret key of f. Since the pseudorandom ciphertext property of the symmetric encryption scheme SKE, we have $\mathbf{Hyb}_{1.b} \approx \mathbf{Hyb}_{2.b.1} \approx \cdots \approx \mathbf{Hyb}_{2.b.\ell}$.

$\underline{\mathbf{Hyb}_{3.b}}$: This is the same as the hybrid $\mathbf{Hyb}_{2.b.\ell}$, except that the challenge ciphertext will be an encryption of the tuple $(m_b, 0, K_E, 1)$ instead of $(m_b, K_p, 0^\lambda, 0)$. Note that the functionality of the functional secret keys generated for the function f is not modified while modifying the challenge ciphertext $\mathsf{CT}_{\mathsf{wSel}}$. Therefore, $\mathbf{Hyb}_{3.b} \approx \mathbf{Hyb}_{2.b.\ell}$ due to the weakly selective security of the underlying FE scheme wSel.

$\underline{\mathbf{Hyb}_{4.b.i}}$: The same as the hybrid $\mathbf{Hyb}_{3.b}$ except that for every functional query f made by the adversary, the challenger generates C_E^i in all the functional secret keys with $\mathsf{SKE.Enc}(K_E^*, \mathsf{CT}^i_{\mathsf{sSel}})$, where $\mathsf{CT}^i_{\mathsf{sSel}} \leftarrow \mathsf{sSel.Enc}((\mathsf{MSK}^*_{\mathsf{sSel}}, m_b^i, \mathsf{PRF}_{K_p^*}(\tau); \hat{R}_i))$, where $\hat{R}_i$ is picked at random. The rest of the hybrid is the same as the previous hybrid. We have $\mathbf{Hyb}_{3.b} \approx \mathbf{Hyb}_{4.b.1} \approx, \cdots, \approx \mathbf{Hyb}_{4.b.\ell}$ due to the security of pseudorandom function.

$\underline{\mathbf{Hyb}_{5.b}}$: The same as $\mathbf{Hyb}_{4.b.\ell}$ except that the challenger generates the ciphertext $\mathsf{CT}_{\mathsf{indpt}}$ as $\mathsf{CT}_{\mathsf{indpt}} \leftarrow \mathsf{sSel.Enc}(\mathsf{MSK}_{\mathsf{sSel}}, \mathsf{PRF}_{K_p^*}(\tau); \hat{R})$, where $\hat{R}$ is picked at random. We have $\mathbf{Hyb}_{4.b.\ell} \approx \mathbf{Hyb}_{5.b}$ due to the security of PRF.

$\underline{\mathbf{Hyb}_{6.b}}$: The same as $\mathbf{Hyb}_{5.b}$ except that the challenger generates the ciphertext $\mathsf{CT}_{\mathsf{indpt}}$ as $\mathsf{CT}_{\mathsf{indpt}} \leftarrow \mathsf{wSel.Enc}(\mathsf{MSK}_{\mathsf{wSel}}, R; \hat{R})$, where R is picked at random. Therefore, we have $\mathbf{Hyb}_{5.b} \approx \mathbf{Hyb}_{6.b}$ due to the security of PRF.

Finally, $\mathbf{Hyb}_{6.0}$ is computationally indistinguishable from $\mathbf{Hyb}_{6.1}$ based on the selective security of the one-ciphertext private key functional encryption scheme sSel. □

Corollary 6.1. *Assuming the existence of one-way functions and the hardness of LWE assumption, if there exists a single-key, weakly selective secure, non-compact* FE *scheme, then there exists a single-key, selective secure, compact* FE *scheme (with only polynomial security loss).*

7 Implications to Indistinguishability Obfuscation

Here we present the implications of our main result to general-purpose indistinguishability obfuscator. We first recall the main result described in [AJ15, BV15].

We first provide some theorems adapted from existing results:

Theorem 7.1 ([AJ15, BV15]). *Public-key (weakly) compact* FE *for* NC^1 *with sub-exponential security in the selective security model for single-key query implies* $i\mathcal{O}$ *for* P/poly.

Theorem 7.2. *Assuming the existence of one-way functions, any multi-key, adaptively secure, width compact* FE *can be polynomially reduced to a single-key, weakly selective secure, weakly compact* FE.

Combining theorem 7.1 with the corollary 6.1, we obtain the following corollary.

Corollary 7.1. *Assuming the existence of one-way functions and the hardness of LWE assumption, if there exists a single-key, weakly selective, non-compact* FE *with sub-exponential security, then there exists an indistinguishability obfuscator for* P/poly.

Combining theorem 7.2 with the corollary 6.1, we obtain the following corollary.

Corollary 7.2. *Assuming the existence of one-way functions and the hardness of LWE assumption, any multi-key, adaptively-secure, width-compact scheme can be polynomially reduced to a single-key, weakly selective secure, non-compact* FE *scheme.*

References

[AB15] Benny Applebaum and Zvika Brakerski. Obfuscating circuits via composite-order graded encoding. In *Theory of Cryptography Conference*, pages 528–556. Springer, 2015.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *Annual Cryptology Conference*, pages 657–677. Springer, 2015.

[ADGM16] Daniel Apon, Nico Döttling, Sanjam Garg, and Pratyay Mukherjee. Cryptanalysis of indistinguishability obfuscations of circuits over ggh13. Cryptology ePrint Archive, Report 2016/1003, 2016. http://eprint.iacr.org/2016/1003.

[AGVW13] Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *Advances in Cryptology–CRYPTO 2013*, pages 500–518. Springer, 2013.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Annual Cryptology Conference*, pages 308–326. Springer, 2015.

[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.

[AR16] Shweta Agrawal and Alon Rosen. Functional encryption for bounded collusions, revisited. Cryptology ePrint Archive, Report 2016/361, 2016. http://eprint.iacr.org/2016/361.

[AS16a] Prabhanjan Ananth and Amit Sahai. Functional encryption for turing machines. In *Theory of Cryptography Conference*, pages 125–153. Springer, 2016.

[AS16b] Prabhanjan Ananth and Amit Sahai. Projective arithmetic functional encryption and indistinguishability obfuscation from degree-5 multilinear maps. Cryptology ePrint Archive, Report 2016/1097, 2016. http://eprint.iacr.org/2016/1097.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Advances in cryptologyâĂŤCRYPTO 2001*, pages 1–18. Springer, 2001.

[BGJ$^+$16] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 345–356. ACM, 2016.

[BGK$^+$14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 221–238. Springer, 2014.

[BGL$^+$15] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. Succinct randomized encodings and their applications. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 439–448. ACM, 2015.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a nash equilibrium. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1480–1498. IEEE, 2015.

[BR14] Zvika Brakerski and Guy N Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. In *Theory of Cryptography Conference*, pages 1–25. Springer, 2014.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In *Theory of Cryptography Conference*, pages 306–324. Springer, 2015.

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

[BV15] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 171–190. IEEE, 2015.

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

[CHJV15] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. Succinct garbling and indistinguishability obfuscation for ram programs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 429–437. ACM, 2015.

[CHL$^+$15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Advances in Cryptology–EUROCRYPT 2015*, pages 3–12. Springer, 2015.

[CLLT16] Jean-SÃľbastien Coron, Moon Sung Lee, TancrÃĺde Lepoint, and Mehdi Tibouchi. Zeroizing attacks on indistinguishability obfuscation over clt13. Cryptology ePrint Archive, Report 2016/1011, 2016. http://eprint.iacr.org/2016/1011.

[CLP15] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In *Annual Cryptology Conference*, pages 287–307. Springer, 2015.

[CLT13] Jean-Sébastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology–CRYPTO 2013*, pages 476–493. Springer, 2013.

[CLT15] Jean-Sebastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. New multilinear maps over the integers. Technical report, Cryptology ePrint Archive, Report 2015/162, 2015. http://eprint. iacr. org, 2015.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49. IEEE, 2013.

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

[GHMS14] Craig Gentry, Shai Halevi, Hemanta K Maji, and Amit Sahai. Zeroizing without zeroes: Cryptanalyzing multilinear maps without encodings of zero. *IACR Cryptology ePrint Archive*, 2014:929, 2014.

[GKP$^+$13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 555–564. ACM, 2013.

[GKW16] Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. Cryptology ePrint Archive, Report 2016/317, 2016. http://eprint.iacr.org/2016/317.

[GMS16] Sanjam Garg, Pratyay Mukherjee, and Akshayaram Srinivasan. Obfuscation without the vulnerabilities of multilinear maps. Technical report, Cryptology ePrint Archive, Report 2016/390, 2016. http://eprint. iacr. org, 2016.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *Annual Cryptology Conference*, pages 579–604. Springer, 2016.

[GR07] Shafi Goldwasser and Guy N Rothblum. On best-possible obfuscation. In *Theory of Cryptography Conference*, pages 194–213. Springer, 2007.

[GS16] Sanjam Garg and Akshayaram Srinivasan. Single-key to multi-key functional encryption with polynomial loss. In *Theory of Cryptography Conference*, pages 419–442. Springer, 2016.

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

[HJO$^+$15] Brett Hemenway, Zahra Jafargholi, Rafail Ostrovsky, Alessandra Scafuro, and Daniel Wichs. Adaptively secure garbled circuits from one-way functions. Technical report, IACR Cryptology ePrint Archive, 2015: 1250, 2015.

[KLW15] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. Indistinguishability obfuscation for turing machines with unbounded memory. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 419–428. ACM, 2015.

[Lin16a] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 28–57. Springer, 2016.

[Lin16b] Huijia Lin. Indistinguishability obfuscation from ddh on 5-linear maps and locality-5 prgs. Cryptology ePrint Archive, Report 2016/1096, 2016. http://eprint.iacr.org/2016/1096.

[LM16a] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. In *Theory of Cryptography Conference*, pages 443–468. Springer, 2016.

[LM16b] Baiyu Li and Daniele Micciancio. Compactness vs collusion resistance in functional encryption. Cryptology ePrint Archive, Report 2016/561, 2016. http://eprint.iacr.org/2016/561.

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. Technical report, Cryptology ePrint Archive, Report 2016/147, 2016.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. *IACR Cryptology ePrint Archive*, 2010:556, 2010.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *International Cryptology Conference*, pages 500–517. Springer, 2014.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 463–472. ACM, 2010.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 475–484. ACM, 2014.

[Wat13] Brent Waters. Functional encryption: origins and recent developments. In *Public-Key Cryptography–PKC 2013*, pages 51–54. Springer, 2013.

Appendices

Appendix A Preliminaries (Cont.)

A.1 Private-Key Functional Encryption

A private-key functional encryption scheme SKFE over a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a tuple $(\mathsf{SKFE.Setup}, \mathsf{SKFE.KG}, \mathsf{SKFE.Enc}, \mathsf{SKFE.Dec})$ of PPT algorithms with the following properties.

- $\mathsf{SKFE.Setup}(1^\lambda)$: The setup algorithm takes as input the unary representation of the security parameter, and outputs a master secret key MSK.
- $\mathsf{SKFE.KG}(\mathsf{MSK}, f)$: The key generation algorithm takes as input a secret key MSK and a function $f \in \mathcal{F}_\lambda$ and outputs a functional secret key SK_f.
- $\mathsf{SKFE.Enc}(\mathsf{MSK}, m)$: The encryption algorithm takes as input a master secret key MSK and a message $m \in \mathcal{M}_\lambda$, and outputs a ciphertext CT.
- $\mathsf{SKFE.Dec}(\mathsf{SK}_f, \mathsf{CT})$: The decryption algorithm takes as input a functional secret key SK_f and a ciphertext CT, and outputs $m \in \mathcal{M}_\lambda \cup \{\perp\}$

We say a private-key functional encryption scheme is defined for a complexity class $\mathcal{C}$ if it supports all the functions that can be implemented in $\mathcal{C}$.

Correctness. We require that there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all sufficiently large $\lambda \in \mathbb{N}$, for every message $m \in \mathcal{M}_\lambda$, and for every function $f \in \mathcal{F}_\lambda$ we have

$$\Pr[\mathsf{SKFE.Dec}(\mathsf{SKFE.KG}(\mathsf{MSK}, f), \mathsf{SKFE.Enc}(\mathsf{MSK}, m)) = f(m)] \geq 1 - \mathsf{negl}(\lambda)$$

where $\mathsf{MSK} \leftarrow \mathsf{SKFE.Setup}(1^\lambda)$, and the probability is taken over the random choices of all algorithms.

Security. We consider the standard (weakly) selective indistinguishability-based notions for private-key functional encryption as shown in the work of Brakerski and Segev [BS15]. Intuitively, these notions ask that encryptions of any two messages, m_0 and m_1, should be computationally indistinguishable given access to functional secret keys for any function f such that $f(m_0) = f(m_1)$. In the case of selective security, adversaries are required to specify the two messages in advance (i.e., before interacting with the system).

Definition A.1 (Weakly Selective Security)**.** A private-key functional encryption scheme SKFE over a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is *weak selective secure* if for any PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathbf{Adv}^{\mathrm{wSel}}_{\mathrm{skfe},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, 1) = 1]\right| \leq \mathrm{negl}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for each $b \in \{0, 1\}$ the experiment $\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, b)$, modeled as a game between the adversary $\mathcal{A}$ and a challenger, is defined as follows:

1. **Challenge Phase**: The adversary $\mathcal{A}$ outputs two messages (m_0, m_1) such that $|m_0| = |m_1|$ and a set of functions $f_1, \cdots, f_q \in \mathcal{F}$ to the challenger. The parameter q and the size of message vectors are apriori-unbounded.

2. The challenger generates $\mathsf{MSK} \leftarrow \mathsf{SKFE.Setup}(1^\lambda)$ and generates the challenger ciphertext $\mathsf{CT} \leftarrow \mathsf{SKFE.Enc}(\mathsf{MSK}, m_b)$. The challenger also computes $\mathsf{SK}_{f,i} \leftarrow \mathsf{SKFE.KG}(\mathsf{MSK}, f_i)$ for all $i \in [q]$. It then sends CT and $\{\mathsf{SK}_{f,i}\}_{i\in[q]}$ to the adversary $\mathcal{A}$.

3. If $\mathcal{A}$ makes a query f_j for some $j \in [q]$ to functional secret key generation oracle such that $f_j(m_0) \neq f_j(m_1)$, the output of the experiment is $\perp$. Otherwise the output is b' which is the output of $\mathcal{A}$

Remark. We say that the functional encryption scheme SKFE is *single-key*, *weakly selective secure* if the adversary $\mathcal{A}$ in $\mathsf{Exp}^{\mathrm{wSel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, b)$ is allowed to obtain the functional secret key for a single function f.

Definition A.2 (Selective Security)**.** A private-key functional encryption scheme SKFE over a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ and a message space $\mathcal{M} = \{\mathcal{M}_\lambda\}_{\lambda\in\mathbb{N}}$ is *selectively secure* if for any PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

$$\mathbf{Adv}^{\mathrm{Sel}}_{\mathrm{skfe},\mathcal{A}}(\lambda) = \left|\Pr[\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, 1) = 1]\right| \leq \mathrm{negl}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for each $b \in \{0, 1\}$ the experiment $\mathsf{Exp}^{\mathrm{Sel}}_{\mathrm{skfe},\mathcal{A}}(\lambda, b)$, modeled as a game between the adversary $\mathcal{A}$ and a challenger, is defined as follows:

1. **Setup Phase**: The challenger samples $\mathsf{MSK} \leftarrow \mathsf{SKFE.Setup}(1^\lambda)$.

2. **Message Queries**: On input 1^λ the adversary submits $(m_1^{(0)}, \cdots, m_p^{(0)}), (m_1^{(1)}, \cdots, m_p^{(1)})$ for some polynomial $p = p(\lambda)$. The challenger replies with $(c_1, \cdots, c_p)$, where $c_i \leftarrow \mathsf{SKFE.Enc}(\mathsf{MSK}, m_i^{(b)})$ for every $i \in [p]$.

3. **Function Queries**: The adversary adaptively queries the challenger with any function $f \in \mathcal{F}_\lambda$ such that $f(m_i^{(0)}) = f(m_i^{(1)})$ for every $i \in [p]$. For each such query, the challenger replies with $\mathsf{SK}_f \leftarrow \mathsf{SKFE.KG}(\mathsf{MSK}, f)$.

4. **Output Phase**: The adversary outputs a bit b' which is defined as the output of the experiment.

Efficiency. We now define the efficiency requirements of a SKFE scheme.

Definition A.3 (Fully Compact)**.** A private-key functional encryption scheme SKFE is said to be fully compact if for all security parameter $\lambda \in \mathbb{N}$ and for all message $m \in \{0,1\}^*$ the running time of the encryption algorithm $\mathsf{SKFE.Enc}$ is $\mathsf{poly}(\lambda, |m|)$.

Definition A.4 (Weakly Compact)**.** A private-key functional encryption scheme SKFE is said to be weakly compact if for all security parameter $\lambda \in \mathbb{N}$ and for all message $m \in \{0,1\}^*$ the running time of the encryption algorithm $\mathsf{SKFE.Enc}$ is $s^\gamma \cdot \mathsf{poly}(\lambda, |m|)$, where $\gamma < 1$ is a constant and $s = \max_{f \in \mathcal{F}} |C_f|$, where C_f is a circuit implementing the function f.

A private-key functional encryption scheme is said to be *non-compact* if the running time of the encryption algorithm can depend arbitrarily on the maximum circuit size of the function family.

Appendix B Transformation in the Private-Key Setting

In this section we describe the transformation from weakly selective security to selective security in the private-key functional encryption scheme. The only difference from the public-key setting described in the section 5 is that there is only one master key $\mathsf{MSK}_{\mathrm{wSel}}$ which acts as either an encryption key or a master secret key. Note that we will use the same notation as described in the section 5, except that $\mathsf{pSel} = (\mathsf{pSel.Setup}, \mathsf{pSel.KG}, \mathsf{pSel.Enc}, \mathsf{pSel.Dec})$ represents a selectively-secure *private-key* functional encryption scheme and $\mathsf{wSel} = (\mathsf{wSel.Setup}, \mathsf{wSel.KG}, \mathsf{wSel.Enc}, \mathsf{wSel.Dec})$ represents a weakly selectively-secure *private-key* functional encryption scheme.

B.1 Construction

We construct the private-key functional encryption scheme $\mathsf{pSel} = (\mathsf{pSel.Setup}, \mathsf{pSel.KG}, \mathsf{pSel.Enc}, \mathsf{pSel.Dec})$ as follows.

Setup $\mathsf{pSel.Setup}(1^\lambda)$: On input a security parameter λ in unary, it executes the algorithm $\mathsf{wSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\mathrm{wSel}}$. The algorithm outputs the master secret key $\mathsf{MSK}_{\mathrm{pSel}} = \mathsf{MSK}_{\mathrm{wSel}}$.

Key Generation $\mathsf{pSel.KG}(\mathsf{MSK}_{\mathrm{pSel}}, f)$: Takes as input a master secret key $\mathsf{MSK}_{\mathrm{pSel}}$ and a function f, it first executes $\mathsf{sSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$. Then it samples a random ciphertext $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$ and a random tag $\tau \leftarrow \{0,1\}^{\ell_2(\lambda)}$. It constructs a circuit $G = G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau]$ as described in the figure 1 and then generates a functional secret key $\mathsf{SK}_G \leftarrow \mathsf{wSel.KG}(G, \mathsf{MSK}_{\mathrm{wSel}})$ and a functional secret key $\mathsf{SK}'_f \leftarrow \mathsf{sSel.KG}(\mathsf{MSK}_{\mathrm{sSel}}, f)$. Finally it outputs $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ as the functional secret key.

Encryption $\mathsf{pSel.Enc}(m, \mathsf{MSK}_{\mathrm{pSel}})$: Takes as input the message m and the master secret key $\mathsf{MSK}_{\mathrm{pSel}}$, which is parsed as $\mathsf{MSK}_{\mathrm{wSel}}$. It samples a PRF key $K_p \leftarrow \mathcal{K}$ and outputs the ciphertext $\mathsf{CT}_{\mathrm{pSel}} \leftarrow \mathsf{wSel.Enc}\left(\mathsf{MSK}_{\mathrm{wSel}}, (m, K_p, 0^\lambda, 0)\right)$.

Decryption $\mathsf{pSel.Dec}(\mathsf{SK}_f, \mathsf{CT}_{\mathrm{pSel}})$: On input a functional secret key $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ and the ciphertext $\mathsf{CT}_{\mathrm{pSel}}$, it computes $\mathsf{CT}_{\mathrm{sSel}} \leftarrow \mathsf{wSel.Dec}(\mathsf{CT}_{\mathrm{pSel}}, \mathsf{SK}_G)$ and outputs $f(m) \leftarrow \mathsf{sSel.Dec}(\mathsf{CT}_{\mathrm{sSel}}, \mathsf{SK}_f)$.

```
$G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$
1. If $\beta = 1$, outputs $\mathsf{SKE.Dec}(K_E, C_E)$.
2. Otherwise outputs $\mathsf{CT}_{\text{sSel}} \leftarrow \mathsf{sSel.Enc}\left((\mathsf{MSK}_{\text{sSel}}, m); \mathsf{PRF}_{K_p}(\tau)\right)$.
```

Figure 3: The circuit $G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau]$

The correctness of the above scheme easily follows from the underlying building blocks, and in the remainder of this section we prove the following theorem:

Theorem B.1. *Assuming that (1) fully compact, single-key, weakly selectively secure private-key functional encryption scheme, (2) one-ciphertext selectively secure private-key functional encryption scheme, (3) symmetric encryption with pseudorandom ciphertext and (4) a pseudorandom function family, then there exists a fully compact, bounded-key(≥ 1 key queries), selectively-secure private-key functional encryption scheme.*

Proof. The proof in the private-key setting is essentially the same as that in the public-key setting. Therefore we will omit the proof details and just give the description of each hybrid arguments.

For security, we only give a proof sketch by listing the transformations in each hybrid arguments.

$\mathbf{Hyb}_{1.b}$: This corresponds to the real experiment where the challenger encrypts the message m_b, that is, $\mathsf{CT}_{\text{pSel}}$ is obtained by executing $\mathsf{wSel.Enc}\left(\mathsf{MSK}_{\text{wSel}}, (m_b, K_p, 0^\lambda, 0)\right)$.

$\mathbf{Hyb}_{2.b}$: For every functional query f, the challenger replaces C_E with a symmetric encryption $\mathsf{SKE.Enc}(K_E, \mathsf{CT}_{\text{sSel}})$, where $\mathsf{CT}_{\text{sSel}} \leftarrow \mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\text{sSel}}, m_b); \mathsf{PRF}_{K_p^*}(\tau)\right)$ (note that each functional secret key has its own different C_E), and K_p^* is a PRF key sampled from the key space $\mathcal{K}$. The symmetric encryption is computed with respect to K_E^* where K_E^* is the output of $\mathsf{SKE.Setup}(1^\lambda)$ and τ is the random tag associated to the functional secret key of f. The same K_E^* and K_p^* are used while generating all the functional secret keys, and K_p^* is used generating the challenge ciphertext $\mathsf{CT}^*_{\text{pSel}} = \mathsf{wSel.Enc}\left(\mathsf{MSK}^*_{\text{wSel}}, (m, K_p^*, 0^\lambda, 0)\right)$. The rest of hybrid is the same as the previous hybrid $\mathbf{Hyb}_{1.b}$. Note that the symmetric key K_E^* is not used for any purpose other than generating the values C_E.

Therefore, the pseudorandom ciphertexts property of the symmetric encryption scheme implies that $\mathbf{Hyb}_{2.b}$ and $\mathbf{Hyb}_{1.b}$ are indistinguishable.

$\mathbf{Hyb}_{3.b}$: This is the same as $\mathbf{Hyb}_{2.b}$, except that the challenge ciphertext will be an encryption of $(m_b, 0, K_E, 1)$ instead of $(m_b, K_p, 0^\lambda, 0)$. Note that the functionality of the functional secret keys generated for the function f is not modified while modifying the challenger ciphertext $\mathsf{CT}_{\text{pSel}}$. Therefore, we prove that the weakly selective security implies that $\mathbf{Hyb}_{3.b}$ is indistinguishable from the hybrid $\mathbf{Hyb}_{2.b}$.

$\mathbf{Hyb}_{4.b}$: For every function query f made by the adversary, the challenger generates C_E in all the functional secret keys with $\mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\text{sSel}})$, where $\mathsf{CT}_{\text{sSel}}$ is the output of $\mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\text{sSel}}, x_b); R\right)$, where R is picked at random. The rest of the hybrid is the same as the previous hybrid. Note that the PRF key K_p^* is not explicitly needed in the previous hybrid.

Therefore the pseudorandomness of $\mathcal{F}$ implies that $\mathbf{Hyb}_{4.b}$ is indistinguishable from $\mathbf{Hyb}_{3.b}$.

Finally we can prove that $\mathbf{Hyb}_{4.0}$ is computationally indistinguishable from $\mathbf{Hyb}_{4.1}$ based on the selective security of the one-ciphertext private key functional encryption scheme. This finishes the security proof.

For efficiency, we prove that our transformation is compact-preserving. Namely, the resulting scheme is also fully compact. We note that the encryption algorithm of the resulting scheme is the encryption using algorithm $\mathsf{wSel.Enc}$, therefore the compactness of the resulting scheme only depends on the compactness of the underlying weakly selectively secure private-key FE scheme wSel. Therefore, if the scheme wSel is compact, then the resulting scheme pSel is also compact. More specifically, we denote the size of a circuit C in a family of circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ as $|C|$, and then we have

$$\begin{aligned}|\mathsf{pSel.Enc}| &= |\mathsf{wSel.Enc}|\\ &= \mathrm{poly}(\lambda, \left|(m, K_p, 0^\lambda, 0)\right|)\\ &= \mathrm{poly}(\lambda, |m|)\end{aligned}$$

which proves that our transformation is compact-preserving. □