From Selective to Adaptive Security in Functional Encryption

Prabhanjan Ananth* UCLA
Zvika Brakerski† Weizmann
Gil Segev‡ Hebrew U
Vinod Vaikuntanathan§ MIT

Abstract

In a functional encryption (FE) scheme, the owner of the secret key can generate restricted decryption keys that allow users to learn specific functions of the encrypted messages and nothing else. In many known constructions of FE schemes, security is guaranteed only for messages that are fixed ahead of time (i.e., before the adversary even interacts with the system). This so-called *selective security* is too restrictive for many realistic applications. Achieving *adaptive security* (also called *full security*), where security is guaranteed even for messages that are adaptively chosen at any point in time, seems significantly more challenging. The handful of known adaptively-secure schemes are based on specifically tailored techniques that rely on strong assumptions (such as obfuscation or multilinear maps assumptions).

We show that any sufficiently-expressive *selectively-secure* FE scheme can be transformed into an *adaptively-secure* one without introducing any additional assumptions. We present a black-box transformation, for both public-key and private-key schemes, making novel use of *hybrid encryption*, a classical technique that was originally introduced for improving the efficiency of encryption schemes. We adapt the hybrid encryption approach to the setting of functional encryption via a technique for embedding a "hidden execution thread" in the decryption keys of the underlying scheme, which will only be activated within the proof of security of the resulting scheme. As an additional application of this technique, we show how to construct functional encryption schemes for arbitrary circuits starting from ones for shallow circuits (NC1 or even TC0).

Keywords: Functional encryption, adaptive security, generic constructions.

*E-mail: `prabhanjan@cs.ucla.edu`. This work was done in part while visiting MIT, and was supported in part by the Northrop Grumman Cybersecurity Consortium. Research supported in part from a DARPA/ONR PROCEED award, NSF Frontier Award 1413955, NSF grants 1228984, 1136174, 1118096, and 1065276. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11- 1-0389. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

†E-mail: `zvika.brakerski@weizmann.ac.il`. Supported by an Alon Fellowship and by ISF.

‡E-mail: `segev@cs.huji.ac.il`. Supported by the European Union's Seventh Framework Programme (FP7) via a Marie Curie Career Integration Grant, by the Israel Science Foundation (Grant No. 483/13), and by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11).

§Email: `vinodv@mit.edu`. Research supported in part by DARPA Grant number FA8750-11-2-0225, an Alfred P. Sloan Research Fellowship, the Northrop Grumman Cybersecurity Research Consortium (CRC), Microsoft Faculty Fellowship, and a Steven and Renee Finn Career Development Chair from MIT.

1 Introduction

Traditional notions of public-key encryption provide all-or-nothing access to data: owners of the secret key can recover the entire message from a ciphertext, whereas those who do not know the secret key learn nothing at all. Functional encryption, a revolutionary notion originating from the work of Sahai and Waters [SW05], is a modern type of encryption scheme where the owner of the (master) secret key can release function-specific secret keys sk_f, referred to as *functional keys*, which enable a user holding an encryption of a message x to compute $f(x)$ but nothing else (see [KSW08,LOS+10,BSW11,O'N10] and many others). Intuitively, in terms of indistinguishability-based security, encryptions of any two messages, x_0 and x_1, should be computationally indistinguishable given access to functional keys for any function f such that $f(x_0) = f(x_1)$.

While initial constructions of functional encryption schemes [BF03,BCO+04,KSW08,LOS+10] were limited to restricted function classes such as point functions and inner products, recent developments have dramatically improved the state of the art. In particular, the works of Sahai and Seyalioglu [SS10] and Gorbunov, Vaikuntanathan and Wee [GVW12] showed that a scheme supporting a single functional key can be based on any semantically-secure encryption scheme. This result can be extended to the case where the number of functional keys is polynomial and known a-priori [GVW12]. Goldwasser, Kalai, Popa, Vaikuntanathan and Zeldovich [GKP+13] constructed a scheme with succinct ciphertexts based on a specific hardness assumption (Learning with Errors).

The first functional encryption scheme that supports a-priori unbounded number of functional keys was constructed by Garg, Gentry, Halevi, Raykova, Sahai and Waters [GGH+13], based on the existence of a general-purpose indistinguishability obfuscator (for which a heuristic construction is presented in the same paper). Garg et al. showed that given any such obfuscator, their functional encryption scheme is *selectively secure*. At a high level, selective security guarantees security only for messages that are fixed ahead of time (i.e., before the adversary even interacts with the system). Whereas security only for such messages may be justified in some cases, it is typically too restrictive for realistic applications. A more realistic notion is that of *adaptive security* (often called *full security*), which guarantees security even for messages that can be adaptively chosen at any point in time.

Historically, the first functional encryption schemes were only proven selectively secure [BB04,GPS+06, KSW08,GVW13,GKP+13]. The problem of constructing adaptively secure schemes seems significantly more challenging and only few approaches are known. A simple observation is that if a selectively-secure scheme's message space is not too large, e.g., $\{0,1\}^n$ for a relatively small n, then any adaptively-chosen message x can be guessed ahead of time with probability 2^{-n}. Starting with a *sub-exponential* hardness assumption, and taking the security parameter to be polynomial in n allows us to argue that the selectively-secure scheme is in fact also adaptively secure. This observation is known as "complexity leveraging" and is clearly not satisfactory in general.

The powerful "dual system" approach, put forward by Waters [Wat09], has been used to construct adaptively-secure attribute-based encryption scheme (a restricted notion of functional encryption) for formulas, as well as an adaptively-secure functional encryption scheme for linear functions [LOS+10]. However, this method is a general outline, and each construction was so far required to tailor the solution based on its specialized assumption. In some cases, such as attribute-based encryption for circuits, it is still not known how to implement dual system encryption to achieve adaptive security (although Garg, Gentry, Halevi and Zhandry [GGH+14a] show how to do this with custom-built methods and hardness assumptions).

Starting with [GGH+13], there has been significant effort in the research community to construct an adaptively-secure general-purpose functional encryption scheme with an unbounded number of functional keys. Boyle, Chung and Pass [BCP14] constructed an adaptively secure scheme, under the assumption that differing-input obfuscators exist (these are stronger primitives than the indistinguishability obfuscators used by [GGH+13]). Following their work, Waters [Wat14] and Garg, Gentry, Halevi and Zhandry [GGH+14b] constructed specific adaptively-secure schemes assuming indistinguishability obfuscation and assuming non-standard assumptions on multilinear maps, respectively. Despite this significant progress, each of these constructions relies on somewhat tailored methods and techniques.

1.1 Our Results: From Selective to Adaptive Security

We show that any selectively-secure functional encryption scheme implies an adaptively-secure one, without relying on any additional assumptions. Our transformation applies equally to public-key schemes and to private-key ones, where the resulting adaptive scheme inherits the public-key or private-key flavor of the underlying scheme. The following theorem informally summarizes our main contribution.

Theorem 1 (informal)**.** *Given any public-key (resp. private-key) selectively-secure functional encryption scheme for the class of all polynomial size circuits, there exists an adaptively-secure public-key (resp. private-key) functional encryption scheme with similar properties.*

Specifically, the adaptive scheme supports slightly smaller circuits than those supported by the selective scheme we started with.

Our transformation can be applied, in particular, to the selectively-secure schemes of Garg et al. [GGH$^+$13] and Waters [Wat14], resulting in adaptively-secure schemes based on indistinguishability obfuscation (and one-way functions).[1]

We view the significance of our result in a number of dimensions. First of all, it answers the basic call of cryptographic research to substantiate the existence of rather complex primitives on that of somewhat simpler ones. We feel that this is of special interest in the case of adaptive security, where it seemed that ad-hoc methods were required. Secondly, our construction, being of fairly low overhead, will allow to focus the attention of the research community in studying selectively-secure functional encryption schemes, rather than investing unwarranted efforts in obtaining adaptively-secure ones. Lastly, we hope that our methods will be extended towards weaker forms of functional encryption schemes for which adaptive security is yet unattained generically, such as attribute-based encryption for all polynomial-size circuits.

1.2 Our Techniques

Our result is achieved by incorporating a number of techniques which will be explained in this section. In a nutshell, our main observation is that *hybrid encryption* (a.k.a key encapsulation) can be employed in the context of functional encryption, and has great potential in going from selective to adaptive security of encryption schemes. At a first glance, *hybrid functional encryption* should lead to a selective-to-adaptive transformation, given an additional weak component: A *symmetric* FE which is adaptively secure when only a single message query is allowed. We show that the latter can be constructed from any one-way function as a corollary of [GVW12, BS15]. However, the intuitive reasoning fails to translate into a proof of security. To resolve this issue, we use a technique we call *The Trojan Method*, which originates from De Caro et al.'s "trapdoor circuits" [CIJ$^+$13] (similar ideas had been since used by Gentry et al. [GHR$^+$14] and Brakerski and Segev [BS15]).

We conclude this section with a short comparison of our technique with the aforementioned "dual system encryption" technique that had been used to achieve adaptively secure attribute based encryption.

Hybrid Functional Encryption. Hybrid encryption is a veteran technique in cryptography and has been used in a variety of settings. We show that in the context of functional encryption it is especially powerful.

The idea in hybrid encryption is to combine two encryption schemes: An "external" scheme (sometimes called KEM – Key Encapsulation Mechanism) and an "internal" scheme (sometimes called DEM – Data Encapsulation Mechanism). In order to encrypt a message in the hybrid scheme, a fresh key is generated for the internal scheme, and is used to encrypt the message. Then the key itself is encrypted using the external scheme. The final hybrid ciphertext contains the two ciphertexts: $(\mathsf{Enc}_{\mathsf{ext}}(k), \mathsf{Enc}_{\mathsf{int},k}(m))$ (all external ciphertexts use the same key). To decrypt, one first decrypts the external ciphertext, retrieves k and applies it to the internal ciphertext. Note that if, for example, the external scheme is public-key and the internal is symmetric key, then the resulting scheme will also be public key. Hybrid encryption is often used in cases where the external scheme is less efficient (e.g. in encrypting long messages) and thus there is an advantage

[1] Waters [Wat14] also constructed an adaptively-secure scheme, but using specific ad-hoc techniques and in a significantly more complicated manner.

in using it to encrypt only a short key, and encrypt the long message using the more efficient internal scheme. Lastly, note that the internal scheme only needs to be able to securely encrypt a single message.

The intuition as to why hybrid encryption may be good for achieving adaptive security is that the external scheme only encrypts keys for the internal scheme. Namely, it only encrypts messages from a predetermined and known distribution, so selective security should be enough for the external scheme. The hardness of adaptive security is "pushed" to the internal scheme, but there the task is easier since the internal scheme only needs to be able to encrypt a single message, and it can be private-key rather than public-key.

Let us see how to employ this idea in the case where both the internal and external schemes are FE schemes. To encrypt, we will generate a fresh master secret key for the internal scheme, and encrypt it under the external scheme. To generate a key for the function f, the idea is to generate a key for the function $G_f(\mathsf{msk_{int}})$ which takes a master key for the internal scheme, and outputs a secret key for function f under the internal scheme, using $\mathsf{msk_{int}}$ (randomness is handled using a PRF). This will allow to decrypt in a two-step process as above. First apply the external secret-key for G_f to the external ciphertext, this will give you an internal secret key for f, which is in turn applied to the internal ciphertext to produce $f(x)$.

For the external scheme, we will use a selectively secure FE scheme (for the sake of concreteness, let us say public-key FE). As explained above, selective security is sufficient here since all the messages encrypted using the external scheme can be generated ahead of time (i.e. they do not depend on the actual x's that the user wishes to encrypt).

For the internal scheme, we require an FE scheme that is *adaptively secure*, but only supports the encryption of a single message. Fortunately, such a primitive can be derived from the works of [GVW12, BS15]. In [GVW12], the authors present an adaptively 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. This construction is based on the existence of semantically secure encryption, so the public-key version needs public-key encryption and the symmetric version needs symmetric encryption. While this primitive seems dual to what we need for our purposes, [BS15] shows how to transform private-key FE schemes into *function private* FE. In function-private FE, messages and functions enjoy the same level of privacy, in the sense that a user that produces x_0, x_1, f_0, f_1 such that $f_0(x_0) = f_1(x_1)$ cannot distinguish between $(\mathsf{Enc}(x_0), \mathsf{sk}_{f_0})$ and $(\mathsf{Enc}(x_1), \mathsf{sk}_{f_1})$. Therefore, after applying the [BS15] transformation, we can switch the roles of the functions and messages, and obtain a symmetric FE scheme which is adaptively secure for a *single message and many functions*. (We note that the symmetric version of the [GVW12] scheme can be shown to be function private even without the [BS15] transformation, however since this claim is not made explicitly in the paper we choose not to rely on it.)

Whereas intuitively this should solve the problem, it is not clear how to prove security of the new construction. Standard security proofs for hybrid encryption follow by first relying on the security of the external scheme and removing the encapsulated key, and then relying on the security of the internal scheme and removing the message. However, in our case, removing the encapsulated key is easily distinguishable, since the adversary is allowed to obtain functional keys and apply them to the ciphertext (so long as $f(x_0) = f(x_1)$). Without the internal key, the decryption process no longer works. To resolve this difficulty, we use the Trojan method.

Before we describe the Trojan method, we pause to note that our idea so far can be thought of as "boosting" a single-message, many-key, adaptive symmetric-key FE into a many-message, many-key, adaptive public-key FE (using a selective public-key FE as a "catalyst"). The recent work of Waters [Wat14] proceeds along a similar train of thought, and indeed, motivated our approach. However, while our transformation is simple and general, Waters has to rely on a powerful catalyst, namely an indistinguishability obfuscator.

The Trojan Method. The Trojan Method, which is a generalization of techniques used in [CIJ+13] and later in [GHR+14, BS15], is a way to embed a hidden functionality thread in an FE secret-key that can only be invoked by special ciphertexts generated using special (secret) back-door information. This thread remains completely unused in the normal operation of the scheme (and can be instantiated with meaningless functionality). In the proof, however, the secret thread will be activated by the challenge ciphertext in such a way that is indistinguishable to the user (= attacker). Namely, the user will not be able to tell that it is executing the secret thread and not the main thread. This will be extremely beneficial to prove security. We wish to argue that in the view of the user, the execution of the main thread does not allow to distinguish between the encryption of two messages x_0, x_1. The problem is that for functionality purposes, the main thread has to know which input it is working on. This is where the hidden thread comes into the play. We will design the hidden thread so that in the eyes of the user, it is computationally indistinguishable from the main thread on the special messages x_0, x_1. However, in the hidden thread, the output can be computed in a way that does not distinguish between x_0 and x_1 (either by a statistical or a computational argument), which will allow us to conclude that encryptions of x_0, x_1 are indistinguishable.

In particular, this method will resolve the aforementioned conundrum in our proof outline above. In the proof, we will use the Trojan method to embed a hidden thread in which $\mathsf{msk}_{\mathsf{int}}$ is not used at all, but rather G_f produces a precomputed internal sk_f. This will allow us to remove $\mathsf{msk}_{\mathsf{int}}$ from the challenge ciphertext and use the security properties of the internal scheme to argue that a internal encryption of x_0, x_1 are identical so long as $f(x_0) = f(x_1)$.

We note that an important special case of the above outline is when the trojan thread is a constant function. This had been the case in [CIJ$^+$13, GHR$^+$14], and this is the case in this work as well. However, we emphasize that our description here allows for greater generality since we allow the trojan thread to implement functionality that depends on the input x. We feel that this additional power may be useful for future applications.

Technically, the hidden thread is implemented using (standard) symmetric-key encryption, which in turn can be constructed starting with any one-way function. In the functional secret-key generation process for a function f, the secret-key generation process will produce a symmetric-key ciphertext c (which can just be encryption of 0 or another fixed message, since it only needs to have meaningful content in the security proof). It will then consider the function $G_{f,c}$ that takes as input a pair (x, s), and first checks whether it can decrypt c using s as a symmetric key. If it cannot, then it just runs f on x and returns the output. If s actually decrypts c, we consider $f^* = \mathsf{Dec}_s(c)$ (i.e. c encrypts a description of a function), and the output is the execution of $f^*(x)$. The value c is therefore used as a Trojan Horse: Its contents are hidden from the users of the scheme, however given a hidden command (in the form of the symmetric s) it can embed functionality that "takes over" the functional secret-key.

We note that in order to support the Trojan method, the decryption keys of our FE scheme need to perform symmetric decryption, branch operations, and execution of the function f^*. Thus we need to start with an FE scheme which allows for the generation of sufficiently expressive keys.

Our Trojan method can be seen as a weak form of function privacy in FE, but one that can be applied even in the context of public-key FE. In essence, we cannot hide the main thread of the evaluated function (this is unavoidable in public-key FE). However, we can hide the secret thread and thus allow the function to operate in a designated way for specially generated ciphertexts. (This interpretation is not valid for previous variants of this method such as "trapdoor circuits" [CIJ$^+$13].)

A simple application of the Trojan method is our reduction in Section 4, showing that FE that only supports secret-keys for functions with shallow circuits (e.g. logarithmic depth) implies a scheme that works for circuits of arbitrary depth (although with a size bound). Essentially, instead of producing a secret key for the desired functionality, we output a key for the function that computes a *randomized encoding* of that functionality. A *(computational) randomized encoding* [IK00, AIK05] of an input-function pair $\mathsf{RE}(f, x)$ is, in a nutshell, a representation of $f(x)$ that reveals no information except $f(x)$ on one hand, but can be computed with less resources on the other (in our case, lower depth). To make the proof work, the Trojan thread will contain a precomputed $\mathsf{RE}(f, x_0)$ value, which will allow us to use the security property of the encoding scheme and switch it to $\mathsf{RE}(f, x_1)$. See Section 4 for details. We note that a similar approach is used in [GHR$^+$14, Appendix D] to achieve FE that works for RAM machines.

Relation to Dual-System Encryption. Our approach takes some resemblance to the "Dual-System Encryption" method of Waters [Wat09] and followup works [LW10, LW12]. This method had been used to prove adaptive security for Identity Based Encryption and Attribute Based Encryption, based on the hardness of some problems on groups with bilinear-maps. In broad terms, in their proof the distribution of the ciphertext is changed into "semi-functional" mode in a way that is indiscoverable by an observer. A semi-functional ciphertext is still decryptable by normal secret keys. Then, the secret-keys are modified into semi-functional form, which is useless in decrypting semi-functional ciphertexts. This is useful since in IBE and ABE, the challenge ciphertext is not supposed to be decryptable by those keys given to the adversary. Still, a host of algebraic techniques are used to justify the adversary's inability to produce other semi-functional ciphertexts in addition to the challenge, which would foil the reduction.

Our proof technique also requires changing the distributions of the keys and challenge ciphertext. However, there are also major differences. Our modified ciphertext is not allowed to interact with properly generated secret keys, and therefore the distinction between "normal" and "semi-functional" does not fit here. Furthermore, in Identity Based and Attribute Based Encryption, the attacker in the security game is not allowed to receive keys that reveal any information on the message, which allows to generate semi-functional ciphertexts that do not contain any information, whereas in our case, there is a structured and well-defined output for any ciphertext and any key. This means that the information required for decryption (which can be a-priori unbounded) needs to be embedded in the keys. Lastly, our proof is completely generic and does not rely on the algebraic structure of the underlying hardness assumption as in previous implementations of this method.

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. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For a randomized function f and an input $x \in \mathcal{X}$, we denote by $y \leftarrow f(x)$ the process of sampling a value y from the distribution $f(x)$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for any polynomial $p(\lambda)$ it holds that $\mathsf{negl}(\lambda) < 1/p(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

2.1 Pseudorandom Functions and Symmetric Encryption

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

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

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

We say that a pseudorandom function family $\mathcal{F}$ is implementable in NC^1 if every function in $\mathcal{F}$ can be implemented by a circuit of depth $c \cdot \log(n)$, for some constant c. We also consider the notion of a *weak* pseudorandom function family, asking that the above definition holds for adversaries that may access the functions on random inputs (that is, the oracles $\mathsf{PRF}_\mathsf{K}(\cdot)$ and $\mathsf{R}(\cdot)$ take no input, and on each query they sample a uniform input r and output $\mathsf{PRF}_\mathsf{K}(r)$ and $\mathsf{R}(r)$, respectively).

Symmetric encryption with pseudorandom ciphertexts. A symmetric encryption scheme consists of a tuple of PPT algorithms ($\mathsf{Sym.Setup}, \mathsf{Sym.Enc}, \mathsf{Sym.Dec}$). The algorithm $\mathsf{Sym.Setup}$ takes as input a security parameter λ in unary and outputs a key K_E. The encryption algorithm $\mathsf{Sym.Enc}$ takes as input a symmetric key K_E and a message m and outputs a ciphertext CT. The decryption algorithm $\mathsf{Sym.Dec}$ takes as input a symmetric key K_E and a ciphertext CT and outputs the message m.

In this work, we require a symmetric encryption scheme Π where the ciphertexts produced by $\mathsf{Sym.Enc}$ 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 $\mathsf{Sym.Enc}(\mathsf{Sym.K}, m; r)$. Let $R_{\ell(\lambda)}(\cdot)$ 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:

$$\mathsf{Adv}^{\mathsf{symPR}}_{\Pi,\mathcal{A}}(\lambda) = \left|\mathsf{Pr}\left[\mathcal{A}^{\mathsf{OEnc}_{\mathsf{Sym.K}}(\cdot)}(1^\lambda) = 1\ \right] - \mathsf{Pr}\left[\mathcal{A}^{\mathsf{R}_{\ell(\lambda)}(\cdot)}(1^\lambda) = 1\ \right]\right|$$

where the probability is taken over the choice of $\mathsf{Sym.K} \leftarrow \mathsf{Sym.Setup}(1^\lambda)$, and over the internal randomness of $\mathcal{A}$, 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 $\mathsf{Sym.Enc}(\mathsf{K}, m; r) = (r, \mathsf{PRF}_{\mathsf{K}}(r) \oplus m)$ (see [Gol04] for more details).

2.2 Public-Key Functional Encryption

A public-key functional encryption (FE) scheme Π_{Pub} 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 $(\mathsf{Pub.Setup}, \mathsf{Pub.KeyGen}, \mathsf{Pub.Enc}, \mathsf{Pub.Dec})$ of PPT algorithms with the following properties:

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

We say that such a scheme is defined for a complexity class $\mathcal{C}$ if it supports all the functions that can be implemented in $\mathcal{C}$. In terms of 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 $f \in \mathcal{F}_\lambda$ it holds that $\Pr\left[\mathsf{Pub.Dec}(\mathsf{Pub.KeyGen}(\mathsf{MSK}, f), \mathsf{Pub.Enc}(\mathsf{MPK}, m)) = f(m)\right] \geq 1 - \mathsf{negl}(\lambda)$, where $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Pub.Setup}(1^\lambda)$, and the probability is taken over the random choices of all algorithms.

We consider the standard selective and adaptive indistinguishability-based notions for functional encryption (see, for example, [BSW11, O'N10]). 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 public key and functional keys.[2]

Definition 2 (Selective security)**.** *A public-key functional encryption scheme* $\Pi = (\mathsf{Sel.Setup}, \mathsf{Sel.KeyGen}, \mathsf{Sel.Enc}, \mathsf{Sel.Dec})$ *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*

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

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

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

[2] 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.

2. **Challenge phase:** *On input* 1^λ *the adversary submits* (m_0, m_1)*, and the challenger replies with* $\mathsf{Sel.MPK}$ *and* $\mathsf{CT} \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MPK}, m_b)$*.*

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* $\mathsf{Sel}.sk_f \leftarrow \mathsf{Sel.KeyGen}(\mathsf{Sel.MSK}, f)$*.*

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

Definition 3 (Adaptive security)**.** *A public-key functional encryption scheme* $\Pi = (\mathsf{Ad.Setup}, \mathsf{Ad.KeyGen}, \mathsf{Ad.Enc}, \mathsf{Ad.Dec})$ *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* adaptively secure *if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr[\mathsf{Expt}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1] \right| \leq \mathsf{negl}(\lambda)$$

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

1. **Setup phase:** *The challenger samples* $(\mathsf{Ad.MPK}, \mathsf{Ad.MSK}) \leftarrow \mathsf{Ad.Setup}(1^\lambda)$*, and sends* $\mathsf{Ad.MPK}$ *to the adversary.*

2. **Query phase I:** *The adversary adaptively queries the challenger with any function* $f \in \mathcal{F}_\lambda$*. For each such query, the challenger replies with* $\mathsf{Sel}.sk_f \leftarrow \mathsf{Ad.KeyGen}(\mathsf{Ad.MSK}, f)$*.*

3. **Challenge Phase***: The adversary submits* (m_0, m_1) *such that* $f(m_0) = f(m_1)$ *for all function queries* f *made so far, and the challenger replies with* $\mathsf{CT} \leftarrow \mathsf{Ad.Enc}(\mathsf{Ad.MSK}, m_b)$*.*

4. **Query phase II:** *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* $\mathsf{Sel}.sk_f \leftarrow \mathsf{Ad.KeyGen}(\mathsf{Ad.MSK}, f)$*.*

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

3 Our Transformation in the Public-Key Setting

In this section we present our transformation from selective security to adaptive security for public-key functional encryption schemes. In addition to any selectively-secure public-key functional encryption scheme (see Definition 2), our transformation requires a *private-key* functional encryption scheme that is adaptively-secure for a single message query and many function queries. Based on [GVW12, BS15], such a scheme can be based on any one-way function[3].

More specifically, we rely on the following building blocks (all of which are implied by any selectively-secure public-key functional encryption scheme):

1. A selectively-secure public-key functional encryption scheme $\mathsf{Sel} = (\mathsf{Sel.Setup}, \mathsf{Sel.KeyGen}, \mathsf{Sel.Enc}, \mathsf{Sel.Dec})$.

2. An adaptively-secure single-ciphertext private-key functional encryption scheme[4] $\mathsf{OneCT} = (\mathsf{OneCT.Setup}, \mathsf{OneCT.KeyGen}, \mathsf{OneCT.Enc}, \mathsf{OneCT.Dec})$.

3. A symmetric encryption scheme with pseudorandom ciphertexts $\mathsf{SYM} = (\mathsf{Sym.Setup}, \mathsf{Sym.Enc}, \mathsf{Sym.Dec})$.

[3]Gorbunov et al. [GVW12] constructed a private-key functional encryption scheme that is adaptively secure for a single function query and many message queries based on any private-key encryption scheme (and thus based on any one-way function). Any such scheme can be turned into a function private one using the generic transformation of Brakerski and Segev [BS15], and then one can simply switch the roles of functions and messages [AAB+13, BS15]. This results in a private-key scheme that is adaptively secure for a single message query and many function queries.

[4]That is, a private-key functional encryption scheme that is adaptively-secure for a single message query and many function queries (as discussed above).

4. A pseudorandom function family $\mathcal{F}$ with a key space $\mathcal{K}$.

Our adaptively-secure scheme $\mathsf{Ad} = (\mathsf{Ad.Setup}, \mathsf{Ad.KeyGen}, \mathsf{Ad.Enc}, \mathsf{Ad.Dec})$ is defined as follows.

- **The setup algorithm:** On input 1^λ the setup algorithm $\mathsf{Ad.Setup}$ samples $(\mathsf{Sel.MPK}, \mathsf{Sel.MSK}) \leftarrow \mathsf{Sel.Setup}(1^\lambda)$, and outputs $\mathsf{Ad.MPK} = \mathsf{Sel.MPK}$ and $\mathsf{Ad.MSK} = \mathsf{Sel.MSK}$.
- **The key-generation algorithm:** On input the secret key $\mathsf{Ad.MSK} = \mathsf{Sel.MSK}$ and a function f, the key-generation algorithm $\mathsf{Ad.KeyGen}$ first samples $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$ and $\tau \leftarrow \{0,1\}^{\ell_2(\lambda)}$ uniformly and independently. Then, it computes and outputs $\mathsf{Ad}.sk_f = \mathsf{Sel}.sk_G \leftarrow \mathsf{Sel.KeyGen}(\mathsf{Sel.MSK}, G_{f,C_E,\tau})$, where the function $G_{f,C_E,\tau}$ is defined in figure 1.
- **The encryption algorithm:** On input the public key $\mathsf{Ad.MPK} = \mathsf{Sel.MPK}$ and a message m, the encryption algorithm $\mathsf{Ad.Enc}$ first samples $\mathsf{K} \leftarrow \mathcal{K}_\lambda$ and $\mathsf{OneCT.SK} \leftarrow \mathsf{OneCT.Setup}(1^\lambda)$. Then, it outputs $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$, where

$$\mathsf{CT}_0 \leftarrow \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}, m) \text{ and}$$
$$\mathsf{CT}_1 \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MPK}, (\mathsf{OneCT.SK}, \mathsf{K}, 0^\lambda, 0)).$$

- **The decryption algorithm:** On input a functional key $\mathsf{Ad}.sk_f = \mathsf{Sel}.sk_G$ and a ciphertext $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$, the decryption algorithm $\mathsf{Ad.Dec}$ first computes $\mathsf{OneCT}.sk_f \leftarrow \mathsf{Sel.Dec}(\mathsf{Sel}.sk_G, \mathsf{CT}_1)$. Then, it computes $m \leftarrow \mathsf{OneCT.Dec}(\mathsf{OneCT}.sk_f, \mathsf{CT}_0)$ and outputs m.

$G_{f,C_E,\tau}$(OneCT.SK, K, Sym.K, β):

1. If $\beta = 1$ output $\mathsf{OneCT}.sk_f \leftarrow \mathsf{Sym.Dec}(\mathsf{Sym.K}, C_E)$.
2. Otherwise, output $\mathsf{OneCT}.sk_f \leftarrow \mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}, f; \mathsf{PRF}_\mathsf{K}(\tau))$.

Figure 1: The function $G_{f,C_E,\tau}$.

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

Theorem 2. *Assuming that: (1)* Sel *is a selectively-secure public-key functional encryption scheme, (2)* OneCT *is an adaptively-secure single-ciphertext private-key functional encryption scheme, (3)* SYM *is a symmetric encryption scheme with pseudorandom ciphertexts, and (4) $\mathcal{F}$ is a pseudorandom function family, then* Ad *is an* adaptively-secure *public-key functional encryption scheme.*

Proof. We show that any PPT adversary $\mathcal{A}$ succeeds in the adaptive security game (see Definition 3) with only negligible probability. We will show this in a sequence of hybrids. We denote the advantage of the adversary in $\mathsf{Hybrid}_{i.b}$ to be the probability that the adversary outputs 1 in this hybrid and this quantity is denoted by $\mathsf{Adv}^{\mathcal{A}}_{i.b}$. For $b \in \{0,1\}$, we define the following hybrids.

$\mathsf{Hybrid}_{1.b}$: This corresponds to the real experiment when the challenger encrypts the message m_b. More precisely, the challenger produces an encryption $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$ where

$$\mathsf{CT}_0 \leftarrow \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}, m) \text{ and}$$
$$\mathsf{CT}_1 \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MPK}, (\mathsf{OneCT.SK}, \mathsf{K}, 0^\lambda, 0)).$$

$\mathsf{Hybrid}_{2.b}$: The challenger replaces the hard-coded ciphertext C_E in every functional key corresponding to a query f made by the adversary, with a symmetric key encryption of $\mathsf{OneCT}.sk_f$ (note that each key has its own different C_E). Here, $\mathsf{OneCT}.sk_f$ is the output of $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; \mathsf{PRF}_{\mathsf{K}^*}(\tau))$ and K^* is a PRF key drawn from the key space $\mathcal{K}$. Further, the symmetric encryption is computed with respect to $\mathsf{Sym.K}^*$, where $\mathsf{Sym.K}^*$ is the output of $\mathsf{Sym.Setup}(1^\lambda)$ and τ is the tag associated to the functional key of f. The same $\mathsf{Sym.K}^*$ and K^* are used while generating all the functional keys, and K^* is used for generating the challenge ciphertext $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ (that is, $\mathsf{CT}_0^* \leftarrow \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m_b)$ and $\mathsf{CT}_1^* \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, (\mathsf{OneCT.SK}^*, \mathsf{K}^*, 0^\lambda, 0)))$. The rest of the hybrid is the same as the previous hybrid, $\mathsf{Hybrid}_{1.b}$.

Note that the symmetric key $\mathsf{Sym.K}^*$ is not used for any purpose other than generating the values C_E. Therefore, the pseudorandom ciphertexts property of the symmetric scheme implies that $\mathsf{Hybrid}_{2.b}$ and $\mathsf{Hybrid}_{1.b}$ are indistinguishable.

Claim 1. *Assuming the pseudorandom ciphertexts property of* SYM*, for each* $b \in \{0,1\}$ *we have* $|\mathsf{Adv}_{1.b}^{\mathcal{A}} - \mathsf{Adv}_{2.b}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose there exists an adversary such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of SYM. The reduction internally executes the adversary by simulating the role of the challenger in the adaptive public-key FE game. It answers both the message and the functional queries made by the adversary as follows. The reduction first executes $\mathsf{OneCT.Setup}(1^\lambda)$ to obtain $\mathsf{OneCT.SK}^*$. It then samples K^* from $\mathcal{K}$. Further, the reduction generates $\mathsf{Sel.MSK}$, which is the output of $\mathsf{Sel.Setup}(1^\lambda)$ and $\mathsf{Sym.K}^*$, which is the output of $\mathsf{Sym.Setup}(1^\lambda)$. When the adversary submits a functional query f, the reduction first picks τ at random. The reduction executes $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; \mathsf{PRF}(\mathsf{K}^*(\tau)))$ to obtain $\mathsf{OneCT}.sk_f$. It then sends $\mathsf{OneCT}.sk_f$ 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 $\mathsf{OneCT}.sk_f$. The reduction then generates a selectively-secure FE functional key of $G_{f,C_E,\tau}$ and denote the result by $\mathsf{Sel}.sk_G$ which is sent to the adversary. The message queries made by the adversary are handled as in Hybrid_1. That is, the adversary submits the message-pair query (m_0, m_1) and the reduction sends $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ back to the adversary, where $\mathsf{CT}_0^* = \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m_b)$ and $\mathsf{CT}_1^* = \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1))$.

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 $\mathsf{Hybrid}_{1.b}$, otherwise we are in $\mathsf{Hybrid}_{2.b}$. 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 claim. ■

$\mathsf{Hybrid}_{3.b}$: The challenger modifies the challenge ciphertext $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ so that CT_1^* is an encryption of $(0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)$. The ciphertext component CT_0^* is not modified (i.e., $\mathsf{CT}_0^* = \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m_b)$). The rest of the hybrid is the same as the previous hybrid, $\mathsf{Hybrid}_{2.b}$.

Note that the functionality of the functional keys generated using the underlying selectively-secure scheme is unchanged with the modified CT_1^*. Therefore, its selective security implies that $\mathsf{Hybrid}_{3.b}$ and $\mathsf{Hybrid}_{2.b}$ are indistinguishable.

Claim 2. *Assuming the selective security of* Sel*, for each* $b \in \{0,1\}$ *we have* $|\mathsf{Adv}_{2.b}^{\mathcal{A}} - \mathsf{Adv}_{3.b}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose the claim is not true for some adversary $\mathcal{A}$, we construct a reduction that breaks the security of Sel. Our reduction will internally execute $\mathcal{A}$ by simulating the role of the challenger of the adaptive FE game.

Our reduction first executes $\mathsf{OneCT.Setup}(1^\lambda)$ to obtain $\mathsf{OneCT.SK}^*$. It then samples K^* from $\mathcal{K}$. It also executes $\mathsf{Sym.Setup}(1^\lambda)$ to obtain $\mathsf{Sym.K}^*$. The reduction then sends the message pair $\big((\mathsf{OneCT.SK}^*, \mathsf{K}^*, 0^\lambda, 0), (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)\big)$ to the challenger of the selective game. The challenger replies back with the public key $\mathsf{Sel.MPK}$ and the challenge ciphertext CT_1^*. The reduction is now ready to interact with the adversary $\mathcal{A}$. If $\mathcal{A}$ makes a functional query f then the reduction constructs the circuit $G_{f,C_E,\tau}$ as in $\mathsf{Hybrid}_{2.b}$. It then queries the challenger of the selective game with the function G and in return it gets the key $\mathsf{Sel}.sk_G$. The reduction then sets $\mathsf{Ad}.sk_f$ to be $\mathsf{Sel}.sk_G$ which it then sends back to $\mathcal{A}$. If $\mathcal{A}$ submits a message pair (m_0, m_1), the reduction executes $\mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m_0)$ to obtain CT_0^*. It then sends the ciphertext $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ to the adversary. The output of the reduction is the output of $\mathcal{A}$.

We claim that the reduction is a legal adversary in the selective security game of Sel, i.e., for challenge message query $(M_0 = (\mathsf{OneCT.SK}^*, \mathsf{K}^*, 0^\lambda, 0),\ M_1 = (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1))$ and every functional query of the form $G_{f,C_E,\tau}$ made by the reduction, we have that $G_{f,C_E,\tau}(M_0) = G_{f,C_E,\tau}(M_1)$: By definition, $G_{f,C_E,\tau}(M_0)$ is the functional key of f, with respect to key $\mathsf{OneCT.SK}^*$ and randomness $\mathsf{PRF}_{\mathsf{K}^*}(\tau)$. Further, $G_{f,C_E,\tau}(M_1)$ is the decryption of C_E which is nothing but the functional key of f, with respect to key $\mathsf{OneCT.SK}^*$ and randomness $\mathsf{PRF}_{\mathsf{K}^*}(\tau)$. This proves that the reduction is a legal adversary in the selective security game.

If the challenger of the selective game sends back an encryption of $(\mathsf{OneCT.SK}^*, \mathsf{K}^*, 0^\lambda, 0)$ then we are in $\mathsf{Hybrid}_{2.b}$ else if the challenger encrypts $(0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)$ then we are in $\mathsf{Hybrid}_{3.b}$. By our hypothesis, this means the reduction breaks the security of the selective game with non-negligible probability that contradicts the security of Sel. This completes the proof of the claim. ■

$\mathsf{Hybrid}_{4.b}$: For every function query f made by the adversary, the challenger generates C_E by executing $\mathsf{Sym.Enc}(\mathsf{Sym.K}^*, \mathsf{OneCT}.sk_f)$, with $\mathsf{OneCT}.sk_f$ being the output of $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; R)$, where R is picked at random. The rest of the hybrid is the same as the previous hybrid.

Note that the PRF key K^* is not explicitly needed in the previous hybrid, and therefore the pseudorandomness of $\mathcal{F}$ implies that $\mathsf{Hybrid}_{4.b}$ and $\mathsf{Hybrid}_{3.b}$ are indistinguishable.

Claim 3. *Assuming that $\mathcal{F}$ is a pseudorandom function family, for each $b \in \{0,1\}$ we have* $|\mathsf{Adv}_{3.b}^{\mathcal{A}} - \mathsf{Adv}_{4.b}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose the claim is false for some PPT adversary $\mathcal{A}$, we construct a reduction that internally executes $\mathcal{A}$ and breaks the security of the pseudorandom function family $\mathcal{F}$. The reduction simulates the role of the challenger of the adaptive game when interacting with $\mathcal{A}$. The reduction answers the functional queries, made by the adversary as follows; the message queries are answered as in $\mathsf{Hybrid}_{3.b}$ (or $\mathsf{Hybrid}_{4.b}$). 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 $\mathsf{Sym.Enc}(\mathsf{Sym.K}^*, \mathsf{OneCT}.sk_f)$, where $\mathsf{OneCT}.sk_f = \mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; R^*)$. The reduction then proceeds as in the previous hybrids to compute the functional key $\mathsf{Ad}.sk_f$ which it then sends to $\mathcal{A}$.

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

We now conclude the proof of the theorem by showing that $\mathsf{Hybrid}_{4.0}$ is computationally indistinguishable from $\mathsf{Hybrid}_{4.1}$ based on the adaptive security of the underlying single-ciphertext scheme.

Claim 4. *Assuming the adaptive security of the scheme* OneCT, *we have* $|\mathsf{Adv}_{4.0}^{\mathcal{A}} - \mathsf{Adv}_{4.1}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose there exists a PPT adversary $\mathcal{A}$, such that the claim is false. We design a reduction $\mathcal{B}$ that internally executes $\mathcal{A}$ to break the adaptive security of OneCT.

The reduction simulates the role of the challenger of the adaptive public-key FE game. It answers both the functional as well as message queries made by the adversary as follows. If $\mathcal{A}$ makes a functional query f then it forwards it to the challenger of the adaptively-secure single-ciphertext FE scheme. In return it receives $\mathsf{OneCT}.sk_f$. It then encrypts it using the symmetric encryption scheme, where the symmetric key is picked by the reduction itself, and denote the resulting ciphertext to be C_E. The reduction then constructs the circuit $G_{f,C_E,\tau}$, with τ being picked at random, as in the previous hybrids. Finally, the reduction computes the selective public-key functional key of $G_{f,C_E,\tau}$, where the reduction itself picks the master secret key of selective public-key FE scheme. The resulting functional key is then sent to $\mathcal{A}$. If $\mathcal{A}$ makes a message-pair query (m_0, m_1), the reduction forwards this message pair to the challenger of the adaptive game. In response it receives CT_0^*. The reduction then generates CT_1^* on its own where CT_1^* is the selective FE encryption of $(0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)$. The reduction then sends $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ to $\mathcal{A}$. The output of the reduction is the output of $\mathcal{A}$.

We note that the reduction is a legal adversary in the adaptive game of OneCT, i.e., for every challenge message query (m_0, m_1), functional query f, we have that $f(m_0) = f(m_1)$: this follows from the fact that (i) the functional queries (resp., challenge message query) made by the adversary (of Ad) is the same as the functional queries (resp., challenge message query) made by the reduction, and (ii) the adversary (of Ad) is a legal adversary. This proves that the reduction is a legal adversary in the adaptive game.

If the challenger sends an encryption of m_0 then we are in $\mathsf{Hybrid}_{4.0}$ and if the challenger sends an encryption of m_1 then we are in $\mathsf{Hybrid}_{4.1}$. From our hypothesis, this means that the reduction breaks the security of OneCT. This proves the claim. ■

■

4 From Shallow Circuits to All Circuits

In this section we show that a functional encryption scheme that supports functions computable by shallow circuits can be transformed into one that supports functions computable by arbitrarily deep circuits. In particular, the shallow class can be any class in which weak pseudorandom functions can be computed and has some composition properties.[5] For concreteness we consider here the class NC^1, which can compute weak pseudorandom functions under standard cryptographic assumptions such as DDH or LWE (a lower complexity class such as TC^0 is also sufficient under standard assumptions). We focus here on private-key functional encryption schemes, and note that an essentially identical transformation applies for public-key scheme.

While we present a direct reduction below, we notice that this property can be derived from the transformation in Section 3, by recalling some properties of Gurbunov et al.'s [GVW12] single-key functional encryption scheme. One can verify that their setup algorithm can be implemented in NC^1 (under the assumption that it can evaluate weak pseudorandom functions), regardless of the depth of the function being implemented. This property carries through even after applying the function privacy transformation of Brakerski and Segev [BS15]. Lastly, to implement our approach we need a symmetric encryption scheme with decryption in NC^1, which again translates to the evaluation of a weak pseudorandom function [NR04,BPR12].

(Computational) Randomized encodings [IK00, AIK05]. A (computational) randomized encoding scheme for a function class $\mathcal{F}$ consists of two PPT algorithms $(\mathsf{RE.Encode}, \mathsf{RE.Decode})$. The PPT algorithm $\mathsf{RE.Encode}$ takes as input $(1^\lambda, F, x, r)$, where λ is the security parameter, $F : \{0,1\}^\lambda \to \{0,1\}$ is a function in $\mathcal{F}$, instance $x \in \{0,1\}^\lambda$ and randomness r. The output is denoted by $\hat{F}(x; r)$. The PPT algorithm $\mathsf{RE.Decode}$ takes as input $\hat{F}(x; r)$ and outputs $y = F(x)$.

The security property states that there exists a PPT algorithm Sim that takes as input $(1^\lambda, F(x))$ and outputs $\mathsf{SimOut}_{F(x)}$ such that any PPT adversary cannot distinguish the distribution $\{\hat{F}(x; r)\}$ from the distribution $\{\mathsf{SimOut}_{F(x)}\}$. The following corollary is derived from applying Yao's garbled circuit technique using a weak PRF based encryption algorithm.

Corollary 1. *Assuming a family of weak pseudorandom functions that can be evaluated in* NC^1*, there exists a randomized encoding scheme* $(\mathsf{RE.Encode}, \mathsf{RE.Decode})$ *for the class of polynomial size circuits, such that* $\mathsf{RE.Encode}$ *is computable in* NC^1*.*

Our transformation. Let $\mathcal{NCFE} = (\mathsf{NCFE.Setup}, \mathsf{NCFE.KeyGen}, \mathsf{NCFE.Enc}, \mathsf{NCFE.Dec})$ be a private-key functional encryption scheme for the class NC^1. We assume that $\mathcal{NCFE}$ supports functions with multi-bit outputs, as otherwise it is always possible to produce a functional key for each output bit separately. We also use a pseudorandom function family denoted by $\mathcal{F} = \{\mathsf{PRF}_\mathsf{K}(\cdot)\}_{\mathsf{K} \in \mathcal{K}}$ and a symmetric encryption

[5] Similarly to the class WEAK defined in [App14].

scheme $\mathsf{SYM} = (\mathsf{Sym.Setup}, \mathsf{Sym.Enc}, \mathsf{Sym.Dec})$. We construct a private-key functional encryption scheme $\mathcal{PFE} = (\mathsf{PFE.Setup}, \mathsf{PFE.KeyGen}, \mathsf{PFE.Enc}, \mathsf{PFE.Dec})$ as follows.

- **The setup algorithm:** On input 1^λ the algorithm $\mathsf{PFE.Setup}$ samples and outputs $MSK \leftarrow \mathsf{NCFE.Setup}(1^\lambda)$.
- **The key-generation algorithm:** On input the secret key MSK and a circuit F, the algorithm $\mathsf{PFE.KeyGen}$ first samples $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$ and $\tau \leftarrow \{0,1\}^\lambda$ uniformly and independently. Then, it computes a functional key $SK_G \leftarrow \mathsf{NCFE.KeyGen}(MSK, G_{F,C_E,\tau})$, where the function $G_{F,C_E,\tau}$ is defined in figure 2, and outputs (SK_G, F, C_E, τ).
- **The encryption algorithm:** On input the secret key MSK and a message x, the algorithm $\mathsf{PFE.Enc}$ first samples $K_P \leftarrow \{0,1\}^\lambda$, and then computes and outputs $C \leftarrow \mathsf{NCFE.Enc}(MSK, (x, K_P, 0^\lambda, 0))$.
- **The decryption algorithm:** On input a functional key $SK_F = (SK_G, F, C_E, \tau)$ and a ciphertext C, the decryption algorithm $\mathsf{PFE.Dec}$ computes $\widehat{F}(x) \leftarrow \mathsf{NCFE.Dec}(SK_G, (F, C_E, \tau), C)$ and then outputs $\mathsf{RE.Decode}(\widehat{F}(x))$.

$\boldsymbol{G_{f,C_E,\tau}(x, K_P, K_E, \beta)}$:

1. If $\beta = 1$ output $\mathsf{Sym.Dec}_{K_E}(C_E)$.
2. Otherwise, output $\widehat{F}(x; \mathsf{PRF}_{K_P}(\tau)) = \mathsf{RE.Encode}(F, x; \mathsf{PRF}_{K_P}(\tau))$.

Figure 2: The function $G_{f,C_E,\tau}$.

The correctness of the above scheme easily follows from that of its underlying building blocks, and in the remainder of this section we provide a sketch for proving the following theorem:

Theorem 3. *Assuming that: (1) $\mathcal{NCFE}$ is a selectively-secure private-key functional encryption scheme for* NC^1*, (2)* SYM *is a symmetric encryption scheme with pseudorandom ciphertexts whose decryption circuit is in* NC^1*, (3)* PRF *is a weak pseudorandom function family which can be evaluated in* NC^1*, and (4)* $(\mathsf{RE.Encode}, \mathsf{RE.Decode})$ *is a randomized encoding scheme with encoding in* NC^1*, then $\mathcal{PFE}$ is a selectively-secure private-key functional encryption scheme for P.*

Proof Sketch. The proof proceeds by a sequence of hybrids. For simplicity, we consider the case when the adversary submits a single challenge pair (m_0, m_1), and the argument can be easily generalized to the case of multiple challenges.

$\mathsf{\textbf{Hybrid}}_\mathbf{0}$**:** This corresponds to the real experiment where the challenger sends an encryption of m_0 to the adversary.

$\mathsf{\textbf{Hybrid}}_\mathbf{1}$**:** For every functional query F, the challenger replaces C_E with a symmetric encryption $\mathsf{Sym.Enc}(K_E, \widehat{F}(m_0; \mathsf{PRF}_{K_P}(t))$ in the functional key for F. By a sequence of intermediate hybrids (as many as the number of function queries), Hybrid_1 can be shown to be computationally indistinguishable from Hybrid_0 based on the pseudorandom ciphertexts property of the symmetric encryption scheme.

$\mathsf{\textbf{Hybrid}}_\mathbf{2}$**:** The challenge ciphertext will consist of an encryption of $(m_0, 0, K_E, 1)$ instead of $(m_0, K_P, 0^\lambda, 0)$. This hybrid is computationally indistinguishable from Hybrid_1 by the security of the underlying functional encryption scheme.

$\mathsf{\textbf{Hybrid}}_\mathbf{3}$**:** For every function query F, the challenger replaces C_E in all the functional keys with $\mathsf{Sym.Enc}(K_E, \widehat{F}(m_0; r))$ for a uniform r. By a sequence of intermediate hybrids (as many as the number of function queries), Hybrid_3 can be shown to be computationally indistinguishable from Hybrid_2 based on the security of PRF.

Hybrid_4: Finally, for every function query F, the challenger replaces $\widehat{F}(m_0; r)$ in the ciphertext hardwired in the functional key for F by the simulated randomized encoding $\mathsf{Sim}(1^\lambda, F(m_0))$. By a sequence of intermediate hybrids (as many as the number of function queries), Hybrid_4 can be shown to be computationally indistinguishable from Hybrid_3 based on the security of randomized encodings. Note that the this hybrid does not depend on whether m_0 or m_1 was encrypted since for all function queries F it holds that $F(m_0) = F(m_1)$, and this proves the security of $\mathcal{PFE}$.

References

[AAB+13] S. Agrawal, S. Agrawal, S. Badrinarayanan, A. Kumarasubramanian, M. Prabhakaran, and A. Sahai. Function private functional encryption and property preserving encryption: New definitions and positive results. Cryptology ePrint Archive, Report 2013/744, 2013.

[AIK05] B. Applebaum, Y. Ishai, and E. Kushilevitz. Computationally private randomizing polynomials and their applications. In *20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA*, pages 260–274. IEEE Computer Society, 2005.

[App14] B. Applebaum. Bootstrapping obfuscators via fast pseudorandom functions. In *Advances in Cryptology – ASIACRYPT '14*, pages 162–172, 2014.

[BB04] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT '04*, pages 223–238, 2004.

[BCO+04] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology – EUROCRYPT '04*, pages 506–522, 2004.

[BCP14] E. Boyle, K. Chung, and R. Pass. On extractability obfuscation. In *Proceedings of the 11th Theory of Cryptography Conference*, pages 52–73, 2014.

[BF03] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Advances in Cryptology – EUROCRYPT '12*, pages 719–737, 2012.

[BS15] Z. Brakerski and G. Segev. Function-private functional encryption in the private-key setting. To appear in *Proceedings of the 12th Theory of Cryptography Conference*, 2015. Available as Cryptology ePrint Archive, Report 2014/550.

[BSW11] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Proceedings of the 8th Theory of Cryptography Conference*, pages 253–273, 2011.

[CIJ+13] A. D. Caro, V. Iovino, A. Jain, A. O'Neill, O. Paneth, and G. Persiano. On the achievability of simulation-based security for functional encryption. In *Advances in Cryptology – CRYPTO '13*, pages 519–535, 2013.

[GGH+13] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 40–49, 2013.

[GGH+14a] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure attribute based encryption from multilinear maps. *IACR Cryptology ePrint Archive*, 2014:622, 2014.

[GGH+14b] S. Garg, C. Gentry, S. Halevi, and M. Zhandry. Fully secure functional encryption without obfuscation. Cryptology ePrint Archive, Report 2014/666, 2014.

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

[GHR+14] C. Gentry, S. Halevi, M. Raykova, and D. Wichs. Outsourcing private RAM computation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 404–413. IEEE Computer Society, 2014.

[GKP+13] S. Goldwasser, Y. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 555–564, 2013.

[Gol04] O. Goldreich. Foundations of Cryptography – Volume 2: Basic Applications. Cambridge University Press, 2004.

[GPS+06] V. Goyal, O. Pandey, A. Sahai, and B. 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, 2006.

[GVW12] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology – CRYPTO '12*, pages 162–179, 2012.

[GVW13] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 545–554, 2013.

[IK00] Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 294–304, 2000.

[KSW08] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology – EUROCRYPT '08*, pages 146–162, 2008.

[LOS+10] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Advances in Cryptology – EUROCRYPT '10*, pages 62–91, 2010.

[LW10] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *Proceedings of the 7th Theory of Cryptography Conference*, pages 455–479, 2010.

[LW12] A. B. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *Advances in Cryptology – CRYPTO '12*, pages 180–198, 2012.

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

[O'N10] A. O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

[SS10] A. Sahai and H. Seyalioglu. Worry-free encryption: functional encryption with public keys. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pages 463–472, 2010.

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Advances in Cryptology – EUROCRYPT '05*, pages 457–473, 2005.

[Wat09] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Advances in Cryptology – CRYPTO '09*, pages 619–636, 2009.

[Wat14] B. Waters. A punctured programming approach to adaptively secure functional encryption. Cryptology ePrint Archive, Report 2014/588, 2014.

A Preliminaries (Cont.)

A.1 Private-Key Functional Encryption

A private-key functional encryption (FE) scheme Priv 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 $(\mathsf{Priv.Setup}, \mathsf{Priv.KeyGen}, \mathsf{Priv.Enc}, \mathsf{Priv.Dec})$ of PPT algorithms with the following properties:

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

In terms of 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 $f \in \mathcal{F}_\lambda$ it holds that

$$\mathsf{Priv.Dec}(\mathsf{Priv.KeyGen}(\mathsf{Priv.MSK}, f), \mathsf{Priv.Enc}(\mathsf{Priv.MSK}, m)) = f(m)$$

with probability at least $1 - \mathsf{negl}(\lambda)$, where $\mathsf{Priv.MSK} \leftarrow \mathsf{Priv.Setup}(1^\lambda)$, and the probability is taken over the random choices of all algorithms.

We consider the standard selective and adaptive indistinguishability-based notions for private-key functional encryption (see, for example, [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 $f(m_0) = f(m_1)$ and to an encryption oracle.

Definition 4 (Selective security)**.** *A private-key functional encryption scheme* $\Pi = (\mathsf{Sel.Setup}, \mathsf{Sel.KeyGen}, \mathsf{Sel.Enc}, \mathsf{Sel.Dec})$ *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*

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

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

1. **Setup phase:** *The challenger samples* $\mathsf{Sel.MSK} \leftarrow \mathsf{Sel.Setup}(1^\lambda)$.
2. **Message queries:** *On input* 1^λ *the adversary submits* $((m_1^{(0)}, \ldots, m_p^{(0)}), (m_1^{(1)}, \ldots, m_p^{(1)}))$ *for some polynomial* $p = p(\lambda)$*. The challenger replies with* $(c_1, \ldots, c_p)$*, where* $c_i \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, m_i^{(b)})$ *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* $\mathsf{Sel}.sk_f \leftarrow \mathsf{Sel.KeyGen}(\mathsf{Sel.MSK}, f)$.

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

Definition 5 (Adaptive security)**.** *A private-key functional encryption scheme* $\Pi = (\mathsf{Ad.Setup}, \mathsf{Ad.KeyGen}, \mathsf{Ad.Enc}, \mathsf{Ad.Dec})$ *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* adaptively secure *if for any PPT adversary* $\mathcal{A}$ *there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda) = \left| \Pr[\mathsf{Expt}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Expt}^{\mathsf{Ad}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1] \right| \leq \mathsf{negl}(\lambda),$$

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

1. **Setup phase:** *The challenger samples* $\mathsf{Ad.MSK} \leftarrow \mathsf{Ad.Setup}(1^\lambda)$*.*
2. **Query phase:** *The adversary adaptively queries the challenger with message queries and function queries, in an arbitrary order, as follows:*
   - **Message queries***: The adversary submits* (m_0, m_1) *such that* $f(m_0) = f(m_1)$ *for all function queries* f *made so far. The challenger replies with* $\mathsf{CT} = \mathsf{Ad.Enc}(\mathsf{Ad.MSK}, m_b)$*.*
   - **Function queries***: The adversary submits a function* f *such that* $f(m_0) = f(m_1)$ *for all message queries* (m_0, m_1) *made so far. The challenger replies with* $\mathsf{Ad}.sk_f \leftarrow \mathsf{Ad.KeyGen}(\mathsf{Ad.MSK}, f)$*.*
3. **Output phase:** *The adversary outputs a bit b' which is defined as the output of the experiment.*

B Our Transformation in the Private-Key Setting

The exact same transformation as above works in the private-key setting as well. Namely, given a private-key selectively secure FE, we obtain a private-key adaptively secure FE. The transformation is identical with the obvious exception that there is no public-key, and the master secret key is used for both encryption and key generation. We denote the selectively-secure FE that we use by $\mathsf{Sel} = (\mathsf{Sel.Setup}, \mathsf{Sel.KeyGen}, \mathsf{Sel.Enc}, \mathsf{Sel.Dec})$. The adaptively-secure FE that we construct is denoted by $\mathsf{Ad} = (\mathsf{Ad.Setup}, \mathsf{Ad.KeyGen}, \mathsf{Ad.Enc}, \mathsf{Ad.Dec})$.

- **The setup algorithm:** On input 1^λ, the algorithm $\mathsf{Ad.Setup}$ executes $\mathsf{Sel.Setup}(1^\lambda)$ to obtain $\mathsf{Sel.MSK}$. Output $\mathsf{Ad.MSK} = \mathsf{Sel.MSK}$.
- **The key-generation algorithm:** On input secret key $\mathsf{Ad.MSK} = \mathsf{Sel.MSK}$ and a function f, the algorithm $\mathsf{Ad.KeyGen}$ first samples a uniformly random string $C_E \leftarrow \{0,1\}^{\ell_1(\lambda)}$ and a uniformly random tag $\tau \leftarrow \{0,1\}^{\ell_2(\lambda)}$. It then computes $\mathsf{Sel}.sk_G \leftarrow \mathsf{Sel.KeyGen}(\mathsf{Sel.MSK}, G_{f,C_E,\tau})$ and outputs $\mathsf{Ad}.sk_f = \mathsf{Ad}.sk_G$, where $G_{f,C_E,\tau}$ is as defined in Figure 3.

$G_{f,C_E,\tau}$(OneCT.SK, K, Sym.K, β):

1. If $\beta = 0$ output $\mathsf{OneCT}.sk_f \leftarrow \mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}, f; \mathsf{PRF}_{\mathsf{K}}(\tau))$.
2. Otherwise, output $\mathsf{Sym.Dec}(\mathsf{Sym.K}, C_E)$.

Figure 3: The function $G_{f,C_E,\tau}$.

- **The encryption algorithm:** On input secret key $\mathsf{Ad.MSK} = \mathsf{Sel.MSK}$ and a message m, the algorithm $\mathsf{Ad.Enc}$ samples $\mathsf{OneCT.SK}$ by executing $\mathsf{OneCT.Setup}(1^\lambda)$ and $\mathsf{K} \leftarrow \mathcal{K}_\lambda$. Then, it outputs $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$, where

$$\mathsf{CT}_0 \leftarrow \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}, m)$$
$$\mathsf{CT}_1 \leftarrow \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, M = (\mathsf{OneCT.SK}, \mathsf{K}, 0^\lambda, 0)).$$

- **The decryption algorithm:** On input a functional key $\mathsf{Ad}.sk_f = \mathsf{Sel}.sk_G$ and a ciphertext $\mathsf{CT} = (\mathsf{CT}_0, \mathsf{CT}_1)$, the decryption algorithm Ad.Dec first computes $\mathsf{OneCT}.sk_f \leftarrow \mathsf{Sel.Dec}(\mathsf{Sel}.sk_G, \mathsf{CT}_1)$. Then, it computes $m \leftarrow \mathsf{OneCT.Dec}(\mathsf{OneCT}.sk_f, \mathsf{CT}_0)$ and outputs m.

The correctness is straightforward. The proof of security in this case is slightly more complicated than its public-key counterpart. Since in the symmetric setting, the adversary is allowed to make multiple message queries, we have to employ a sequence of hybrids, handling each message query at a time. Each of these hybrids is identical to our proof of Theorem 2 above. We prove the following theorem.

Theorem 4. *Assuming that: (1)* Sel *is a selectively-secure private-key functional encryption scheme, (2)* OneCT *is an adaptively-secure single-ciphertext private-key functional encryption scheme, (3)* SYM *is a symmetric encryption scheme with pseudorandom ciphertexts, and (4) $\mathcal{F}$ is a pseudorandom function family, then* Ad *is an* adaptively-secure *private-key functional encryption scheme.*

Proof. We show that any PPT adversary $\mathcal{A}$ succeeds in the adaptive security game of Ad with only negligible probability. We will show this in a sequence of hybrids. We denote the advantage of the adversary in $\mathbf{Hybrid}_i^j$ to be the probability that the adversary outputs 1 in that hybrid and this quantity is denoted by $\mathsf{Adv}_{i,j}^{\mathcal{A}}$.

We define the hybrids, $\mathsf{Hybrid}_{i.b}^j$, for $j \in [p]$, $i \in [4]$, and $b \in \{0,1\}$, where p denotes the number of message queries made by $\mathcal{A}$. We then prove the following quantities:

1. $|\mathsf{Adv}_0^{\mathcal{A}} - \mathsf{Adv}_{1.0,1}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.
2. $|\mathsf{Adv}_{i.0,j}^{\mathcal{A}} - \mathsf{Adv}_{i+1.0,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$, for all $i \in [3], j \in [p]$.
3. $|\mathsf{Adv}_{i.1,j}^{\mathcal{A}} - \mathsf{Adv}_{i+1.1,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$, for all $i \in [3], j \in [p]$.
4. $|\mathsf{Adv}_{4.0,j}^{\mathcal{A}} - \mathsf{Adv}_{4.1,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$, for all $j \in [p]$.
5. $|\mathsf{Adv}_{1.0,j+1}^{\mathcal{A}} - \mathsf{Adv}_{1.1,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$, for all $j \in [p-1]$.
6. $|\mathsf{Adv}_{1.1,p}^{\mathcal{A}} - \mathsf{Adv}_5^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.

We now describe the hybrids.

$\mathbf{Hybrid_0}$**:** This corresponds to the real experiment when the challenger uses the encryption oracle, parameterized by bit 0, to generate the challenge ciphertexts. That is, for all message queries of the form (m_0, m_1), the challenger sends an encryption of m_0 to the adversary. The output of this hybrid is the same as the output of the adversary.

$\mathbf{Hybrid}_{1.b}^j$ **for $b \in \{0,1\}, j \in [p]$:** This is the same as the hybrid $\mathsf{Hybrid}_{1.b}^{j-1}$ (if $j = 1$ then we refer to Hybrid_0) except that the challenger encrypts the b^{th} message in the j^{th} message pair query submitted by the adversary. More precisely, the only change is the following: If the adversary submits the j^{th} message pair (m_0, m_1) to the challenger, the challenger then sends the challenge ciphertext CT^* back to the adversary, where CT^* is the encryption of message m_b. We observe that the hybrid $\mathsf{Hybrid}_{1.0}^1$ is identical to Hybrid_0 and also, $\mathsf{Hybrid}_{1.1}^{j-1}$ is identical to $\mathsf{Hybrid}_{1.0}^j$, for $j \in [p]$ and $j > 1$.

$\mathbf{Hybrid}_{2.b}^j$ **for $b \in \{0,1\}, j \in [p]$:** This is identical to $\mathsf{Hybrid}_{1.b}^j$ except for the following change. The challenger replaces C_E in every functional key, corresponding to the query f made by the adversary, with a symmetric encryption of $\mathsf{OneCT}.sk_f$, where $\mathsf{OneCT}.sk_f$ is the output of $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; \mathsf{PRF}_{\mathsf{K}^*}(\tau))$ and K^* is a PRF key sampled from the key space $\mathcal{K}$. Further the symmetric encryption is computed with respect to $\mathsf{Sym.K}^*$, where $\mathsf{Sym.K}^*$ is the output of $\mathsf{Sym.Setup}(1^\lambda)$ and τ is the tag associated to the functional key of f. We emphasize that the same $\mathsf{Sym.K}^*$ and K^* is used while generating all the functional keys.

Claim 5. *Assuming the pseudorandom ciphertexts property of* SYM*, for every PPT adversary $\mathcal{A}$, for $b \in \{0,1\}$, $j \in [p]$, we have $|\mathsf{Adv}_{1.b,j}^{\mathcal{A}} - \mathsf{Adv}_{2.b,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists an adversary such that the difference in the advantages is non-negligible, then we construct a reduction that can break the security of SYM. The reduction internally executes the adversary by simulating the role of the challenger in the adaptive private-key FE game. It answers both the message and the functional queries made by the adversary as follows. The reduction first executes $\mathsf{OneCT.Setup}(1^\lambda)$ to obtain $\mathsf{OneCT.SK}^*$ It then samples K^* from $\mathcal{K}$. Further, the reduction generates $\mathsf{Sel.MSK}$, which is the output of $\mathsf{Sel.Setup}(1^\lambda)$ and $\mathsf{Sym.K}^*$, which is the output of $\mathsf{Sym.Setup}(1^\lambda)$. When the adversary submits a functional query f, the reduction first picks τ at random. The reduction executes $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; \mathsf{PRF}(\mathsf{K}^*(\tau)))$ to obtain $\mathsf{OneCT}.sk_f$. It then sends $\mathsf{OneCT}.sk_f$ 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 $\mathsf{OneCT}.sk_f$. The reduction then generates a selectively-secure FE functional key of $G_{f,C_E,\tau}$ and denote the result by $\mathsf{Sel}.sk_G$ which is sent to the adversary. The message queries made by the adversary are handled as in $\mathsf{Hybrid}^j_{1.b}$. That is, the adversary submits the i^{th} message-pair query of the form (m^i_0, m^i_1) and the reduction sends $\mathsf{CT}^* = (\mathsf{CT}^*_0, \mathsf{CT}^*_1)$ back to the adversary, where $\mathsf{CT}^*_0 = \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m_{b_i})$ and $\mathsf{CT}^*_1 = \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1))$; we define $b_i = 1$ for $i < j$, $b_j = b$ and for $i > j$, we have $b_i = 0$.

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 $\mathsf{Hybrid}^j_{1.b}$, otherwise we are in $\mathsf{Hybrid}^j_{2.b}$. 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. This proves the claim. ■

$\mathbf{Hybrid}^j_{3.b}$ **for** $b \in \{0,1\}, j \in [p]$**:** The challenger modifies the j^{th} challenge ciphertext $\mathsf{CT}^* = (\mathsf{CT}^*_0, \mathsf{CT}^*_1)$. In particular it generates CT^*_1 using the message $(0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)$ instead of $(\mathsf{OneCT.SK}^*, \mathsf{K}^*, 0^\lambda, 0)$. The ciphertext component CT^*_0 is generated the same way as in the previous hybrid, $\mathsf{Hybrid}^j_{2.b}$. More formally, the j^{th} challenge ciphertext is now $\mathsf{CT}^* = (\mathsf{CT}^*_0 = \mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*, m^j_b)$, $\mathsf{CT}^*_1 = \mathsf{Sel.Enc}(\mathsf{Sel.MSK}, (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1))$. The rest of the hybrid is the same as the previous hybrid, $\mathsf{Hybrid}^j_{2.b}$.

Claim 6. *Assuming the selective security of* Sel*, for every PPT adversary* $\mathcal{A}$*, for* $b \in \{0,1\}$*,* $j \in [p]$*, we have* $|\mathsf{Adv}^{\mathcal{A}}_{2.b,j} - \mathsf{Adv}^{\mathcal{A}}_{3.b,j}| \leq \mathsf{negl}(\lambda)$.

Proof. Suppose the claim is not true for some PPT adversary $\mathcal{A}$, we construct a reduction that breaks the security of Sel. Our reduction will internally execute $\mathcal{A}$ by simulating the role of the challenger of the adaptive FE game.

For every $i \in [p]$, the reduction does the following. It first executes $\mathsf{OneCT.Setup}(1^\lambda)$ to obtain $\mathsf{OneCT.SK}^*_i$. It then samples K^*_i from $\mathcal{K}$. It also executes $\mathsf{Sym.Setup}(1^\lambda)$ to obtain $\mathsf{Sym.K}^*_i$. If $i \neq j$, the reduction then sends the message pair $\big((\mathsf{OneCT.SK}^*_i, \mathsf{K}^*_i, 0^\lambda, 0)), (\mathsf{OneCT.SK}^*_i, \mathsf{K}^*_i, 0^\lambda, 0)\big)$ to the challenger of the selective game and if $i = j$, the reduction instead sends the message pair $\big((\mathsf{OneCT.SK}^*_j, \mathsf{K}^*_j, 0^\lambda, 0), (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*_j, 1)\big)$. For the i^{th} message query, the challenger responds back with the challenge ciphertext $\mathsf{CT}^*_{1,i}$.

The reduction is now ready to interact with the adversary $\mathcal{A}$. If $\mathcal{A}$ makes a functional query f then the reduction constructs the circuit $G_{f,C_E,\tau}$ as in $\mathsf{Hybrid}^j_{2.b}$. It then queries the challenger of the selective game with the function G and in return it gets the key $\mathsf{Sel}.sk_G$. The reduction then sets $\mathsf{Ad}.sk_f$ to be $\mathsf{Sel}.sk_G$ which it then sends back to $\mathcal{A}$. The message queries made by $\mathcal{A}$ are handled as follows. When $\mathcal{A}$ submits the i^{th} message pair (m^i_0, m^i_1), the reduction executes $\mathsf{OneCT.Enc}(\mathsf{OneCT.SK}^*_i, m^i_0)$ to obtain $\mathsf{CT}^*_{0,i}$. It then sends the ciphertext $\mathsf{CT}^* = (\mathsf{CT}^*_{0,i}, \mathsf{CT}^*_{1,i})$ to the adversary. The output of the reduction is the output of $\mathcal{A}$.

We claim that the reduction is a legal adversary in the selective security game of Sel. To argue this, note that we only need to consider the j^{th} message query since the left and the right messages in all other message queries are the same. For the j^{th} message query $(M_0 = (\mathsf{OneCT.SK}^*_j, \mathsf{K}^*_j, 0^\lambda, 0), M_1 = (0^\lambda, 0^\lambda, \mathsf{Sym.K}^*_j, 1))$ and every functional query of the form $G_{f,C_E,\tau}$ made by the reduction, we have that $G_{f,C_E,\tau}(M_0) = G_{f,C_E,\tau}(M_1)$: By definition, $G_{f,C_E,\tau}(M_0)$ is the functional key of f, with respect to key $\mathsf{OneCT.SK}^*_j$ and randomness $\mathsf{PRF}_{\mathsf{K}^*_j}(\tau)$. Further, $G_{f,C_E,\tau}(M_1)$ is the decryption of C_E which is nothing but the functional key of f, with respect to key $\mathsf{OneCT.SK}^*_j$ and randomness $\mathsf{PRF}_{\mathsf{K}^*_j}(\tau)$. This proves that the reduction is a legal adversary in the selective security game.

If the challenger of the selective game sends back an encryption of $(\mathsf{OneCT.SK}_j^*, \mathsf{K}_j^*, 0^\lambda, 0)$ then we are in $\mathsf{Hybrid}_{2.b}^j$ else if the challenger encrypts $(0^\lambda, 0^\lambda, \mathsf{Sym.K}_j^*, 1)$ then we are in $\mathsf{Hybrid}_{3.b}^j$. By our hypothesis, this means the reduction breaks the security of the selective game with non-negligible probability that contradicts the security of Sel. This completes the proof of the claim. ■

$\mathsf{Hybrid}_{4.b}^j$ for $b \in \{0,1\}, j \in [p]$: For every functional query f made by the adversary, the challenger generates C_E by executing $\mathsf{Sym.Enc}(\mathsf{Sym.K}^*, \mathsf{OneCT}.sk_f)$, with $\mathsf{OneCT}.sk_f$ being the output of $\mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; R)$, where R is picked at random. The rest of the hybrid is the same as the previous hybrid.

Claim 7. *Assuming the security of the pseudorandom function family $\mathcal{F}$, for every PPT adversary $\mathcal{A}$, for $b \in \{0,1\}$, $j \in [p]$, we have $|\mathsf{Adv}_{3.b,j}^{\mathcal{A}} - \mathsf{Adv}_{4.b,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.*

Proof. Suppose the claim is false for some PPT adversary $\mathcal{A}$, we construct a reduction that internally executes $\mathcal{A}$ and breaks the security of the pseudorandom function family $\mathcal{F}$. The reduction simulates the role of the challenger of the adaptive game when interacting with $\mathcal{A}$. The reduction answers the functional queries, made by the adversary as follows; the message queries are answered as in $\mathsf{Hybrid}_{3.b}^j$ (or $\mathsf{Hybrid}_{4.b}^j$). 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 $\mathsf{Sym.Enc}(\mathsf{Sym.K}^*, \mathsf{OneCT}.sk_f)$, where $\mathsf{OneCT}.sk_f = \mathsf{OneCT.KeyGen}(\mathsf{OneCT.SK}^*, f; R^*)$. The reduction then proceeds as in the previous hybrids to compute the functional key $\mathsf{Ad}.sk_f$ which it then sends to $\mathcal{A}$.

If the challenger of the PRF game sent $R^* = \mathsf{PRF}_{\mathsf{K}^*}(\tau)$ back to the reduction then we are in $\mathsf{Hybrid}_{3.b}^j$ else if R^* is generated at random, for every query τ, by the challenger then we are in $\mathsf{Hybrid}_{4.b}^j$. From our hypothesis this means that the probability that the reduction distinguishes the pseudorandom values from random values is non-negligible, contradicting the security of the pseudorandom function family $\mathcal{F}$. ■

We now show that $\mathsf{Hybrid}_{4.0}^j$ is computationally indistinguishable from $\mathsf{Hybrid}_{4.1}^j$.

Claim 8. *Assuming the adaptive security of OneCT, for $j \in [p]$, for every PPT adversary $\mathcal{A}$ we have $|\mathsf{Adv}_{4.0,j}^{\mathcal{A}} - \mathsf{Adv}_{4.1,j}^{\mathcal{A}}| \leq \mathsf{negl}(\lambda)$.*

Proof. Suppose there exists a PPT adversary $\mathcal{A}$, such that the claim is false. We design a reduction that internally executes $\mathcal{A}$ to break the adaptive security of OneCT.

The reduction simulates the role of the challenger of the adaptive private-key FE game. It answers both the functional as well as message queries made by the adversary as follows. If $\mathcal{A}$ makes a functional query f then it forwards it to the challenger of the adaptively-secure single-ciphertext FE scheme. In return it receives $\mathsf{OneCT}.sk_f$. It then encrypts it using the symmetric encryption scheme, where the symmetric key is picked by the reduction itself, and denote the resulting ciphertext to be C_E. The reduction then constructs the circuit $G_{f,C_E,\tau}$ as in the previous hybrids. Finally, the reduction computes the selective private-key functional key of $G_{f,C_E,\tau}$, where the reduction itself picks the master secret key of selective private-key FE scheme. The resulting functional key is then sent to $\mathcal{A}$. The message queries are handled as follows. Suppose the adversary $\mathcal{A}$ makes the i^{th} message-pair query (m_0^i, m_1^i). If $i \neq j$, then the reduction answers the query himself. That is, $\mathcal{B}$ samples the single-ciphertext FE master key $\mathsf{OneCT.SK}_i$ and PRF key K_i by himself. It then computes a single-ciphertext FE encryption of m_{b_i} using $\mathsf{OneCT.SK}_i$ and denote the result by CT_0^i: we define $b_i = 1$ if $i < j$ and $b_i = 0$ if $i > j$. Further, it computes a (selective) private-key FE encryption of $(\mathsf{OneCT.SK}_i, \mathsf{K}_i, 0^\lambda, 0)$, which is represented by CT_1^i. The challenger sends the ciphertext $\mathsf{CT}_i = (\mathsf{CT}_0^i, \mathsf{CT}_1^i)$ to $\mathcal{A}$. When $i = j$, the reduction forwards the message pair (m_0^j, m_1^j) to the challenger of the adaptive game. In response it receives CT_0^*. The reduction then generates CT_1^* on its own where CT_1^* is the selective FE encryption of $(0^\lambda, 0^\lambda, \mathsf{Sym.K}^*, 1)$. The reduction then sends $\mathsf{CT}^* = (\mathsf{CT}_0^*, \mathsf{CT}_1^*)$ to $\mathcal{A}$. The output of the reduction is the output of $\mathcal{A}$.

We note that the reduction is a legal adversary in the adaptive game of OneCT, i.e., for the message query (m_0^j, m_1^j), functional query f, we have that $f(m_0^j) = f(m_1^j)$: this follows from the fact that (i) the functional queries (resp., challenge message query) made by the adversary (of Ad) is the same as the functional queries (resp., challenge message query) made by the reduction, and (ii) the adversary (of Ad) is a legal adversary. This proves that the reduction is a legal adversary in the adaptive game.

If the challenger sends an encryption of m_0 then we are in $\mathsf{Hybrid}_{4.0}^{j}$ and if the challenger sends an encryption of m_1 then we are in $\mathsf{Hybrid}_{4.1}^{j}$. From our hypothesis, this means that the reduction breaks the security of OneCT. This proves the claim. ■

Hybrid_5: This corresponds to the real experiment when the challenger uses the encryption oracle, parameterized by bit 1, to generate the challenge ciphertexts. That is, for all message queries of the form (m_0, m_1), the challenger sends an encryption of m_1 to the adversary. The output of this hybrid is the same as the output of the adversary. We note that this hybrid is identical to the hybrid $\mathsf{Hybrid}_{1.1}^{p}$.

The above claims imply that Hybrid_0 is computationally indistinguishable from Hybrid_5 which proves the adaptive security of Ad. This completes the proof of theorem. ■