# From Weakly Selective to Selective Security in Compact Functional Encryption

Linfeng Zhou*

October 7, 2016

**Abstract**

Functional Encryption (FE) generalizes the notion of traditional encryption system by providing fine-grained access control. In a functional encryption scheme, the owner of the secret key can generate restricted functional keys that allow users to obtain specific functions over the encrypted messages and nothing else.

In this work, we show a generic transformation from weakly selective secure functional encryption to selectively secure functional encryption and this transformation preserves the compactness of the FE scheme. This result is given by reusing techniques in the work of Ananth et al.(CRYPTO 2015) through a modified approach. Furthermore, combining recent results, we remark that this result gives an alternative approach of recent work by Garg and Srinivasan (TCC-B 2016). Namely, a single-key weakly selective secure functional encryption scheme, whose ciphertext size is *sublinear* in the size of the function for which the functional key is issued, implies all other notions of functional encryption generically.

## 1 Introduction

The disclosure of confidential data (e.g., financial data, user profile and medical data) has been a central problem in recent era. This problem has been a top concern when more and more applications move to cloud computing platforms and these disclosures often happen when some untrustworthy party is handling confidential data. Therefore, ensuring data confidentiality on third-party servers that may be untrustworthy is currently a top concern.

A very beautiful approach to solve this problem is *functional encryption*[LOS+10, BSW11, BSW12], which is a generalized notion of predicate encryption[KSW08, GVW15], attribute-based encryption[GPSW06, BSW07, Wat11] and identity-based encryption [Sha84, BF01], in which each of them provides different levels of access to the underlying plaintext. In the model of functional encryption, anyone could encrypt data with a public key MPK and the holder of the master secret key can provide functional keys $\mathsf{SK}_f$ for a function $f$. To decrypt the ciphertext CT for the plaintext $m$, anyone could decrypt it if he has access to the function key $\mathsf{SK}_f$ and obtain a computation over the plaintext $f(m)$.

The security of this class of systems is captured by an indistinguishability-based security (IND) game between a challenger and an adversary. In this game the challenger will first generate the public key which is sent to the adversary. The adversary begins by opening the first key query phase where it will issue a polynomial number of key queries, each for a function $f$ in the corresponding family of functions. For each query the adversary receives back a functional key $\mathsf{SK}_f$ corresponding to the function $f$. Next the adversary submits two challenge messages $m_0, m_1$ with the restriction that $f(m_0) = f(m_1)$ for all functions $f$ queried

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

before. The challenger will flip a coin $b \in \{0,1\}$ and return a challenge ciphertext $\mathsf{CT}^*$ encrypting $m_b$. Next, the adversary will engage in a second set of private key queries with the same restrictions. Finally, it will output a guess $b' \in \{0,1\}$ and win if $b = b'$. For any secure scheme the advantage of the adversary to win the game should be negligible.

The above game, called *adaptive* security (Adp) game, captures the intuitive notion of what an IND-based security game should look like. Namely, that an adversary cannot distinguish between two messages unless he receives keys that trivially allow him to, even though the adversary is allowed to adaptively choose what the keys and messages are. However, there is no such thing as a free lunch. Research in recent years demonstrates that it is difficult to achieve adaptive security if we not only want to restrict ourselves to polynomial loss in the reductions, but also strive for a new functionality while avoid relying on sub-exponential hardness assumptions. To ease the initial pathway people often consider security under a weaker notion of *selective* (Sel) security where the adversary is forced to commit the challenge messages $(m_0, m_1)$ before seeing the public key. Furthermore, it is possible to further weaken the security notion of FE to weakly selective security (wSel), where the adversary must commit not only to the challenge messages $(m_0, m_1)$, but also to all the functional queries before seeing the public key. Thus, we can move from selective to adaptive security in functional encryption scheme.

Furthermore, we also consider the size of the encryption circuit (i.e., the size of the ciphertexts) which captures the central notion of efficiency of functional encryption. The most basic efficiency notion of FE scheme is the *non-compactness*. A FE scheme is said to have *non-compact* ciphertexts (NC) if the size of the encryption circuit can depend arbitrarily on the circuit size of the functions in the function family $\mathcal{F}$. There are several relaxations to the efficiency notion that have been considered in literature where the size of the ciphertext not only depends on the size of the plaintext but also (somewhat) on the circuit size of the functions in the corresponding function family of the functional encryption. A FE scheme supporting function family $\mathcal{F}$ has *weakly compact* ciphertexts (WC) if the size of the encryption circuit grows *sub-linearly* with the maximum circuit size of functions in the function family $\mathcal{F}$. The most strong efficiency notion of FE scheme is called *full compactness*. A FE scheme is said to be *fully compact* (FC) 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.

The number of functional key queries and the number of challenge ciphertexts are also essential parameters considered in the FE system. More specifically, FE system can be parameterized based on whether the adversary obtains bounded (Bou) or unbounded (Unb) number of functional key queries and whether she is allowed to query bounded or unbounded number of challenge ciphertexts.

In this work we use the notation in the work of Garg and Srinivasan[GS16] for the convenience of describing different notions of functional encryption scheme. Namely, we represent security and efficiency notions of IND-based FE scheme in the form as $\{qq, sss, ee\}$-IND-FE, where $qq$ denotes the number of functional keys obtained by the adversary, i.e., $qq \in \{1, \mathsf{Bou}, \mathsf{Unb}\}$; $sss$ refers to the security setting considered i.e., $\{\mathsf{wSel}, \mathsf{Sel}, \mathsf{Adp}\}$ setting and $ee$ denotes the efficiency of the scheme, that is, $\{\mathsf{FC}, \mathsf{WC}, \mathsf{NC}\}$. Furthermore, we occasionally denote $ee$ by WidC, which means the IND-based FE scheme is *width compact*. A FE scheme supporting function family $\mathcal{F}$ has *width compact* ciphertexts (WidC) if the size of the encryption circuit grows with the *width* of circuits in the function family $\mathcal{F}$. The focus of this work is studying the relationship between different notions of IND-based security of FE scheme. It can be easily seen from a standard hybrid argument that $\{qq, sss, ee\}$-IND-FE with one challenge ciphertext implies $\{qq, sss, ee\}$-IND-FE with unbounded challenge ciphertexts. Hence in the rest of the introduction we focus on the case where the adversary obtains a single challenge ciphertext.

**Prior Work**. Ananth et al.[ABSV15] introduced a transformation from $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{NC}\}$-IND-FE to a $\{\mathsf{Unb}, \mathsf{Adp}, \mathsf{NC}\}$-IND-FE. However, even though their transformation preserves the property of unbounded functional key queries, it doesn't preserve compactness property even if the input scheme is a fully-compact one. Ananth and Jain[AJ15] and Bitansky and Vaikuntanathan [BV15] showed that $\{1, \mathsf{wSel}, \mathsf{WC}\}$-IND-FE implies Indistinguishability Obfuscation ($i\mathcal{O}$)[BGI+12] which in turn implies $\{\mathsf{Unb}, \mathsf{Adp}, \mathsf{FC}\}$-IND-FE as shown by Ananth and Sahai [AS16]. However, the transformation from $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ to $i\mathcal{O}$ suffers an exponential security loss. Namely, the transformation is actually starting from *sub-exponentially secure* $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$. Ananth, Jain and Sahai [AJS15] and Bitansky and Vaikuntanathan [BV15] gave a generic transformation from $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{NC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ to $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$. This transformation requires the input non-compact scheme satisfying the security against unbounded collusions and the output of the transformation must be selectively secure no matter what the security of the input FE scheme satisfies. Recently, Li and Micciancio [LM16] proved that polynomially hard weakly compact $\mathsf{FE}$ scheme implies multi-key $\mathsf{FE}$ scheme, which in turn implies compact functional encryption scheme. However, their transformation only supports boosting from single-key to multi-key FE. Namely, if the input $\mathsf{FE}$ scheme of the transformation is weakly compact (resp., weakly selective secure), then the resulting scheme is also weakly compact (resp. weakly selective secure). We remark that our transformation actually solves one of these gaps of their transformation, and this is the key point to obtain a new approach from single-key weakly selective, weakly compact FE to other notions. Furthermore, Garg and Srinivasan[GS16] introduced a transformation from $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ to $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ with only polynomial security loss and their transformation in turn implies all other notions. We remark that this work provide an alternative approach of their results. More recently, Goyal, Koppula and Waters [GKW16] proposed a new way which can transform any selectively secure $\mathsf{FE}$ scheme to a semi-adaptive one, and we further noticed that this transformation is compact-preserving (this property is not mentioned by the authors in the original paper but their transformation indeed satisfies this property). Namely, if the input scheme is fully compact, then the resulting scheme is also fully compact.

## 1.1 Our Contributions

In this work, we made the following two contributions.

- Contribution 1: We give a generic transformation from $\{1, \mathsf{wSel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme to $\{\mathsf{Bou}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme with only *polynomial security loss*.
- Contribution 2: Combining our result above with existing results, we obtain an *alternative approach*, in contrast with the work of Garg and Srinivasan [GS16], from $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme to $\{\mathsf{Unb}, \mathsf{Adp}, \mathsf{WidC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme by the following steps. (Even though we note that the resulting scheme is always bounded-key no matter the input scheme is single-key or multi-key, we remark that our transformation suffices to give an alternative approach.)

  1. Applying the generic transformation in [LM16] we obtain a $\{\mathsf{Unb}, \mathsf{wSel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme starting from a $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme[1].
  2. Applying the generic transformation of us described as contribution 1, we obtain a $\{\mathsf{Bou}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme. [2]
  3. Applying the generic transformation in [LM16] again we obtain a $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme.
  4. Applying a generic transformation from selective to adaptive security, we obtain a $\mathsf{FE}$ scheme that is adaptively secure against unbounded collusions. We note that there are two approaches to do this.
     - Applying the generic transformation from selective to adaptive security of [AS16, ABSV15] we obtain a $\mathsf{FE}$ scheme that is adaptively secure against unbounded collusions.
     - Another way to do so is firstly applying the generic transformation of [GKW16] we obtain a $\mathsf{FE}$ scheme that is semi-adaptively secure (note that this transformation is compact-preserving even though this property is not mentioned in the original paper), and then we note that

[1] Actually this step can be divided in two steps. Li and Micciancio[LM16] showed that $\{1, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme implies $\{\mathsf{Unb}, \mathsf{wSel}, \mathsf{WC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme, which in turn implies $\{\mathsf{Unb}, \mathsf{wSel}, \mathsf{FC}\}$-$\mathsf{IND}$-$\mathsf{FE}$ scheme using the generic transformation proposed by Ananth, Jain and Sahai[AJS15].

[2] Note that our transformation also works when the starting $\mathsf{FE}$ scheme supports unbounded key queries.

the transformation of [AS16, ABSV15] also works if we start from semi-adaptively secure FE scheme. Thus we can apply this weaker generic transformation of [AS16, ABSV15] we obtain a FE scheme that is adaptively secure against unbounded collusions.

For efficiency, the resulting adaptively-secure functional encryption could be *width* compact if we combine the work of Hemenway et al. in [HJO$^+$15] and the transformation of Ananth and Sahai in [AS16] along with adaptively secure garbled circuits [HJO$^+$15].

Furthermore, we could also get a Traitor Tracing system which supports apriori-bounded length identities from $\{1, \mathsf{wSel}, \mathsf{WC}\}$-IND-FE scheme as shown in [GS16].

More specifically, we give a figure 1 below which describes the state-of-art relationships between notions of IND-FE scheme while embedding this work together.

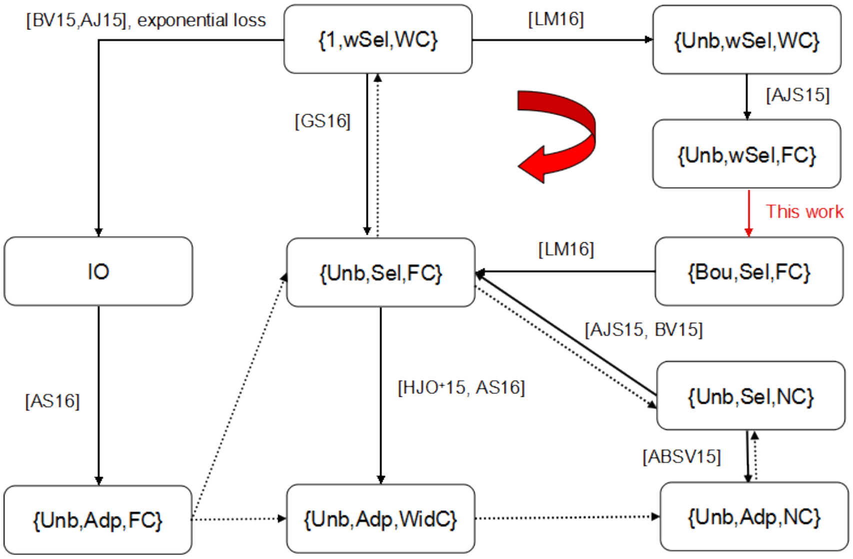

Figure 1: The state-of-art relationships between different notions of IND-FE scheme. This work is given by the red arrow and the alternative approach is given by the bold red arrow. Non-trivial relationships are given by solid arrows and trivial relationships are given by dotted arrows.

## 1.2 Our Technique

We now give an overview of our techniques used in constructing selectively secure FE scheme from weakly selective secure FE scheme. The intuition of our technique is achieved by tunnelling and combining those techniques, which are used in the generic transformation from selective security to adaptive security and the generic transformation from FE scheme for shallow circuits to FE scheme for all circuits in the work of Ananth et al.[ABSV15], based on the structure of the weak selective security game and selective security game of FE scheme. More specifically, our techniques modify the structure of their transformation such that the resulting transformation is amenable to the reduction from selective security to weakly selective security, while preserving the compactness of the input FE scheme.

We now briefly describe the two transformations (in the public key setting) in [ABSV15]. For simplicity, we denote by Trans1 as the generic transformation from selectively secure FE to adaptively secure FE and Trans2 as the generic transformation from FE for shallow circuits to FE for all circuits.

- $\mathsf{Trans1}$: The transformation 1 starts by generating the master key pair $(\mathsf{MPK}_{\mathsf{sel}}, \mathsf{MSK}_{\mathsf{sel}})$ with respect to the underlying selectively secure single-key $\mathsf{FE}$ scheme. To generate functional keys, it takes as input a function $f$ and the master secret key $\mathsf{MSK}_{\mathsf{sel}}$ and outputs a functional key $\mathsf{SK}_G$, where $G$ is a trapdoor circuit [DCIJ+13, GHRW14, BS15] which executes in two threads as follows: the circuit $G$, which is hardwired with the function $f$, a pseudorandom ciphertext $C_E$ and a random tag $\tau$, takes as input a master secret key $\mathsf{MSK}_{\mathsf{adp}}$, a PRF key $K_p$, a symmetric key $K_E$ and a bit $\beta$. If $\beta = 1$, the circuit $G$ outputs a symmetric decryption over the ciphertext $C_E$ using the symmetric key $K_E$, otherwise (i.e., if $\beta = 0$) the circuit $G$ outputs a functional key $\mathsf{SK}_f$ which is generated using the input master secret key $\mathsf{MSK}_{\mathsf{adp}}$. To encrypt the message $m$, $\mathsf{Trans1}$ deploys hybrid functional encryption and dual-system encryption technique. It outputs two ciphertexts $\mathsf{CT}_0 = \mathsf{Enc}_{\mathsf{adp}}(\mathsf{MSK}_{\mathsf{adp}}, m)$ and $\mathsf{CT}_1 = \mathsf{Enc}_{\mathsf{sel}}\left(\mathsf{MPK}_{\mathsf{sel}}, (m, K_p, 0^\lambda, 0)\right)$, where $K_p$ and $\mathsf{MSK}_{\mathsf{adp}}$, which is respect to an adaptively-secure one-ciphertext private-key $\mathsf{FE}$ scheme, are both newly generated in the encryption algorithm. To decrypt, one can use the functional key $\mathsf{SK}_G$ to decrypt the ciphertext $\mathsf{CT}_1$, which corresponds to the external system, to obtain a functional key $\mathsf{SK}_f$ and then use the functional key $\mathsf{SK}_f$ to decrypt the ciphertext $\mathsf{CT}_0$, which corresponds to the internal system, to obtain the result of the function $f$ over the message $m$.

- $\mathsf{Trans2}$: The setup and key generation algorithm of $\mathsf{Trans2}$ are as the same as $\mathsf{Trans1}$, except that the input of the trapdoor circuit is different than that in $\mathsf{Trans1}$. While the circuit $G$ in $\mathsf{Trans1}$ takes as input the tuple $(\mathsf{MSK}_{\mathsf{adp}}, K_p, K_E, \beta)$, the circuit $G$ in $\mathsf{Trans2}$ takes as input the tuple $(m, K_p, K_E, \beta)$ where $m$ is the plaintext, and it outputs a randomized encoding over the function $f$ and the plaintext $m$ if $\beta = 0$. Otherwise it outputs a symmetric decryption of the ciphertext $C_E$ using the symmetric key $K_E$. To decrypt, one can decrypt the ciphertext $\mathsf{CT}$ using the functional key $\mathsf{SK}_G$ to obtain a randomized encoding $\widehat{f}(m)$ and then evaluate it to get the final result $f(m)$.

**Compatibility between Trapdoor Circuits and Security Games**. From $\mathsf{Trans1}$ and $\mathsf{Trans2}$ described above, we noticed the potential of modifying the $\mathsf{Trans1}$ into a transformation from a weakly selective security to selective security. However, we note that the structure of the trapdoor circuit $G[f, C_E, \tau](\mathsf{MSK}_{\mathsf{adp}}, K_p, K_E, \beta)$ used in $\mathsf{Trans1}$ is not compatible with our goal. Interestingly, we found out that the structure of the trapdoor circuit $G[f, C_E, \tau](m, K_p, K_E, \beta)$ used in $\mathsf{Trans2}$ is indeed compatible with the (weak) selective game.

To illustrate this, let us first show the tradeoff between the weakly selective security game and the selective security game, in the message challenge phase, by describing a reduction, which could internally execute some adversary to break the underlying weakly selective secure $\mathsf{FE}$ scheme, while simulating the role of the challenger of the selectively secure $\mathsf{FE}$ scheme. At the very beginning, the adversary first submits a pair of messages $(m_0, m_1)$, and then the reduction which simulates the role of the challenger returns back an functional encryption of $M_b$ where $b$ is a random coin flipped by the challenger of the underlying weakly selectively secure $\mathsf{FE}$ scheme (We use the capital "$M_b$" because $M_b$ is a valid challenge message possibly modified from the challenge message $m_b$ by the reduction while preserving the property that the size of $M_0$ is equivalent to the size of $M_1$). Furthermore, the message challenge phase of selective game is the same as the one of weakly selective game, except that the adversary in the weakly selective game must submit a function query (note that the weakly selectively secure $\mathsf{FE}$ scheme only supports single-key query) *along with* the pair of messages $(M_0, M_1)$ together. We note that this difference does not effect the challenger of the weakly selective game to compute the functional encryption over the message $M_b$. Namely, the reduction can handle the message challenge phase by sending back the challenge ciphertext from the challenger of weakly selective game to the adversary. Nevertheless, the bad news is from the tradeoff between weak selective and selective game in this message challenge phase. In the weakly selective game, the adversary (reduction) needs to submit the function query along with two challenge messages $(M_0, M_1)$ and then the reduction could receive back the functional key and a functional encryption over $M_b$. However, the obstacle is where the function query comes from? Recall that we have shown the reduction could smoothly construct the pair of challenge messages $(M_0, M_1)$ from the pair of messages $(m_0, m_1)$ submitted from the adversary, but the reduction does not receive any function query $f$ which should be an element hardwired in the constructed trapdoor circuit $G[f, C_E, \tau]$ as shown in $\mathsf{Trans1}$ and $\mathsf{Trans2}$. What's worse, the key query phase between the reduction and the adversary will not be open since the reduction does not receive back any functional encryption of $M_b$ because the reduction cannot submit a valid function query. Thus the intractable point to construct the reduction is how to submit the function query while not receiving any function query from the adversary.

**One Functional Key v.s. Two Functional Keys**. To solve the tradeoff we described above, we deploy the *two-key one-ciphertext paradigm* rather than the one-key two-ciphertext paradigm used in $\mathsf{Trans1}$. Recall that the one-key two-ciphertext paradigm used in $\mathsf{Trans1}$ essentially captures the dual-system encryption technique and hybrid functional encryption technique. Namely, it outputs two ciphertexts $\mathsf{CT}_0$ and $\mathsf{CT}_1$ as the final ciphertext $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$. However, the two-key one-ciphertext paradigm we utilize is in order to separate the key generation step such that the reduction could submit the function query to the challenger of weakly selective game without the information of the function query from the adversary, and then the reduction could receive back the challenge ciphertext from the challenger and in turn breaks the tradeoff between the weakly selective game and the selective game. More specifically, our idea is to let the key generation algorithm output two functional keys, $\mathsf{SK}_f$ and $\mathsf{SK}_G$ as the final functional key. The functional key $\mathsf{SK}_f$ is generated by executing the key generation algorithm, $\mathsf{KG}(\mathsf{MSK}_{\mathrm{sSel}}, f)$, of a selectively secure *private-key* $\mathsf{FE}$ scheme, where the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$ is generated by the reduction itself. The functional key $\mathsf{SK}_G$ corresponds to a circuit $G$ which can be constructed by the reduction itself using the master secret key of the underlying weakly selective secure $\mathsf{FE}$ scheme, but we need the circuit $G$ is *independent* of the function $f$. Note that these already allow the reduction to obtain the challenge ciphertext from the challenger and we give more details by describing the interaction between the reduction and the adversary: After the reduction receiving the pair of committed messages $(m_0, m_1)$ from the adversary, the reduction could generate the circuit $G$ by itself without any information of the function $f$, then the reduction could submit the pair of messages $(M_0, M_1)$ and the circuit $G$ , and then receive back a functional encryption over $M_b$ and a functional key $\mathsf{SK}_G$. Now the reduction can open the key query phase by sending the challenge ciphertext $\mathsf{Enc}(M_b)$ to the adversary. In the key query phase, once the adversary submits a function query $f$, the reduction computes $\mathsf{SK}_f$ by itself and sends back the pair of functional keys $(\mathsf{SK}_f, \mathsf{SK}_G)$. Note that for each function query $f$, each pair $(\mathsf{SK}_f, \mathsf{SK}_G)$ is different since each $\mathsf{SK}_f$ is different, even though the reduction sends the same functional key $\mathsf{SK}_G$ along with the functional key $\mathsf{SK}_f$.

**Hybrid Key Generation**. Having decided the strategy to solve the tradeoff between weakly selective game and selective game, the final step is to ensure the structure of the circuit $G$. From the strategy above we have decided some elements of the circuit $G$ since we will use the structure of the circuit $G$ in $\mathsf{Trans2}$ because it is compatible with the selective game (resp. weakly selective game). Firstly let us recall its structure. The circuit $G$ in $\mathsf{Trans2}$, which is hardwired with a function $f$, a random ciphertext $C_E$ and a random tag $\tau$, takes as input the message $m$, a PRF key $K_p$, a symmetric key $K_E$ and a bit $\beta$. When $\beta = 1$ it outputs the symmetric decryption over the ciphertext $C_E$ using the input symmetric key $K_E$, otherwise it outputs a randomized encoding over the function $f$ and the input message $m$. However, our circuit $G$ must be separate from the function $f$ as we described above and hence it is of the form $G[*, C_E, \tau](m, K_p, K_E, \beta)$. Therefore, we need to decide what element(s) (denoted by $*$) are hardwired in the circuit $G$ and how the circuit $G$ executes. Recall the dual-system encryption technique and the hybrid functional encryption used in $\mathsf{Trans1}$, there are two ciphertexts, $\mathsf{CT}_0 = \mathsf{Enc}_{\mathsf{int},k}(m)$ and $\mathsf{CT}_1 = \mathsf{Enc}_{\mathsf{ext}}(k)$, that control the dual-system encryption externally and internally in a hybrid approach. However, in this work, we apply this structure in the key generation algorithm instead of in the encryption algorithm by considering the fact that we need to use two functional keys $\mathsf{SK}_f$ and $\mathsf{SK}_G$ as we illustrated above. Therefore, we propose an innovative method called *hybrid key generation*. Unlike the hybrid encryption method, the purpose of the hybrid key generation method is to essentially combine two key generation algorithms for two circuits $G$ and $f$. More specifically, we use the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$ to generate $\mathsf{SK}_f$ as we mentioned and use the underlying master secret key $\mathsf{MSK}_{\mathrm{wSel}}$ of the underlying weakly selectively secure $\mathsf{FE}$ scheme to generate $\mathsf{SK}_G$, where the master secret key $\mathsf{MSK}_{\mathrm{sSel}}$ is hardwired in the circuit $G$. Finally the hybrid functional key contains two functional keys of the form $(\mathsf{KG}_{\mathsf{ext}}(G_k), \mathsf{KG}_{\mathsf{int},k}(f))$, where $k$ is the hardwired master secret key $\mathsf{MSK}_{\mathrm{sSel}}$. To decrypt, one first decrypts the ciphertext using the external functional key $\mathsf{SK}_G$ to retrieve the internal ciphertext $\mathsf{CT}_{\mathsf{int}}$, which is the output of the circuit $G_k$ and then applies the internal functional key $\mathsf{SK}_f$ to decrypt the internal ciphertext $\mathsf{CT}_{\mathsf{int}}$ to obtain the final result $f(m)$.

**Our Construction in a Nutshell**. Now we have decided the structure of the circuit $G$ and finished introducing techniques to be used in our construction. Now 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 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 $\tau$, takes as input the message $m$, a PRF key $K_p$, a symmetric key $K_E$ and a bit $\beta$ 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 keys $(\mathsf{SK}_f, \mathsf{SK}_G)$ as the functional 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 selectively secure FE scheme. To decrypt, one can decrypt the ciphertext using the functional key $\mathsf{SK}_G$ to release the internal ciphertext and then to decrypt the internal ciphertext using the functional key $\mathsf{SK}_f$.

### 1.3 Comparison with Prior Works

Garg and Srinivasan [GS16] gives a construction from $\{1, \mathsf{wSel}, \mathsf{WC}\}$-IND-FE scheme to a $\{\mathsf{Unb}, \mathsf{Sel}, \mathsf{FC}\}$-IND-FE scheme through simplifying the $i\mathcal{O}$-based selectively-secure FE scheme against unbounded collusions. More specifically, they applied techniques of Garg et al.[GPS15, GPSZ16] and garbled circuits [Yao86]. Even though their construction could start from a single-key weakly selectively secure FE with only weak compactness and has covered our transformation, we believe our transformation is still meaningful. That is, our transformation applies additional building blocks that are the same as Trans1 in the work of Ananth et al.[ABSV15] in an *innovative* way. Surprisingly, our transformation is compact-preserving whereas Trans1 will lost the compactness even though the input scheme is fully compact. This is essentially because our transformation uses two-key one-ciphertext paradigm instead of one-key two-ciphertext paradigm. While still using the Trojan method and dual-system encryption technique, we apply the hybrid key generation technique instead of the hybrid encryption technique due to observation of the tradeoff between the weak selective game and the selective game of FE scheme.

## 2 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 $\mathsf{negl} : \mathbb{N} \rightarrow \mathbb{R}$ is *negligible* if for any polynomial $\mathsf{poly}(\cdot)$ we have $\mathsf{negl}(\lambda) < 1/\mathsf{poly}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

### 2.1 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 $\mathsf{MPK}$ and a master secret key $\mathsf{MSK}$.
- $\mathsf{PKFE.KG}(\mathsf{MSK}, f)$: The key generation algorithm takes as input a secret key $\mathsf{MSK}$ and a function $f \in \mathcal{F}_\lambda$ and outputs a functional key $\mathsf{SK}_f$.
- $\mathsf{PKFE.Enc}(\mathsf{MPK}, m)$: The encryption algorithm takes as input a master public key $\mathsf{MPK}$ and a message $m \in \mathcal{M}_\lambda$, and outputs a ciphertext $\mathsf{CT}$.
- $\mathsf{PKFE.Dec}(\mathsf{SK}_f, \mathsf{CT})$: The decryption algorithm takes as input a functional key $\mathsf{SK}_f$ and a ciphertext $\mathsf{CT}$, and outputs $m \in \mathcal{M}_\lambda \cup \{\perp\}$

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 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 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 2.1** (Weakly Selective Security)**.** A public-key functional encryption scheme $\mathsf{PKFE}$ 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 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 $\mathsf{PKFE}$ 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 key for a single function $f$.

**Definition 2.2** (Selective Security)**.** A public-key functional encryption scheme $\mathsf{PKFE}$ 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

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

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 $\mathsf{MPK}$ 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 $\mathsf{PKFE}$ scheme.

**Definition 2.3** (Fully Compact)**.** A public-key functional encryption scheme $\mathsf{PKFE}$ 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 2.4** (Weakly Compact)**.** A public-key functional encryption scheme $\mathsf{PKFE}$ 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 2.5** (Bounded Collusions)**.** We say a functional encryption is *$q$-bounded* if the adversary is given functional keys for a-priori bounded number of functions $f_1, \cdots, f_q$, which can be made adaptively.

## 2.2 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 2.6** (pseudorandom function)**.** A family $\mathcal{F} = \{\mathsf{PRF}_K : \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)} : K \in \mathcal{K}\}$ of efficiently-computable functions is *pseudorandom* if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that

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

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

## 2.3 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 $\lambda$ 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 $\lambda$:

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

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 $\mathsf{OEnc}$ 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 $\mathsf{SKE.Enc}(K_E, m; r) = (r, \mathsf{PRF}_K(r) \oplus m)$.

# 3 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. [3]
- A symmetric encryption scheme with pseudorandom ciphertext $\mathsf{SKE} = (\mathsf{SKE.Setup}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$.
- A pseudorandom function $\mathsf{PRF}$.

[3] 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.

## 3.1 Construction

We construct the 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 $\lambda$ 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 $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$[4] 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 2 and then generates a functional key $\mathsf{SK}_G \leftarrow \mathsf{wSel.KG}(G, \mathsf{MSK}_{\mathrm{wSel}})$ and a functional 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 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 2: 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 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 3.1.** *Assuming that (1) fully compact, single-key, public-key functional encryption scheme with weakly selective security, (2) selectively 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, bounded-key ($\geq 1$ key queries), public-key functional encryption scheme with selective security.*

*Proof.* *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 $i$th 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 $\mathbf{Hyb}_{i.b}$.

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

$\mathbf{Hyb}_{2.b}$: For every functional query $f$, the challenger replaces $C_E$ with a symmetric encryption $\mathsf{SKE.Enc}(K_E, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}}$ is computed by executing $\mathsf{sSel.Enc}\Big((\mathsf{MSK}^*_{\mathrm{sSel}}, m_b); \mathsf{PRF}_{K^*_p}(\tau)\Big)$ (note that each functional 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 $\tau$ is the

[4] The length of $C_E$ is determined as follows. Denote by $\ell_{\mathsf{sSel}}$ be the length of the ciphertext obtained by encrypting a message of length $|m|$, using $\mathsf{sSel.Enc}$. Further, denote by $\ell_1$ to be the length of ciphertext obtained by encrypting a message of length $\ell_{\mathsf{sSel}}$, using $\mathsf{SKE.Dec}$. We set the length of $C_E$ to be $\ell_1$

random tag associated to the functional key of $f$. The same $K_E^*$ and $K_p^*$ are used while generating all the functional keys, and $K_p^*$ is used in generating the challenge ciphertext $\mathsf{CT}_{\text{pSel}}^* = \mathsf{wSel.Enc}\left(\mathsf{MPK}_{\text{wSel}}^*, (m, K_p^*, 0^\lambda, 0)\right)$. 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 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 3.1.** *Assuming the pseudorandom ciphertexts property of* $\mathsf{SKE}$*, for each* $b \in \{0,1\}$*, we have*

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

*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 $\mathsf{SKE}$. 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 $\mathsf{MSK}_{\text{sSel}}^*$ 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 $(\mathsf{MPK}_{\text{wSel}}, \mathsf{MSK}_{\text{wSel}})$ 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 $\mathsf{CT}_{\text{pSel}}^* \leftarrow \mathsf{wSel.Enc}\left(\mathsf{MPK}_{\text{wSel}}, (m_b, K_p^*, 0^\lambda, 0)\right)$. Now the reduction is ready to handle functional key queries from the adversary. When the adversary submits a functional query $f$, the reduction first picks the tag $\tau$ at random. The reduction obtains $\mathsf{CT}_{\text{sSel}}$ by executing $\mathsf{sSel.Enc}\Big((\mathsf{MSK}_{\text{sSel}}^*, m_b); \mathsf{PRF}_{K_p^*}(\tau)\Big)$. It then sends $\mathsf{CT}_{\text{sSel}}$ 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 $\mathsf{CT}_{\text{sSel}}$. Then the reduction generates a functional key $\mathsf{SK}_G$ by executing $\mathsf{wSel.KG}(G[\mathsf{MSK}_{\text{sSel}}^*, C_E, \tau], \mathsf{MSK}_{\text{wSel}})$ and a functional key $\mathsf{SK}_f'$ by executing $\mathsf{sSel.KG}(\mathsf{MSK}_{\text{sSel}}^*, f)$, then the reduction denotes the tuple $(\mathsf{SK}_f', \mathsf{SK}_G)$ by $\mathsf{SK}_f$ which is sent to the adversary as the functional 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. □

$\mathbf{Hyb}_{3.b}$: 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 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 $\mathbf{Hyb}_{2.b}$.

**Lemma 3.2.** *Assuming the weak selective security of* $\mathsf{wSel}$*, for each* $b \in \{0,1\}$*, we have*

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

*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 $\mathsf{wSel}$. The reduction internally executes the adversary $\mathcal{A}$ by simulating the role of the challenger of the selective $\mathsf{FE}$ 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 $\mathsf{MSK}_{\text{sSel}}^*$ and then sample a random tag $\tau$. Then it generates a symmetric key $K_E^*$ and a PRF key $K_p^*$. The reduction computes $C_E = \mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\text{sSel}})$, where $\mathsf{CT}_{\text{sSel}}$ is the output of $\mathsf{sSel.Enc}\left(\mathsf{MSK}^*_{\mathrm{sSel}}, m_b; \mathsf{PRF}_{K_p^*}(\tau)\right)$, and then it constructs the circuit $G[\mathsf{MSK}^*_{\mathrm{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$. 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 $G[\mathsf{MSK}^*_{\mathrm{sSel}}, C_E, \tau](m, K_p, K_E, \beta)$ to the challenger of the weakly-selectively secure FE scheme (Note that the underlying weakly selectively secure $\mathsf{FE}$ scheme only supports a single-key query). Then the challenger returns back a challenge ciphertext $\mathsf{CT}^*_{\mathrm{wSel}}$ and the functional key $\mathsf{SK}_G$ to the reduction. The reduction denote $\mathsf{CT}^*_{\mathrm{wSel}}$ by $\mathsf{CT}^*_{\mathrm{pSel}}$ as the challenge ciphertext and sends it to the adversary. Now the reduction is ready to handle the functional key queries from the adversary. In the functional key query phase, when the adversary submits a function query $f$, the reduction generates $\mathsf{SK}'_f$ by executing $\mathsf{sSel.KG}(\mathsf{MSK}^*_{\mathrm{sSel}}, f)$ and sends back $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ as the functional 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 $\mathsf{wSel}$, 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 $G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau]$ made by the reduction, we have that $G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau](M_0) = G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau](M_1)$. $G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau](M_0)$ is the functional key which is independent of the function $f$, with respect to the key $\mathsf{MSK}^*_{\mathrm{sSel}}$ and randomness $\mathsf{PRF}_{K_p^*}(\tau)$. Furthermore, $G[\mathsf{MSK}_{\mathrm{sSel}}, C_E, \tau](M_1)$ is the decryption of $C_E$ which is nothing but the encryption of the input message $m_b$ with respect to key $\mathsf{MSK}^*_{\mathrm{sSel}}$ 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 $\mathsf{wSel}$. This completes the proof of the lemma. □

$\mathbf{Hyb}_{4.b}$: For every function query $f$ made by the adversary, the challenger generates $C_E$ in all the functional keys with $\mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}}$ is the output of $\mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\mathrm{sSel}}, m_b); R\right)$, 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 $\mathbf{Hyb}_{3.b}$.

**Lemma 3.3.** *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 $\mathsf{FE}$ 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 $\tau$ 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 $\mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}} = \mathsf{sSel.Enc}(\mathsf{MSK}^*_{\mathrm{sSel}}, m_b; R^*)$. The reduction then proceeds as in the previous hybrids to compute the functional key $\mathsf{SK}_f$ 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. □

Now we 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.

**Lemma 3.4.** *Assuming the selective security of the scheme* $\mathsf{sSel}$*, 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 $\mathsf{sSel}$. 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 $\mathsf{CT}_{\text{sSel}}$ and then the reduction computes $C_E$ as $C_E = \mathsf{SKE.Enc}(K^*_E, \mathsf{CT}_{\text{sSel}})$ where $K^*_E$ is the output of $\mathsf{SKE.Setup}(1^\lambda)$. The reduction first generates $\mathsf{MPK}_{\text{wSel}}$ and the symmetric key $K^*_E$ which is the output of $\mathsf{SKE.Setup}(1^\lambda)$, and then it sends back the challenge ciphertext $\mathsf{CT}^*_{\text{pSel}} = \mathsf{wSel.Enc}(\mathsf{MPK}_{\text{wSel}}, (m_b, 0, K^*_E, 1))$. (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 $\mathsf{MSK}_{\text{sSel}}$ 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 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 key $\mathsf{SK}'_f$. Now the reduction generates the functional key $\mathsf{SK}_G$ by it self and sends back $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ to the adversary as the functional 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 $\mathsf{sSel}$, 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 $\mathsf{pSel}$ is the same as each functional query made by the reduction and the adversary of $\mathsf{pSel}$ 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 $\mathsf{sSel}$. 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 selectively secure public-key FE scheme $\mathsf{wSel}$. Therefore, if the scheme $\mathsf{wSel}$ is compact, then the resulting scheme $\mathsf{pSel}$ 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

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

which proves that our transformation is compact-preserving. □

# References

[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.

[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.

[AS16] Prabhanjan Ananth and Amit Sahai. Functional encryption for turing machines. In *Theory of Cryptography Conference*, pages 125–153. Springer, 2016.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Advances in CryptologyCRYPTO 2001*, pages 213–229. Springer, 2001.

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

[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.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*, pages 253–273. Springer, 2011.

[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.

[DCIJ$^+$13] Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam ONeill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In *Advances in Cryptology–CRYPTO 2013*, pages 519–535. Springer, 2013.

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

[GHRW14] Craig Gentry, Shai Halevi, Mariana Raykova, and Daniel Wichs. Outsourcing private ram computation. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 404–413. IEEE, 2014.

[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.

[GPS15] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. Cryptology ePrint Archive, Report 2015/1078, 2015. http://eprint.iacr.org/2015/1078.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm, 2006.

[GPSZ16] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the subexponential barrier in obfustopia. Cryptology ePrint Archive, Report 2016/102, 2016. http://eprint.iacr.org/2016/102.

[GS16] Sanjam Garg and Akshayaram Srinivasan. Unifying security notions of functional encryption. Cryptology ePrint Archive, Report 2016/524, 2016. http://eprint.iacr.org/2016/524.

[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.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from lwe. In *Annual Cryptology Conference*, pages 503–523. Springer, 2015.

[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.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 146–162. Springer, 2008.

[LM16] 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.

[LOS$^+$10] Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 62–91. Springer, 2010.

[Sha84] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 47–53. Springer, 1984.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *International Workshop on Public Key Cryptography*, pages 53–70. Springer, 2011.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 162–167. IEEE, 1986.

# Appendices

## Appendix A Preliminaries (Cont.)

### A.1 Private-Key Functional Encryption

A private-key functional encryption scheme $\mathsf{SKFE}$ 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 $\mathsf{MSK}$.

- $\mathsf{SKFE.KG}(\mathsf{MSK}, f)$: The key generation algorithm takes as input a secret key $\mathsf{MSK}$ and a function $f \in \mathcal{F}_\lambda$ and outputs a functional key $\mathsf{SK}_f$.
- $\mathsf{SKFE.Enc}(\mathsf{MSK}, m)$: The encryption algorithm takes as input a master secret key $\mathsf{MSK}$ and a message $m \in \mathcal{M}_\lambda$, and outputs a ciphertext $\mathsf{CT}$.
- $\mathsf{SKFE.Dec}(\mathsf{SK}_f, \mathsf{CT})$: The decryption algorithm takes as input a functional key $\mathsf{SK}_f$ and a ciphertext $\mathsf{CT}$, 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 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 $\mathsf{SKFE}$ 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 $\mathsf{CT}$ 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 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 $\mathsf{SKFE}$ 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 key for a single function $f$.

**Definition A.2** (Selective Security)**.** A private-key functional encryption scheme $\mathsf{SKFE}$ 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 $1^\lambda$ 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 $\mathsf{SKFE}$ scheme.

**Definition A.3** (Fully Compact)**.** A private-key functional encryption scheme $\mathsf{SKFE}$ 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 $\mathsf{SKFE}$ 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 3 is that there is only one master key $\mathsf{MSK}_{\text{wSel}}$ 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 3, 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 $\lambda$ in unary, it executes the algorithm $\mathsf{wSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\text{wSel}}$. The algorithm outputs the master secret key $\mathsf{MSK}_{\text{pSel}} = \mathsf{MSK}_{\text{wSel}}$.

**Key Generation** $\mathsf{pSel.KG}(\mathsf{MSK}_{\text{pSel}}, f)$: Takes as input a master secret key $\mathsf{MSK}_{\text{pSel}}$ and a function $f$, it first executes $\mathsf{sSel.Setup}(1^\lambda)$ to obtain the master secret key $\mathsf{MSK}_{\text{sSel}}$. 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 $G = G[\mathsf{MSK}_{\text{sSel}}, C_E, \tau]$ as described in the figure 2 and then generates a functional key $\mathsf{SK}_G \leftarrow \mathsf{wSel.KG}(G, \mathsf{MSK}_{\text{wSel}})$ and a functional key $\mathsf{SK}'_f \leftarrow \mathsf{sSel.KG}(\mathsf{MSK}_{\text{sSel}}, f)$. Finally it outputs $\mathsf{SK}_f = (\mathsf{SK}'_f, \mathsf{SK}_G)$ as the functional key.

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

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

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

**Decryption** $\mathsf{pSel.Dec}(\mathsf{SK}_f, \mathsf{CT}_{\mathrm{pSel}})$: On input a functional 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 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($\geq 1$ 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, $\mathsf{CT}_{\mathrm{pSel}}$ is obtained by executing $\mathsf{wSel.Enc}\left(\mathsf{MSK}_{\mathrm{wSel}}, (m_b, K_p, 0^\lambda, 0)\right)$.

$\mathbf{Hyb}_{2.b}$: For every functional query $f$, the challenger replaces $C_E$ with a symmetric encryption $\mathsf{SKE.Enc}(K_E, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}} \leftarrow \mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\mathrm{sSel}}, m_b); \mathsf{PRF}_{K_p^*}(\tau)\right)$ (note that each functional 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 $\tau$ is the random tag associated to the functional key of $f$. The same $K_E^*$ and $K_p^*$ are used while generating all the functional keys, and $K_p^*$ is used generating the challenge ciphertext $\mathsf{CT}^*_{\mathrm{pSel}} = \mathsf{wSel.Enc}\left(\mathsf{MSK}^*_{\mathrm{wSel}}, (m, K_p^*, 0^\lambda, 0)\right)$. 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 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 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 $\mathbf{Hyb}_{2.b}$.

$\mathbf{Hyb}_{4.b}$: For every function query $f$ made by the adversary, the challenger generates $C_E$ in all the functional keys with $\mathsf{SKE.Enc}(K_E^*, \mathsf{CT}_{\mathrm{sSel}})$, where $\mathsf{CT}_{\mathrm{sSel}}$ is the output of $\mathsf{sSel.Enc}\left((\mathsf{MSK}^*_{\mathrm{sSel}}, x_b); R\right)$, 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 $\mathsf{wSel}$. Therefore, if the scheme $\mathsf{wSel}$ is compact, then the resulting scheme $\mathsf{pSel}$ 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. □