Functional Encryption for Randomized Functionalities in the Private-Key Setting from Minimal Assumptions

Ilan Komargodski[1]*, Gil Segev[2]**, and Eylon Yogev[1]*

[1] Weizmann Institute of Science, Rehovot 76100, Israel.
{ilan.komargodski,eylon.yogev}@weizmann.ac.il
[2] Hebrew University of Jerusalem, Jerusalem 91904, Israel.
segev@cs.huji.ac.il

Abstract. We present a construction of a private-key functional encryption scheme for any family of randomized functionalities based on *any such scheme for deterministic functionalities* that is sufficiently expressive. Instantiating our construction with existing schemes for deterministic functionalities, we obtain schemes for any family of randomized functionalities based on a variety of assumptions (including the LWE assumption, simple assumptions on multilinear maps, and even the existence of any one-way function) offering various trade-offs between security and efficiency.

Previously, Goyal, Jain, Koppula and Sahai [TCC, 2015] constructed a public-key functional encryption scheme for any family of randomized functionalities based on indistinguishability obfuscation.

One of the key insights underlying our work is that, in the private-key setting, a sufficiently expressive functional encryption scheme may be appropriately utilized for implementing proof techniques that were so far implemented based on obfuscation assumptions (such as the punctured programming technique of Sahai and Waters [STOC, 2014]). We view this as a contribution of independent interest that may be found useful in other settings as well.

* Research supported in part by a grant from the Israel Science Foundation, the I-CORE Program of the Planning and Budgeting Committee, BSF and the Israeli Ministry of Science and Technology.

** 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).

1 Introduction

The cryptographic community's vision of functional encryption [28, 11, 27] is rapidly evolving. Whereas traditional encryption schemes offer an all-or-nothing guarantee when accessing encrypted data, functional encryption schemes offer tremendous flexibility. Specifically, such schemes support restricted decryption keys that allow users to learn specific functions of the encrypted data and nothing else.

Motivated by the early examples of functional encryption schemes for specific functionalities (such as identity-based encryption [30, 8, 16]), extensive research has recently been devoted to the construction of functional encryption schemes for rich and expressive families of functions (see, for example, [28, 11, 27, 23, 2, 7, 13, 17, 18, 22, 32, 19, 15, 5] and the references therein).

Until very recently, research on functional encryption has focused on the case of *deterministic* functions. More specifically, in a functional encryption scheme for a family $\mathcal{F}$ of deterministic functions, a trusted authority holds a master secret key msk that enables to generate a functional key sk_f for any function $f \in \mathcal{F}$. Now, anyone holding the functional key sk_f and an encryption of some value x, can compute $f(x)$ but cannot learn any additional information about x. In many scenarios, however, dealing only with deterministic functions may be insufficient, and a more general framework allowing *randomized* functions is required.

Functional encryption for randomized functionalities. Motivated by various real-world scenarios, Goyal et al. [24] have recently put forward a generalization of functional encryption to randomized functionalities. In this setting, given a functional key sk_f for a randomized function f and given an encryption of a value x, one should be able to obtain a sample from the distribution $f(x)$. As Goyal et al. pointed out, the case of randomized functions presents new challenges for functional encryption. These challenge arise already when formalizing the security of functional encryption for randomized functions[3], and then become even more noticeable when designing such schemes.

Goyal et al. [24] presented a realistic framework for modeling the security of functional encryption schemes for randomized functionalities. Even more importantly, within their framework they constructed a public-key functional encryption scheme supporting the set of all randomized functionalities (that are computable by bounded-size circuits). Their construction builds upon the elegant approach of punctured programming due to Sahai and Waters [29], and they prove the security of their construction based on indistinguishability obfuscation [6, 18].

[3] For example, an adversary holding a functional key sk_f and an encryption of a value x, should not be able to tamper with the randomness that is used for sampling from distribution $f(x)$. This is extremely well motivated by the examples provided by Goyal et al. in the contexts of auditing an encrypted database via *randomized* sampling, and of performing differentially-private analysis on an encrypted database via *randomized* perturbations. We refer the reader to [24] for more details.

Identifying the minimal assumptions for functional encryption. The work of Goyal et al. [24] naturally gives rise to the intriguing question of whether functional encryption for randomized functionalities can be based on assumptions that are seemingly weaker than indistinguishability obfuscation. On one hand, it may be the case that functional encryption for randomized functionalities is indeed a significantly more challenging primitive than functional encryption for deterministic functionalities. In this case, it would be conceivable to use the full power of indistinguishability obfuscation for constructing such schemes. On the other hand, however, it may be possible that a functional encryption scheme for randomized functions can be constructed in a direct black-box manner from any such scheme for deterministic functions.

This question is especially interesting since various functional encryption schemes for (general) deterministic functionalities are already known to exist based on assumptions that seem significantly weaker than indistinguishability obfuscation (such as Learning with Errors assumption or even the existence of any one-way function) offering various trade-offs between security and efficiency (see Section 2.2 for more details on the existing schemes).

1.1 Our Contributions

In this work we consider functional encryption in the private-key setting, where the master secret key is used both for generating functional keys and for encryption. In this setting we provide an answer to the above question: we present a construction of a private-key functional encryption scheme for any family $\mathcal{F}$ of *randomized* functions based on *any* private-key functional encryption scheme for *deterministic* functions that is sufficiently expressive[4]. Inspired by the work of Goyal et al. [24] in the public-key setting, we prove the security of our construction within a similarly well-motivated framework for capturing the security of private-key functional encryption for randomized functions.

Instantiations. Our resulting scheme inherits the flavor of security guaranteed by the underlying scheme (e.g., full vs. selective security, and one-key vs. many-keys security), and can be instantiated by a variety of existing functional encryption schemes. Specifically, our scheme can be based either on the Learning with Errors assumption, on obfuscation assumptions, on multilinear-maps assumptions, or even on the existence of any one-way function (offering various trade-offs between security and efficiency – we refer the reader to Section 2.2 for more details on the possible instantiations).

Applicable scenarios. Following-up on the motivating applications given by Goyal et al. [24] in the contexts of auditing an encrypted database via *randomized* sampling, and of performing differentially-private analysis on an encrypted database via *randomized* perturbations, we observe that these two examples are

[4] Our only assumption on the underlying scheme is that it supports the family $\mathcal{F}$ (when viewed as a family of single-input deterministic functions), supports the evaluation procedure of a pseudorandom function family, and supports a few additional basic operations (such as conditional statements).

clearly valid in the private-key setting as well. Specifically, in both applications, the party that provides functional keys is more than likely the same one who encrypts the data.

Obfuscation-based techniques via function privacy. One of the key insights underlying our work is that in the private-key setting, where encryption is performed honestly by the owner of the master secret key, the power of indistinguishability obfuscation may not be needed. Specifically, we observe that in some cases one can instead rely on the weaker notion of *function privacy* [31, 9, 1, 15]. Intuitively, a functional encryption scheme is function private if a functional key sk_f for a function f reveals no "unnecessary" information on f. For functional encryption in the private-key setting, this essentially means that encryptions of messages $m_1, \ldots, m_T$ together with functional keys corresponding to functions $f_1, \ldots, f_T$ reveal essentially no information other than the values $\{f_i(m_j)\}_{i,j \in [T]}$. Brakerski and Segev [15] recently showed that a function-private scheme can be obtained from *any* private-key functional encryption scheme.

Building upon the notion of function privacy, we show that any *private-key* functional encryption scheme may be appropriately utilized for implementing some of the proof techniques that were so far implemented based on indistinguishability obfuscation. These include, in particular, a variant of the punctured programming approach of Sahai and Waters [29]. We view this as a contribution of independent interest that may be found useful in other settings as well.

1.2 Additional Related Work

A related generalization of functional encryption is that of functional encryption for *multiple-input* functions due to Goldwasser et al. [21]. A multiple-input functional encryption scheme for a function family $\mathcal{F}$ allows generating a functional key sk_f for any function $f \in \mathcal{F}$, and this enables to compute $f(x, y)$ given an encryption of x and an encryption of y, while not learning any additional information. Although capturing the security guarantees that can be provided by such schemes is quite challenging, multiple-input functional encryption might be useful for dealing with single-input randomized functionalities: One can view a randomized function $f(x; r)$ as a two-input function, where its first input is the actual input x, and its second input is the randomness r (that is possibly derived by a PRF key). However, the construction of Goldwasser et al. is based on indistinguishability obfuscation, and our goal is to rely on weaker assumptions. In addition, it is not clear that the notion of security of Goldwasser et al. suffices for capturing our notion of "best-possible" message privacy which allows for an a-priori non-negligible advantage in distinguishing the output distributions of two randomized functions (see Sections 1.3 and 3 for our notion of privacy).

Our construction relies on the notion of function privacy for functional encryption schemes, first introduced by Boneh et al. [9, 10] in the public-key setting, and then studied by Agrawal et al. [1] and by Brakerski and Segev [15] in the private-key setting (generalizing the work on predicate privacy in the private-key setting by Shen et al. [31]). As discussed in Section 1.1, for functional encryption in the private-key setting, function privacy essentially means that encryptions of messages $m_1, \ldots, m_T$ together with functional keys corresponding to functions $f_1, \ldots, f_T$ reveal essentially no information other than the values $\{f_i(m_j)\}_{i,j\in[T]}$. In terms of underlying assumptions, we rely on the fact that Brakerski and Segev [15] showed that a function-private scheme can be obtained from any private-key functional encryption scheme.

Lastly, Alwen et al. [3] studied the relationship between functional encryption and fully homomorphic encryption. In their work, they define public-key multi-input functional encryption schemes for randomized functionalities and construct such a scheme assuming a public-key multi-input function encryption scheme for deterministic functionalities. This result is incomparable to ours since *multi-input* functional encryption schemes are a much stronger assumptions (in particular, they imply indistinguishability obfuscation [21]), where our construction can be instantiated assuming seemingly weaker assumptions such as one-way function.

1.3 Overview of Our Approach

A private-key functional encryption scheme for a family $\mathcal{F}$ of randomized functions consists of four probabilistic polynomial-time algorithms $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$. The syntax is identical to that of functional encryption for deterministic functions (see Section 2.2), but the correctness and security requirements are more subtle. In this section we begin with a brief overview of our notions of correctness and security. Then, we provide a high-level overview of our new construction, and the main ideas and challenges underlying its proof of security.

Correctness and independence of decrypted values. Our notion of correctness follows that of Goyal et al. [24] by adapting it to the private-key setting. Specifically, we ask that for any sequence of messages $x_1, \ldots, x_T$ and for any sequence of functions $f_1, \ldots, f_T \in \mathcal{F}$, it holds that the distribution obtained by encrypting $x_1, \ldots, x_T$ and then decrypting the resulting ciphertexts with functional keys corresponding to $f_1, \ldots, f_T$ is computationally indistinguishable from the distribution $\{f_j(x_i; r_{i,j})\}_{i,j\in[T]}$ where the $r_{i,j}$'s are sampled independently and uniformly at random. As noted by Goyal et al. [24], unlike in the case of deterministic functions where is suffices to define correctness for a single ciphertext and a single key, here it is essential to define correctness for multiple (possibly correlated) ciphertexts and keys. We refer the reader to Section 3.1 for our formal definition.

"Best-possible" message privacy. As in functional encryption for deterministic functions, we consider adversaries whose goal is to distinguish between encryptions of two challenge messages, x_0^* and x_1^*, when given access to an encryption oracle (as required in private-key encryption) and to functional keys of various functions. Recall that in the case of deterministic functions, the adversary is allowed to ask for functional keys for any function f such that $f(x_0^*) = f(x_1^*)$.

When dealing with randomized functions, however, it is significantly less clear how to prevent adversaries from choosing functions f that will enable to easily distinguish between encryptions of x_0^* and x_1^*. Our notions of message privacy ask that the functional encryption scheme under consideration will not add a non-negligible advantage to the (*possibly non-negligible*) advantage that adversaries may already have in distinguishing between the distributions $f(x_0^*)$ and $f(x_1^*)$. That is, given that adversaries are able to obtain a sample from the distribution $f(x_0^*)$ or from the distribution $f(x_1^*)$ using the functional key sk_f, and may already have some advantage in distinguishing these distributions, we ask for "best-possible" message privacy in the sense that essentially *no additional advantage can be gained.*

Concretely, if the distributions $f(x_0^*)$ and $f(x_1^*)$ can be efficiently distinguished with advantage at most $\Delta = \Delta(\lambda)$ to begin with (*where Δ does not necessarily have to be negligible*), then we require that no adversary that is given a functional key for f will be able to distinguish between encryptions of x_0^* and x_1^* with advantage larger than $\Delta + \mathsf{neg}(\lambda)$, for some negligible function $\mathsf{neg}(\cdot)$. More generally, an adversary that is given functional keys for $T = T(\lambda)$ such functions (and access to an encryption oracle), should not be able to distinguish between encryptions of x_0^* and x_1^* with advantage larger than $T \cdot \Delta + \mathsf{neg}(\lambda)$. We note that our approach for realistically capturing message privacy somewhat differs from that of Goyal et al. [24], and we refer the reader to the full version [26] for a brief comparison between the two approaches[5].

We put forward two flavors of "best-possible" message privacy, a non-adaptive flavor and an adaptive flavor, depending on the flavor of indistinguishability guarantee that is satisfied by the function family under consideration. Details follow.

Out first notion addresses function families $\mathcal{F}$ such that for a randomly sampled $f \leftarrow \mathcal{F}$, no efficient adversary given f can output x_0 and x_1 and distinguish the distributions $f(x_0)$ and $f(x_1)$ with probability larger than Δ (note again that Δ does not have to be negligible). One possible example for such a function family is a function that on input x samples a public-key pk for a public-key encryption scheme, and outputs pk together with a randomized encryption of x. Our second notion addresses function families $\mathcal{F}$ such that no efficient adversary can output $f \in \mathcal{F}$ together with two inputs x_0 and x_1, and distinguish the distributions $f(x_0)$ and $f(x_1)$ with probability larger than Δ. One possible example for such a function family is that of differentially private mechanisms, as discussed by Goyal et al. [24]. We refer the reader to Section 3.2 for more information and the formal definitions.

Our construction. Let $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ be any private-key functional encryption scheme that provides message privacy and function privacy[6]. Our new scheme is quite intuitive and is described as follows:

[5] We emphasize that we view the main contribution of our paper as basing the security of our scheme on any underlying functional encryption scheme (and avoiding obfuscation-related assumptions), and not as offering alternative notions of message privacy.

[6] As discussed above, function privacy can be assumed without loss of generality using the transformation of Brakerski and Segev [15].

- The setup and decryption algorithms are identical to those of the underlying scheme.
- The encryption algorithm on input a message x, samples a string s uniformly at random, and outputs an encryption $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{msk}, (x, \bot, s, \bot))$ of x and s together with two additional "empty slots" that will be used in the security proof.
- The key-generation algorithm on input a description of a randomized function f, samples a PRF key K, and outputs a functional key for the deterministic function $\mathsf{Left}_{f,K}$ defined as follows: On input (x_L, x_R, s, z) output $f(x_L; r)$ where $r = \mathsf{PRF}_K(s)$.

The correctness and independence of our scheme follow in a straightforward manner from the correctness of the underlying scheme and the assumption that PRF is pseudorandom. In fact, it suffices that PRF is *weakly* pseudorandom (i.e., computationally indistinguishable from a truly random function when evaluated on independent and uniformly sampled inputs).

As for the message privacy of the scheme, recall that we consider adversaries that can access an encryption oracle and a key-generation oracle, and should not be able to distinguish between an encryption $\mathsf{Enc}(\mathsf{msk}, (x_0^*, \bot, s^*, \bot))$ of x_0^* and an encryption $\mathsf{Enc}(\mathsf{msk}, (x_1^*, \bot, s^*, \bot))$ of x_1^* with advantage larger than $T \cdot \Delta + \mathsf{neg}(\lambda)$ (where T is the number of functional keys given to the adversary, and Δ is the a-priori distinguishing advantage for the functions under consideration as described above).

The first step in our proof of security is to replace the challenge ciphertext with a modified challenge ciphertext $\mathsf{Enc}(\mathsf{msk}, (x_0^*, x_1^*, s^*, \bot))$ that contains information on both challenge messages (this is made possible due to the message privacy of the underlying scheme). Next, denoting the adversary's key-generation queries by $f_1, \ldots, f_T$, our goal is to replace the functional keys $\mathsf{Left}_{f_1,K_1}, \ldots, \mathsf{Left}_{f_T,K_T}$ with the functional keys $\mathsf{Right}_{f_1,K_1}, \ldots, \mathsf{Right}_{f_T,K_T}$, where the function $\mathsf{Right}_{f,K}$ is defined as follows: On input (x_L, x_R, s, z) output $f(x_R; r)$ where $r = \mathsf{PRF}_K(s)$. At this point we note that, from the adversary's point of view, when providing only Left keys the modified challenge ciphertext is indistinguishable from an encryption of x_0^*, and when providing only Right keys the modified challenge ciphertext is indistinguishable from an encryption of x_1^*.

The most challenging part of the proof is in bounding the adversary's advantage in distinguishing the sequences of Left and Right keys, based on the function privacy and the message privacy of the underlying scheme. The basic idea is to switch the functional keys from Left to Right one by one, following different proof strategies for pre-challenge keys and for post-challenge keys[7].

When dealing with a pre-challenge key sk_f, the function f is already known when producing the challenge ciphertext. Therefore, we can use the message privacy of the underlying scheme and replace the (already-modified) challenge ciphertext with $\mathsf{Enc}(\mathsf{msk}, (x_0^*, x_1^*, s^*, z^*))$, where $z^* = f(x_0^*; r^*)$ and $r^* = \mathsf{PRF}_K(s^*)$.

[7] We use the term *pre-challenge* keys for all functional keys that are obtained before the challenge phase, and the term *post-challenge* keys for all functional keys that are obtained after the challenge phase.

Then, we use the function privacy of the underlying scheme, and replace the functional key $\mathsf{Left}_{f,K}$ with a functional key for the function $\mathsf{OutputZ}$ that simply outputs z whenever $s = s^*$. From this point on, we use the pseudorandomness of PRF and replace $r^* = \mathsf{PRF}_K(s^*)$ with a truly uniform r^*, and then replace $z^* \leftarrow f(x_0^*)$ with $z^* \leftarrow f(x_1^*)$. Similar steps then enable us to replace the functional key $\mathsf{OutputZ}$ with a functional key for the function $\mathsf{Right}_{f,K}$.

When dealing with a post-challenge key sk_f, we would like to follow the same approach of embedding the value $f(x_0^*; r^*)$ or $f(x_1^*; r^*)$. However, for post-challenge keys, the function f is not known when producing the challenge ciphertext. Instead, in this case, the challenge messages x_0^* and x_1^* are known when producing the functional key sk_f. Combining this with the function privacy of the underlying scheme enables us to embed the above values in the functional key sk_f, and once again replace the Left keys with the Right keys. We refer the reader to Section 4 for the formal description of our scheme and its proof of security.

1.4 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we provide an overview of the basic notation and standard tools underlying our construction. In Section 3 we introduce our notions of security for private-key functional encryption schemes for randomized functionalities. In Section 4 we present our new scheme and prove its security. Formal proofs of the claims that are stated in Section 4 can be found in the full version [26].

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 $f(x)$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. A function $\mathsf{neg} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(\lambda) < \lambda^{-c}$ for all $\lambda > N_c$.

The *statistical distance* between two random variables X and Y over a finite domain Ω is defined as $\mathsf{SD}(X, Y) = \frac{1}{2} \sum_{\omega \in \Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. Two sequences of random variables $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *computationally indistinguishable* if for any probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\left|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]\right| \leq \mathsf{neg}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$.

2.1 Pseudorandom Functions

Let $\{\mathcal{K}_\lambda, \mathcal{X}_\lambda, \mathcal{Y}_\lambda\}_{\lambda\in\mathbb{N}}$ be a sequence of sets and let $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ be a function family with the following syntax:

- $\mathsf{PRF.Gen}$ is a probabilistic polynomial-time algorithm that takes as input the unary representation of the security parameter λ, and outputs a key $K \in \mathcal{K}_\lambda$.
- $\mathsf{PRF.Eval}$ is a deterministic polynomial-time algorithm that takes as input a key $K \in \mathcal{K}_\lambda$ and a value $x \in \mathcal{X}_\lambda$, and outputs a value $y \in \mathcal{Y}_\lambda$.

The sets $\mathcal{K}_\lambda$, $\mathcal{X}_\lambda$, and $\mathcal{Y}_\lambda$ are referred to as the *key space*, *domain*, and *range* of the function family, respectively. For easy of notation we may denote by $\mathsf{PRF.Eval}_K(\cdot)$ or $\mathsf{PRF}_K(\cdot)$ the function $\mathsf{PRF.Eval}(K, \cdot)$ for $K \in \mathcal{K}_\lambda$. The following is the standard definition of a pseudorandom function family.

Definition 1 (Pseudorandomness). *A function family* $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ *is* pseudorandom *if for every probabilistic polynomial-time algorithm* $\mathcal{A}$ *there exits a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\mathsf{Adv}_{\mathsf{PRF},\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \left| \Pr_{K \leftarrow \mathsf{PRF.Gen}(1^\lambda)}\left[\mathcal{A}^{\mathsf{PRF.Eval}_K(\cdot)}(1^\lambda) = 1\right] - \Pr_{f \leftarrow F_\lambda}\left[\mathcal{A}^{f(\cdot)}(1^\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$*, where* F_λ *is the set of functions that map* $\mathcal{X}_\lambda$ *into* $\mathcal{Y}_\lambda$*.*

In addition to the standard notion of a pseudorandom function family, we rely on the seemingly stronger (yet existentially equivalent) notion of a *puncturable* pseudorandom function family [25, 12, 29, 14]. In terms of syntax, this notion asks for an additional probabilistic polynomial-time algorithm, $\mathsf{PRF.Punc}$, that takes as input a key $K \in \mathcal{K}_\lambda$ and a set $S \subseteq \mathcal{X}_\lambda$ and outputs a "punctured" key K_S. The properties required by such a puncturing algorithm are capture by the following definition.

Definition 2 (Puncturable PRF). *A pseudorandom function family* $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval}, \mathsf{PRF.Punc})$ *is* puncturable *if the following properties are satisfied:*

1. ***Functionality:*** *For all sufficiently large* $\lambda \in \mathbb{N}$*, for every set* $S \subseteq \mathcal{X}_\lambda$*, and for every* $x \in \mathcal{X}_\lambda \setminus S$ *it holds that*

$$\Pr_{\substack{K \leftarrow \mathsf{PRF.Gen}(1^\lambda);\\ K_S \leftarrow \mathsf{PRF.Punc}(K,S)}}[\mathsf{PRF.Eval}_K(x) = \mathsf{PRF.Eval}_{K_S}(x)] = 1.$$

2. ***Pseudorandomness at Punctured Points:*** *Let* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *be any probabilistic polyomial-time algorithm such that* $\mathcal{A}_1(1^\lambda)$ *outputs a set* $S \subseteq \mathcal{X}_\lambda$*, a value* $x \in S$*, and state information* state*. Then, for any such* $\mathcal{A}$ *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\mathsf{Adv}_{\mathsf{puPRF},\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \left|\Pr\left[\mathcal{A}_2(K_S, \mathsf{PRF.Eval}_K(x), \mathsf{state}) = 1\right] - \Pr\left[\mathcal{A}_2(K_S, y, \mathsf{state}) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where $(S, x, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$, $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$, $K_S = \mathsf{PRF.Punc}(K, S)$, and $y \leftarrow \mathcal{Y}_\lambda$.

As observed by [25, 12, 29, 14] the GGM construction [20] of PRFs from one-way functions can be easily altered to yield a puncturable PRF.

2.2 Private-Key Functional Encryption

A private-key functional encryption scheme over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ is a quadruple $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms. The setup algorithm Setup takes as input the unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$ and outputs a master-secret key msk. The key-generation algorithm KG takes as input a master-secret key msk and a function $f \in \mathcal{F}_\lambda$, and outputs a functional key sk_f. The encryption algorithm Enc takes as input a master-secret key msk and a message $x \in \mathcal{X}_\lambda$, and outputs a ciphertext ct. In terms of correctness we require that for all sufficiently large $\lambda \in \mathbb{N}$, for every function $f \in \mathcal{F}_\lambda$ and message $x \in \mathcal{X}_\lambda$ it holds that $\mathsf{Dec}(\mathsf{KG}(\mathsf{msk}, f), \mathsf{Enc}(\mathsf{msk}, x)) = f(x)$ with all but a negligible probability over the internal randomness of the algorithms Setup, KG, and Enc.

In terms of security, we rely on the private-key variants existing indistinguishability based notions for message privacy (see, for example, [11, 27, 7]) and function privacy (see [1, 15]). When formalizing these notions it would be convenient to use the following standard notion of a *left-or-right oracle*.

Definition 3 (Left-or-right oracle). *Let $\mathcal{O}(\cdot, \cdot)$ be a probabilistic two-input functionality. For each $b \in \{0, 1\}$ we denote by $\mathcal{O}_b$ the probabilistic three-input functionality $\mathcal{O}_b(k, z_0, z_1) \stackrel{\mathsf{def}}{=} \mathcal{O}(k, z_b)$.*

Message Privacy

A functional encryption scheme is message private if the encryptions of any two messages x_0 and x_1 are computationally indistinguishable given access to an encryption oracle (as required in private-key encryption) and to functional keys for any function f such that $f(x_0^*) = f(x_1^*)$. We consider two variants of message privacy: (*full*) message privacy in which adversaries are fully adaptive, and *selective-function* message privacy in which adversaries must issue their key-generation queries in advance.

Definition 4 (Message privacy). *A functional encryption scheme* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ is* message private *if for any probabilistic polynomial-time adversary $\mathcal{A}$ there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\begin{aligned}
&\mathsf{Adv}^{\mathsf{MP}}_{\mathsf{FE},\mathcal{A},\mathcal{F}}(\lambda) \stackrel{\mathsf{def}}{=} \\
&\left|\Pr\left[\mathcal{A}^{\mathsf{KG}(\mathsf{msk},\cdot),\mathsf{Enc}_0(\mathsf{msk},\cdot,\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{KG}(\mathsf{msk},\cdot),\mathsf{Enc}_1(\mathsf{msk},\cdot,\cdot)}(1^\lambda) = 1\right]\right| \\
&\leq \mathsf{neg}(\lambda)
\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for every $(x_0, x_1) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$ and $f \in \mathcal{F}_\lambda$ with which $\mathcal{A}$ queries the oracles Enc_b and KG, respectively, it holds that $f(x_0) = f(x_1)$. Moreover, the probability is taken over the choice of $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ and the internal randomness of $\mathcal{A}$.

Definition 5 (Selective-function message privacy). *A functional encryption scheme $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is* T-selective-function message private*, where $T = T(\lambda)$, if for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{sfMP}}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) \overset{\mathsf{def}}{=} \\ &\left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$, where for each $b \in \{0, 1\}$ and $\lambda \in \mathbb{N}$ the random variable $\mathsf{Expt}^{(b)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda)$ is defined as follows:

1. $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $(f_1, \ldots, f_T, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$, *where $f_i \in \mathcal{F}_\lambda$ for all $i \in [T]$.*
3. $\mathsf{sk}_{f_i} \leftarrow \mathsf{KG}(\mathsf{msk}, f_i)$ *for all $i \in [T]$.*
4. $b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}_b(\mathsf{msk},\cdot,\cdot)}(\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_T}, \mathsf{state})$, *where for each of $\mathcal{A}_2$'s queries $(x_0, x_1) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$ to $\mathsf{Enc}_b(\mathsf{msk}, \cdot, \cdot)$ it holds that $f_i(x_0) = f_i(x_1)$ for all $i \in [T]$.*
5. *Output b'.*

Such a scheme is selective-function message private *if it is T-selective-function message private for all polynomials $T = T(\lambda)$.*

Known constructions. Private-key functional encryption schemes that satisfy the notions presented in Definitions 4 and 5 (and support circuits of any a-priori bounded polynomial size) are known to exist based on various assumptions. The known schemes are in fact public-key schemes, which are in particular private-key ones.

Specifically, a public-key scheme that satisfies the notion of 1-selective function message privacy was constructed by Gorbunov, Vaikuntanathan and Wee [23] under the sole assumption that public-key encryption exists. In the private-key setting, their transformation can in fact rely on any private-key encryption scheme (and thus on any one-way function). By assuming, in addition, the existence of a pseudorandom generator computable by small-depth circuits (which is known to be implied by most concrete intractability assumptions), they construct a scheme that satisfies the notion of T-selective-function message privacy for any predetermined polynomial $T = T(\lambda)$. However, the length of the ciphertexts in their scheme grows linearly with T and with an upper bound on the circuit size of the functions that the scheme allows (which also has to be known ahead of time). Goldwasser et al. [22] showed that based on the Learning with Errors (LWE) assumption, T-selective-function message privacy can be achieved where the ciphertext size grows with T and with a bound on the depth of allowed functions.

In addition, schemes that satisfy the notion of (full) message privacy (Definition 4) were constructed by Boyle et al. [13] and by Ananth et al. [4] based on differing-input obfuscation, by Waters [32] based on indistinguishability obfuscation, and by Garg et al. [19] based on multilinear maps. Very recently, Ananth et al. [5] gave a generic transformation from selective-message message privacy to full message privacy. We conclude that there is a variety of constructions offering various flavors of security under various assumptions that can be used as a building block in our construction.

Function Privacy

A private-key functional-encryption scheme is function private [31, 1, 15] if a functional key sk_f for a function f reveals no "unnecessary" information on f. More generally, we ask that encryptions of messages $m_1, \ldots, m_T$ together with functional keys corresponding to functions $f_1, \ldots, f_T$ reveal essentially no information other than the values $\{f_i(m_j)\}_{i,j \in [T]}$. We consider two variants of function privacy: (*full*) function privacy in which adversaries are fully adaptive, and *selective-function* function privacy in which adversaries must issue their key-generation queries in advance.

Definition 6 (Function privacy). *A functional encryption scheme* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *over a message space* $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ *and a function space* $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ *is* function private *if for any probabilistic polynomial-time adversary* $\mathcal{A}$ *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{FP}}_{\mathsf{FE},\mathcal{A},\mathcal{F}}(\lambda) \stackrel{\mathsf{def}}{=} \\ &\left|\Pr\left[\mathcal{A}^{\mathsf{KG}_0(\mathsf{msk},\cdot,\cdot),\mathsf{Enc}_0(\mathsf{msk},\cdot,\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{\mathsf{KG}_1(\mathsf{msk},\cdot,\cdot),\mathsf{Enc}_1(\mathsf{msk},\cdot,\cdot)}(1^\lambda) = 1\right]\right| \\ &\leq \mathsf{neg}(\lambda)\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$*, where for every* $(f_0, f_1) \in \mathcal{F}_\lambda \times \mathcal{F}_\lambda$ *and* $(x_0, x_1) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$ *with which* $\mathcal{A}$ *queries the oracles* KG_b *and* Enc_b*, respectively, it holds that* $f_0(x_0) = f_1(x_1)$*. Moreover, the probability is taken over the choice of* $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$ *and the internal randomness of* $\mathcal{A}$*.*

Definition 7 (Selective-function function privacy). *A functional encryption scheme* $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *over a message space* $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ *and a function space* $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ *is said* T-selective-function function private*, where* $T = T(\lambda)$*, if for any probabilistic polynomial-time adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{sfFP}}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) \stackrel{\mathsf{def}}{=} \\ &\left|\Pr\left[\mathsf{Expt}^{(0)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{(1)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda),\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$*, where for each* $b \in \{0,1\}$ *and* $\lambda \in \mathbb{N}$ *the random variable* $\mathsf{Expt}^{(b)}_{\mathsf{FE},\mathcal{A},\mathcal{F},T}(\lambda)$ *is defined as follows:*

1. $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $((f_{0,1},\ldots,f_{0,T}),(f_{1,1},\ldots,f_{1,T}),\mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$*, where* $f_{\sigma,i} \in \mathcal{F}_\lambda$ *for all* $\sigma \in \{0,1\}$ *and* $i \in [T]$.
3. $\mathsf{sk}^*_i \leftarrow \mathsf{KG}(\mathsf{msk}, f_{b,i})$ *for all* $i \in [T]$.
4. $b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}_b(\mathsf{msk},\cdot,\cdot)}(\mathsf{sk}^*_1,\ldots,\mathsf{sk}^*_T,\mathsf{state})$*, where for each query* $(x_0,x_1) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$ *to* $\mathsf{Enc}_b(\mathsf{msk},\cdot,\cdot)$ *it holds that* $f_{0,i}(x_0) = f_{1,i}(x_1)$ *for all* $i \in [T]$.
5. *Output* b'.

Such a scheme is selective-function function private *if it is* T*-selective-function function private for all polynomials* $T = T(\lambda)$.

Known constructions. Brakerski and Segev [15] showed how to transform any (selective-function or fully secure) message-private functional encryption scheme into a (selective-function or fully secure, respectively) functional encryption scheme which is also function private. Thus, any instantiation of a message-private (or selective-function message private) function encryption scheme as discussed in Section 2.2 can be used as a building block in our construction.

3 Private-Key Functional Encryption for Randomized Functionalities

In this section we present a framework for capturing the security of private-key functional encryption for randomized functionalities. Our framework is inspired by that of Goyal et al. [24] in the public-key setting, but takes a slightly different approach as we discuss below.

Throughout this section, we let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a family of randomized functionalities, where for every $\lambda \in \mathbb{N}$ the set $\mathcal{F}_\lambda$ consists of functions of the form $f : \mathcal{X}_\lambda \times \mathcal{R}_\lambda \to \mathcal{Y}_\lambda$. That is, such a function f maps $\mathcal{X}_\lambda$ into $\mathcal{Y}_\lambda$ using randomness from $\mathcal{R}_\lambda$.

A private-key functional encryption scheme for a family $\mathcal{F}$ of randomized functions consists of four probabilistic polynomial-time algorithms (Setup, KG, Enc, Dec) with the same syntax that is described in Section 2.2 for deterministic functions. Although the syntax in this setting is the same as in the deterministic setting, the correctness and security requirements are more subtle.

3.1 Correctness and Independence

In terms of correctness we rely on the definition of Goyal et al. [24] (when adapted to the private-key setting). As discussed in Section 1.3, we ask that for any sequence of messages $x_1,\ldots,x_T$ and for any sequence of functions $f_1,\ldots,f_T \in \mathcal{F}$, it holds that the distribution obtained by encrypting $x_1,\ldots,x_T$ and then decrypting the resulting ciphertexts with functional keys corresponding to $f_1,\ldots,f_T$

is computationally indistinguishable from the distribution $\{f_j(x_i; r_{i,j})\}_{i,j\in[T]}$ where the $r_{i,j}$'s are sampled independently and uniformly at random.

Definition 8 (Correctness). *A functional encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *for a family* $\mathcal{F}$ *of randomized functions is* correct *if for all sufficiently large* $\lambda \in \mathbb{N}$*, for every polynomial* $T = T(\lambda)$*, and for every* $x_1, \ldots, x_T \in \mathcal{X}_\lambda$ *and* $f_1, \ldots, f_T \in \mathcal{F}_\lambda$*, the following two distributions are computationally indistinguishable:*

- $\mathbf{Real}(\boldsymbol{\lambda}) \overset{\mathsf{def}}{=} \left\{\mathsf{Dec}(\mathsf{sk}_{f_j}, \mathsf{ct}_i)\right\}_{i,j\in[T]}$, *where:*
  - $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$,
  - $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{msk}, x_i)$ *for all* $i \in [T]$,
  - $\mathsf{sk}_{f_j} \leftarrow \mathsf{KG}(\mathsf{msk}, f_j)$ *for all* $j \in [T]$.
- $\mathbf{Ideal}(\boldsymbol{\lambda}) \overset{\mathsf{def}}{=} \{f_j(x_i)\}_{i,j\in[T]}$.

As noted by Goyal et al. [24], unlike in the case of deterministic functions where is suffices to define correctness for a single ciphertext and a single key, here it is essential to define correctness for multiple (possibly correlated) ciphertexts and keys. We refer the reader to [24] for more details.

3.2 "Best-Possible" Message Privacy

We consider indistinguishability-based notions for capturing message privacy in private-key functional encryption for randomized functionalities. As in the (standard) case of deterministic functions (see Section 2.2), we consider adversaries whose goal is to distinguish between encryptions of two challenge messages x_0^* and x_1^*, when given access to an encryption oracle (as required in private-key encryption) and to functional keys of various functions. Recall that in the case of deterministic functions, the adversary is allowed to ask for functional keys for any function f such that $f(x_0^*) = f(x_1^*)$.

As discussed in Section 1.3, our notions of message privacy ask that the functional encryption scheme under consideration will not add any non-negligible advantage to the (*possibly non-negligible*) advantage that adversaries holding a functional key for a function f may already have in distinguishing between the distributions $f(x_0^*)$ and $f(x_1^*)$ to begin with. That is, given that adversaries are able to obtain a sample from the distribution $f(x_0^*)$ or from the distribution $f(x_1^*)$ using the functional key sk_f, and may already have some advantage in distinguishing these distributions, we ask for "best-possible" message privacy in the sense that essentially no additional advantage can be gained.

In what follows we put forward two flavors of "best-possible" message privacy, depending on the flavor of indistinguishability guarantee that is satisfied by the function family under consideration.

Message privacy for non-adaptively-admissible functionalities. Our first notion is that of *non-adaptively-admissible* function families. These are families $\mathcal{F}$ such that for a randomly sampled $f \leftarrow \mathcal{F}$, no efficient adversary on input f

can output x_0 and x_1 and distinguish the distributions $f(x_0)$ and $f(x_1)$ with probability larger than Δ (note again that Δ does not have to be negligible). One possible example for such a function family is a function that on input x samples a public-key pk for a public-key encryption scheme, and outputs pk together with a randomized encryption of x.

For such function families we consider a corresponding notion of message privacy in which the adversary obtains functional keys only for functions that are sampled uniformly and independently from $\mathcal{F}$. This is formally captured by the following two definitions.

Definition 9 (Non-adaptively-admissible function family). *A family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ of efficiently-computable randomized functions is* $\Delta(\lambda)$-non-adaptively admissible *if for any probabilistic polynomial-time algorithm $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ it holds that*

$$\mathsf{Adv}^{\mathsf{naADM}}_{\mathcal{F},\mathcal{A}}(\lambda) \overset{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{naADM}}_{\mathcal{F},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq \Delta(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the random variable $\mathsf{Expt}^{\mathsf{naADM}}_{\mathcal{F},\mathcal{A}}(\lambda)$ is defined via the following experiment:

1. $b \leftarrow \{0,1\}$, $f \leftarrow \mathcal{F}_\lambda$.
2. $(x_0, x_1, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda, f)$.
3. $y = f(x_b; r)$ *for* $r \leftarrow \{0,1\}^*$.
4. $b' \leftarrow \mathcal{A}_2(y, \mathsf{state})$.
5. *If* $b' = b$ *then output* 1, *and otherwise output* 0.

Definition 10 (Message privacy; non-adaptive case). *Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a $\Delta(\lambda)$-non-adaptively admissible function family. A private-key functional encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is* message private with respect to $\mathcal{F}$ *if for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and for any polynomial $T = T(\lambda)$ there exists a negligible function $\mathsf{neg}(\lambda)$ such that*

$$\mathsf{Adv}^{\mathsf{naMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda) \overset{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{naMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda) = 1\right] - \frac{1}{2} \right| \leq T(\lambda)\cdot\Delta(\lambda) + \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the random variable $\mathsf{Expt}^{\mathsf{naMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda)$ is defined via the following experiment:

1. $b \leftarrow \{0,1\}$, $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$, $f_1, \ldots, f_T \leftarrow \mathcal{F}_\lambda$.
2. $\mathsf{sk}_{f_i} \leftarrow \mathsf{KG}(\mathsf{msk}, f_i)$ *for all* $i \in [T]$.
3. $(x_0^*, x_1^*, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(\mathsf{msk},\cdot)}(1^\lambda, f_1, \ldots, f_T, \mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_T})$.
4. $c^* = \mathsf{Enc}(\mathsf{msk}, x_b^*)$.
5. $b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}(\mathsf{msk},\cdot)}(c^*, \mathsf{state})$.
6. *If* $b' = b$ *then output* 1, *and otherwise output* 0.

Message privacy for adaptively-admissible functionalities. Our second notion is that of *adaptively-admissible* function families. These are families $\mathcal{F}$ such that no efficient adversary can output $f \in \mathcal{F}$ together with two inputs x_0 and x_1, and distinguish the distributions $f(x_0)$ and $f(x_1)$ with probability larger than Δ. One possible example for such a function family is that of differentially private mechanisms, as discussed by Goyal et al. [24]. Specifically, these are randomized functions that on any two inputs that differ on only a few of their entries, produce output distributions whose *statistical* distance is polynomially small (i.e., Δ is polynomial in $1/\lambda$)[8].

It is easy to observe that there are function families that are non-adaptively admissible but are not adaptively admissible. One possible example is functions of the form f_{pk} that are indexed by a public encryption key pk, and on input x output a randomized encryption of x under pk. Giving adversaries the possibility of adaptively choosing such functions, they can choose a function f_{pk} for which they know the corresponding decryption key sk. In this case, although for a randomly chosen pk the distributions $f_{\mathsf{pk}}(x_0)$ and $f_{\mathsf{pk}}(x_1)$ are computationally indistinguishable, they may be easily distinguishable given the randomness used by the adversary (from which it may be easy to compute the corresponding decryption key sk).

For adaptively-admissible function families we consider a corresponding notion of message privacy in which the adversary obtains functional keys for functions that are adaptively chosen from $\mathcal{F}$. This is formally captured by the following two definitions.

Definition 11 (Adaptively-admissible function family). *A family* $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ *of efficiently-computable randomized functions is* $\Delta(\lambda)$-adaptively admissible *if for any probabilistic polynomial-time algorithm* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *it holds that*

$$\mathsf{Adv}^{\mathsf{aADM}}_{\mathcal{F},\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{aADM}}_{\mathcal{F},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \Delta(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$*, where the random variable* $\mathsf{Expt}^{\mathsf{aADM}}_{\mathcal{F},\mathcal{A}}(\lambda)$ *is defined via the following experiment:*

1. $b \leftarrow \{0,1\}$.
2. $(f, x_0, x_1, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$*, where* $f \in \mathcal{F}_\lambda$.
3. $y = f(x_b; r)$ *for* $r \leftarrow \{0,1\}^*$.
4. $b' \leftarrow \mathcal{A}_2(y, \mathsf{state})$.
5. *If* $b' = b$ *then output* 1*, and otherwise output* 0.

Definition 12 (Message privacy; adaptively-admissible case). *Let* $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ *be a* $\Delta(\lambda)$*-adaptively admissible function family. A private-key functional encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ *is* message private with respect to $\mathcal{F}$ *if for any probabilistic polynomial-time adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *that*

[8] The definitions of differential privacy are in fact stronger than requiring small statistical distance.

issues at most $T = T(\lambda)$ key-generation queries there exists a negligible function $\mathsf{neg}(\lambda)$ *such that*

$$\mathsf{Adv}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2} \right| \leq T(\lambda) \cdot \Delta(\lambda) + \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the random variable $\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A}}(\lambda)$ *is defined via the following experiment:*

1. $b \leftarrow \{0,1\}$, $\mathsf{msk} \leftarrow \mathsf{Setup}(1^\lambda)$.
2. $(x_0^*, x_1^*, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(\mathsf{msk},\cdot),\mathsf{KG}(\mathsf{msk},\cdot)}(1^\lambda)$.
3. $c^* = \mathsf{Enc}(\mathsf{msk}, x_b^*)$.
4. $b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}(\mathsf{msk},\cdot),\mathsf{KG}(\mathsf{msk},\cdot)}(c^*, \mathsf{state})$.
5. *If $b' = b$ then output 1, and otherwise output 0.*

4 Our Functional Encryption Scheme

In this section we present our construction of a private-key functional encryption scheme for randomized functionalities. Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ be a family of randomized functionalities, where for every $\lambda \in \mathbb{N}$ the set $\mathcal{F}_\lambda$ consists of functions of the form $f : \mathcal{X}_\lambda \times \mathcal{R}_\lambda \to \mathcal{Y}_\lambda$ (i.e., f maps $\mathcal{X}_\lambda$ into $\mathcal{Y}_\lambda$ using randomness from $\mathcal{R}_\lambda$). Our construction relies on the following building blocks:

1. A private-key functional encryption scheme $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KG}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$.
2. A pseudorandom function family $\mathsf{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$. We assume that for every $\lambda \in \mathbb{N}$ and for every key K that is produced by $\mathsf{PRF.Gen}(1^\lambda)$, it holds that $\mathsf{PRF.Eval}(K, \cdot) : \{0,1\}^\lambda \to \mathcal{R}_\lambda$.

As discussed in Section 1.1, we assume that the scheme FE is sufficiently expressive in the sense that it supports the function family $\mathcal{F}$ (when viewed as a family of single-input deterministic functions), the evaluation procedure of the pseudorandom function family PRF, and a few additional basic operations (such as conditional statements). Our scheme $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is defined as follows.

- **The setup algorithm.** On input the security parameter 1^λ the setup algorithm Setup samples $\mathsf{FE.msk} \leftarrow \mathsf{FE.Setup}(1^\lambda)$, and outputs $\mathsf{msk} = \mathsf{FE.msk}$.
- **The key-generation algorithm.** On input the master secret key msk and a function $f \in \mathcal{F}_\lambda$, the key-generation algorithm KG samples $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and outputs $\mathsf{sk}_f \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \mathsf{Left}_{f,K})$, where $\mathsf{Left}_{f,K}$ is a deterministic function that is defined in Figure 1.
- **The encryption algorithm.** On input the master secret key msk and a message $x \in \mathcal{X}_\lambda$, the encryption algorithm Enc samples $s \leftarrow \{0,1\}^\lambda$ and outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x, \perp, s, \perp))$.
- **The decryption algorithm.** On input a functional key sk_f and a ciphertext ct, the decryption algorithm Dec outputs $\mathsf{FE.Dec}(\mathsf{sk}_f, \mathsf{ct})$.

```
$\mathsf{Left}_{f,K}(x_L, x_R, s, z)$:
  1. Let $r = \mathsf{PRF.Eval}(K, s)$.
  2. Output $f(x_L; r)$.
```

```
$\mathsf{Right}_{f,K}(x_L, x_R, s, z)$:
  1. Let $r = \mathsf{PRF.Eval}(K, s)$.
  2. Output $f(x_R; r)$.
```

Figure 1: The functions $\mathsf{Left}_{f,K}$ and $\mathsf{Right}_{f,K}$. The function $\mathsf{Left}_{f,K}$ is used by the actual scheme, whereas the function $\mathsf{Right}_{f,K}$ is used in the proofs of its security.

The correctness and independence of the above scheme with respect to any family of randomized functionalities follows in a straightforward manner from the correctness of the underlying functional encryption scheme FE and the assumption that PRF is a pseudorandom function family (in fact, it suffices that PRF is a *weak* pseudorandom function family). Specifically, consider a sequence of messages $x_1, \ldots, x_T$ and a sequence of functions $f_1, \ldots, f_T$. As the encryption $\mathsf{FE.Enc}(\mathsf{msk}, (x_i, \perp, s_i, \perp))$ of each message x_i uses a uniformly sampled $s_i \in \{0,1\}^\lambda$, and the functional key for a function f_j contains a freshly sampled key K_j for the pseudorandom function family, the distribution $\{f_j(x_i; \mathsf{PRF.Eval}(K_j, s_i)\}$ is computationally indistinguishable from the distribution $\{f_j(x_i; r_{i,j})\}$, where the $r_{i,j}$'s are sampled independently and uniformly at random.

The following two theorems capture the security of the scheme. These theorems state that under suitable assumptions on the underlying building blocks, the scheme is message private for non-adaptively-admissible randomized functionalities and for adaptively-admissible randomized functionalities.

Theorem 1. *Assuming that* PRF *is a pseudorandom function family and that* FE *is selective-function function private, then Π is message private for non-adaptively-admissible randomized functionalities.*

Theorem 2. *Assuming that* PRF *is a* puncturable *pseudorandom function family and that* FE *is function private, then Π is message private for adaptively-admissible randomized functionalities.*

As discussed in Sections 2.1 and 2.2, Theorems 1 and 2 can be instantiated based on a variety of known pseudorandom function families and functional encryption schemes. In particular, Theorem 1 can be based on the minimal assumption that a selective-function message-private functional encryption scheme exists, and Theorem 2 can be based on the minimal assumption that a message-private functional encryption scheme exists.

Due to lack of space we omit the proof of Theorem 1 and include only the proof of Theorem 2. We refer to the full version of the paper [26] for the missing details.

4.1 Proof of Theorem 2

We prove that the scheme Π is message private for adaptively-admissible functionalities (see Definition 12) based on the assumptions that PRF is a puncturable pseudorandom function family and that FE is function private (see Definition 6).

Let $\mathcal{A}$ be a probabilistic polynomial-time adversary that issues at most $T_1 = T_1(\lambda)$ pre-challenge key-generation queries, at most $T_2 = T_2(\lambda)$ post-challenge key-generation queries (where $T = T_1 + T_2$), and at most $T = T(\lambda)$ encryption queries (note that T_1, T_2 and T may be any polynomials and are not fixed in advance), and let $\mathcal{F}$ be a Δ-adaptively admissible family of randomized functionalities. We denote by $f_1, \ldots, f_T$ the key-generation queries that are issued by $\mathcal{A}$.

We present a sequence of experiments and upper bound $\mathcal{A}$'s advantage in distinguishing each two consecutive experiments. Each two consecutive experiments differ either in the distribution of their challenge ciphertexts or in the distribution of the functional keys that are produced by the key-generation oracle. The first experiment is the experiment $\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda)$ (see Definition 12), and the last experiment is completely independent of the bit b. This enables us to prove that there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda) \stackrel{\mathsf{def}}{=} \left| \Pr\left[\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda) = 1\right] - \frac{1}{2} \right| \leq T(\lambda) \cdot \Delta(\lambda) + \mathsf{neg}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$. Throughout the proof we use, in addition to the functions $\mathsf{Left}_{f,K}$ and $\mathsf{Right}_{f,K}$ that were defined in Figure 1, the functions $\mathsf{PuncOutputY}_{f,K',y,s^*}$ and $\mathsf{PuncOutputZ}_{f,K',s^*}$ that are defined in Figure 2. In what follows we describe the experiments. We note that in all experiments the encryption oracle is as defined by the encryption procedure of the scheme.

$\mathsf{PuncOutputY}_{f,K',y,s^*}(x_L, x_R, s, z)$:

1. If $s = s^*$ then output y.
2. Otherwise, let $r = \mathsf{PRF.Eval}(K', s)$ and output $f(x_L; r)$.

$\mathsf{PuncOutputZ}_{f,K',s^*}(x_L, x_R, s, z)$:

1. If $s = s^*$ then output z.
2. Otherwise, let $r = \mathsf{PRF.Eval}(K', s)$ and output $f(x_L; r)$.

Figure 2: The functions $\mathsf{PuncOutputY}_{f,K',y,s^*}$ and $\mathsf{PuncOutputZ}_{f,K',s^*}$.

Experiment $\mathcal{H}^{(0)}(\lambda)$. This is the experiment $\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A}}(\lambda)$ (see Definition 12).

Experiment $\mathcal{H}^{(1)}(\lambda)$. This experiment is obtained from the experiment $\mathcal{H}^{(0)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it samples $s^* \leftarrow \{0,1\}^\lambda$ and outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, \boxed{x_1^*}, s^*, \perp))$ instead of $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, \boxed{\perp}, s^*, \perp))$.

Note that for each function $f \in \{f_1, \ldots, f_T\}$ with an associated PRF key K, for the deterministic function $\mathsf{Left}_{f,K}$ and the challenge ciphertext it holds that

$\mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Left}_{f,K}(x_b^*, \perp, s^*, \perp)$. Therefore, the message privacy of the underlying scheme FE (with respect to *deterministic* functions) guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(0)}$ and $\mathcal{H}^{(1)}$. Specifically, let $\mathcal{F}'$ denote the family of deterministic functions $\mathsf{Left}_{f,K}$ and $\mathsf{Right}_{f,K}$ for every $f \in \mathcal{F}$ and PRF key K (as defined in Figure 1) as well as the function $\mathsf{PuncOutputY}_{f,K',y,s^*}$ and $\mathsf{PuncOutputZ}_{f,K',s^*}$ for every $f \in \mathcal{F}$, punctured PRF key K', value $y \in \mathcal{Y}_\lambda$ and string $s^* \in \{0,1\}^\lambda$ (as defined in Figure 2). In the full version (see [26]) we prove the following lemma:

Lemma 1. *There exists a probabilistic polynomial-time adversary $\mathcal{B}^{(0)\to(1)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(0)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(1)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{MP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(0)\to(1)},T}(\lambda).$$

Experiment $\mathcal{H}^{(2,i)}(\lambda)$ where $i \in [T_2 + 1]$. This experiment is obtained from the experiment $\mathcal{H}^{(1)}(\lambda)$ by modifying the post challenge key-generation oracle to generate keys as follows. The functional keys for the $f_{T_1+1}, \ldots, f_{T_1+i-1}$ are generated as $\mathsf{PuncOutputY}_{f,K',y,s^*}$ (the definition of $\mathsf{PuncOutputY}_{f,K',y,s^*}$ appears in Figure 2), where K' is generated by sampling a PRF key $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and then puncturing it at s^*, and where $y \leftarrow f(x_b^*)$, and the functional keys for $f_{T_1+i}, \ldots, f_{T_1+T_2} = f_T$ are generated as $\mathsf{PuncOutputY}_{f,K',y,s^*}$, where K' and s^* are as before but $y = f(x_b^*; \mathsf{PRF}_K(s^*))$.

Note that every $x \neq x_b^*$ with which the encryption oracle is queries (with probability negligibly close to 1) it holds that $s \neq s^*$, hence, using the functionality feature of the punctured PRF, for every $f \in \{f_{T_1+1}, \ldots, f_T\}$ it holds that $\mathsf{Left}_{f,K}(x, x, s, \perp) = \mathsf{PuncOutputY}_{f,K',y,s*}(x, x, s, \perp)$. In addition, for the challenge x_b^* it holds that $\mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{PuncOutputY}_{f,K',y,s*}(x_b^*, x_1^*, s^*, \perp)$ since $\mathsf{PuncOutputY}_{f,K',y,s*}$ simply outputs y, where $y = f(x_b^*; \mathsf{PRF}_K(s^*))$. Thus, the function-privacy of the underlying scheme FE guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(1)}(\lambda)$ and $\mathcal{H}^{(2,1)}(\lambda)$. In the full version (see [26]) we prove the following lemma:

Lemma 2. *There exists a probabilistic polynomial-time adversary $\mathcal{B}^{(1)\to(2,1)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(1)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(2,1)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{FP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(1)\to(2,1)},T}(\lambda) + \mathsf{neg}(\lambda).$$

Moreover, note that the pseudorandomness of $\mathsf{PRF}_K(\cdot)$ at punctured point s^* (see Definition 2) guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(2,i)}$ and $\mathcal{H}^{(2,i+1)}$. In the full version (see [26]) we prove the following lemma:

Lemma 3. *For every $i \in [T_2]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(2,i)\to(2,i+1)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(2,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(2,i+1)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}_{\mathsf{puPRF},\mathcal{B}^{(2,i)\to(2,i+1)}}(\lambda).$$

Experiment $\mathcal{H}^{(3,i)}(\lambda)$ where $i \in [T_2 + 1]$. This experiment is obtained from the experiment $\mathcal{H}^{(2,T_2)}(\lambda)$ by modifying the post-challenge key-generation oracle as follows. The functional keys for the $f_{T_1+1}, \ldots, f_{T_1+i-1}$ are generated as $\mathsf{PuncOutputY}_{f,K',y,s^*}$, where K' is generated by sampling a PRF key $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$ and then puncturing it at s^*, and where $y \leftarrow \boxed{f(x_1^*)}$, and the functional keys for $f_{T_1+i}, \ldots, f_{T_1+T_2}$ are generated as $\mathsf{PuncOutputY}_{f,K',y,s^*}$, where K' and s^* are as before but $y \leftarrow f(x_b^*)$. We observe that $\mathcal{H}^{(2,T+1)}(\lambda) = \mathcal{H}^{(3,1)}(\lambda)$.

The adaptive admissibility of the function family $\mathcal{F}$ (see Definition 11) guarantee that the advantage of the adversary $\mathcal{A}$ in distinguishing experiments $\mathcal{H}^{(3,i)}$ and $\mathcal{H}^{(3,i+1)}$ is at most $\Delta(\lambda)$. In the full version (see [26]) we prove the following lemma:

Lemma 4. *For every $i \in [T_2]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(3,i)\to(3,i+1)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(3,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(3,i+1)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{aADM}}_{\mathcal{F},\mathcal{B}^{(3,i)\to(3,i+1)}} \leq \Delta(\lambda).$$

Experiment $\mathcal{H}^{(4,i)}(\lambda)$ where $i \in [T_1 + 1]$. This experiment is obtained from the experiment $\mathcal{H}^{(3,T)}(\lambda)$ by modifying the pre-challenge key-generation oracle as follows. The functional keys for $f_1, ..., f_{i-1}$ are generated as $\mathsf{sk}_f \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{Right}_{f,K}})$ instead of as $\mathsf{sk}_f \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{Left}_{f,K}})$ (where $\mathsf{Right}_{f,K}$ is defined in Figure 1), and the functional keys for $f_i, ..., f_{T_1}$ are generated as before (i.e., as $\mathsf{sk}_f \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \mathsf{Left}_{f,K})$). We observe that $\mathcal{H}^{(3,T+1)}(\lambda) = \mathcal{H}^{(4,1)}(\lambda)$.

Experiment $\mathcal{H}^{(5,i)}(\lambda)$ where $i \in [T_1]$. This experiment is obtained from the experiment $\mathcal{H}^{(4,i)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it samples $s^* \leftarrow \{0,1\}^\lambda$ and outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, x_1^*, s^*, \boxed{z^*}))$, where $z^* = f_i(x_b^*; \mathsf{PRF.Eval}(K_i, s^*))$, instead of $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, x_1^*, s^*, \boxed{\perp}))$.

Notice that both $\mathsf{Left}_{f,K}$ and $\mathsf{Right}_{f,K}$ are defined to ignore the fourth input z, hence, for the first $i-1$ keys it holds that $\mathsf{Right}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Right}_{f,K}(x_b^*, x_1^*, s^*, z^*)$ and for the next T_1-i+1 keys it holds that $\mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, z^*)$. Therefore, the message privacy of the underlying scheme FE guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(4,i)}$ and $\mathcal{H}^{(5,i)}$. In the full version (see [26]) we prove the following lemma:

Lemma 5. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(4,i)\to(5,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(4,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(5,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{MP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(4,i)\to(5,i)},T}(\lambda).$$

Experiment $\mathcal{H}^{(6,i)}(\lambda)$ where $i \in [T_1]$. This experiment is obtained from the experiment $\mathcal{H}^{(5,i)}(\lambda)$ by modifying the behavior of the pre-challenge key-generation oracle on the ith query f_i (without modifying its behavior on all other queries). On input the ith query f_i, the pre-challenge key-generation oracle compute $\mathsf{sk}_{f_i} \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{PuncOutputZ}_{f_i,K'_i,s^*}})$ instead of $\mathsf{sk}_{f_i} \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{Left}_{f_i,K_i}})$ (where the function $\mathsf{PuncOutputZ}_{f_i,K'_i,s^*}$ is defined in Figure 2).

Note that by the functionality feature of the punctured PRF (see Definition 2), for every ciphertext $(x, \perp, s, z)$ which is not the challenge ciphertext (with probability negligibly close to 1) it holds that $\mathsf{PuncOutputZ}_{f_i,K'_i,s^*}(x, \perp, s, z) = \mathsf{Left}_{f_i,K_i}(x, \perp, s, z)$ (since $s \neq s^*$ with very high probability). For the challenge ciphertext the latter also holds since $\mathsf{PuncOutputZ}_{f_i,K'_i,s^*}(x_b^*, x_1^*, s^*, z^*)$ outputs $z^* = f_i(x_b^*; \mathsf{PRF}_{K_i}(s^*))$. Thus, the function-privacy of the underlying scheme FE guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(6,i)}(\lambda)$ and $\mathcal{H}^{(7,i)}(\lambda)$. In the full version (see [26]) we prove the following lemma:

Lemma 6. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(5,i)\rightarrow(6,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(5,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(6,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{FP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(5,i)\rightarrow(6,i)},T}(\lambda) + \mathsf{neg}(\lambda).$$

Experiment $\boldsymbol{\mathcal{H}^{(7,i)}(\lambda)}$ where $\boldsymbol{i \in [T_1]}$. This experiment is obtained from the experiment $\mathcal{H}^{(6,i)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, x_1^*, s^*, z^*))$, where $z^* = f_i(x_b^*; \boxed{r^*})$ for a fresh and uniformly sampled value r^* instead of $z^* = f_i(x_b^*; \boxed{\mathsf{PRF.Eval}(K_i, s^*)})$.

The pseudorandomness at punctured point s^* of $\mathsf{PRF.Eval}(K_i, \cdot)$ guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(6,i)}$ and $\mathcal{H}^{(7,i)}$. In the full version (see [26]) we prove the following lemma:

Lemma 7. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(6,i)\rightarrow(7,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(6,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(7,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}_{\mathsf{puPRF},\mathcal{B}^{(6,i)\rightarrow(7,i)}}(\lambda).$$

Experiment $\boldsymbol{\mathcal{H}^{(8,i)}(\lambda)}$ where $\boldsymbol{i \in [T_1]}$. This experiment is obtained from the experiment $\mathcal{H}^{(7,i)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, x_1^*, s^*, z^*))$, where $z^* = f_i(\boxed{x_1^*}; r^*)$ instead of $z^* = f_i(\boxed{x_b^*}; r^*)$ (both with fresh and uniform r^*).

The adaptive admissibility of the function family $\mathcal{F}$ (see Definition 11) guarantees that the advantage of the adversary $\mathcal{A}$ in distinguishing experiments $\mathcal{H}^{(7,i)}$ and $\mathcal{H}^{(8,i)}$ is at most $\Delta(\lambda)$. In the full version (see [26]) we prove the following lemma:

Lemma 8. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(7,i)\to(8,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(7,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(8,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{aADM}}_{\mathcal{F},\mathcal{B}^{(7,i)\to(8,i)}} \leq \Delta(\lambda).$$

Experiment $\mathcal{H}^{(9,i)}(\lambda)$ where $i \in [T_1]$. This experiment is obtained from the experiment $\mathcal{H}^{(8,i)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (x_b^*, x_1^*, s^*, z^*))$, where $z^* = f_i(x_1^*; \boxed{\mathsf{PRF.Eval}(K_i, s^*)})$ instead of $z^* = f_i(x_1^*; \boxed{r^*})$ for a fresh and uniformly sampled value r^*.

The pseudorandomness at punctured point s^* of $\mathsf{PRF.Eval}(K_i, \cdot)$ guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(9,i)}$ and $\mathcal{H}^{(10,i)}$. The proof of the following lemma is essentially identical to the proof of Lemma 7 (see [26]):

Lemma 9. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(8,i)\to(9,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(8,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(9,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}_{\mathsf{puPRF},\mathcal{B}^{(8,i)\to(9,i)}}(\lambda).$$

Experiment $\mathcal{H}^{(10,i)}(\lambda)$ where $i \in [T_1]$. This experiment is obtained from the experiment $\mathcal{H}^{(9,i)}(\lambda)$ by modifying the behavior of the pre-challenge key-generation oracle on the ith query f_i (without modifying its behavior on all other queries). On input the ith query f_i, the key-generation oracle compute $\mathsf{sk}_{f_i} \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{Right}_{f_i,K_i}})$ instead of $\mathsf{sk}_{f_i} \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \boxed{\mathsf{PuncOutputZ}_{f_i,K'_i,s^*}})$.

As in the proof of Lemma 6, the function privacy of the underlying scheme FE (with respect to *deterministic* functions) guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(9,i)}$ and $\mathcal{H}^{(10,i)}$. The proof of the following lemma is essentially identical to the proof of Lemma 6 (see [26]):

Lemma 10. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(9,i)\to(10,i)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(9,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(10,i)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{FP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(9,i)\to(10,i)},T}(\lambda) + \mathsf{neg}(\lambda).$$

Next, we observe that experiment $\mathcal{H}^{(4,i+1)}(\lambda)$ is obtained from the experiment $\mathcal{H}^{(10,i)}(\lambda)$ by modifying the challenge ciphertext to be computed using $z^* = \perp$ instead of $z^* = f_i(x_1^*; \mathsf{PRF.Eval}(K_i, s^*))$.

Note that for each function $f \in \{f_1, \ldots, f_T\}$ with an associated PRF key K, for the deterministic functions $\mathsf{Left}_{f,K}$ and $\mathsf{Right}_{f,K}$ and the challenge ciphertext it holds that $\mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Left}_{f,K}(x_b^*, x_1^*, s^*, z^*)$ and $\mathsf{Right}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Right}_{f,K}(x_b^*, x_1^*, s^*, z^*)$. Therefore, the selective-function message privacy of the underlying scheme FE (with respect to *deterministic* functions) guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(10,i)}$ and $\mathcal{H}^{(4,i+1)}$. The proof of the following lemma is essentially identical to the proof of Lemma 5 (see [26]):

Lemma 11. *For every $i \in [T_1]$ there exists a probabilistic polynomial-time adversary $\mathcal{B}^{(10,i)\to(4,i+1)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(10,i)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(4,i+1)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{MP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(10,i)\to(4,i+1)},T}(\lambda).$$

Experiment $\boldsymbol{\mathcal{H}^{(11)}(\lambda)}$. This experiment is obtained from the experiment $\mathcal{H}$ $^{(4,T+1)}(\lambda)$ by modifying the encryption oracle so that on the challenge input (x_0^*, x_1^*) it outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk}, (\boxed{x_1^*}, x_1^*, s^*, \perp))$ instead of $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{msk},$ $(\boxed{x_b^*}, x_1^*, s^*, \perp))$. Note that this experiment is completely independent of the bit b, and therefore $\Pr\left[\mathcal{H}^{(11)}(\lambda) = 1\right] = 1/2$.

In addition, note that for every function $f \in \{f_1, \ldots, f_{T_1}\}$ with an associated PRF key K, for the deterministic function $\mathsf{Right}_{f,K}$ it holds that $\mathsf{Right}_{f,K}(x_b^*, x_1^*, s^*, \perp) = \mathsf{Right}_{f,K}(x_1^*, x_1^*, s^*, \perp)$. Therefore, the message privacy of the underlying scheme FE (with respect to *deterministic* functions) guarantees that the adversary $\mathcal{A}$ has only a negligible advantage in distinguishing experiments $\mathcal{H}^{(4,T_1+1)}$ and $\mathcal{H}^{(11)}$. The proof of the following lemma is essentially identical to the proof of Lemma 1 (see [26]):

Lemma 12. *There exists a probabilistic polynomial-time adversary $\mathcal{B}^{(4,T_1+1)\to(11)}$ such that*

$$\left|\Pr\left[\mathcal{H}^{(4,T_1+1)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}^{(11)}(\lambda) = 1\right]\right| \leq \mathsf{Adv}^{\mathsf{sfFP}}_{\mathsf{FE},\mathcal{F}',\mathcal{B}^{(4,T+1)\to(11)},T}(\lambda).$$

Finally, putting together Lemmas 1–12 with the facts that $\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda) = \mathcal{H}^{(0)}(\lambda)$, $\mathcal{H}^{(1)}(\lambda) = \mathcal{H}^{(2,1)}(\lambda)$, $\mathcal{H}^{(2,T+1)}(\lambda) = \mathcal{H}^{(3,1)}(\lambda)$, $\mathcal{H}^{(3,T+1)}(\lambda) = \mathcal{H}^{(4,1)}(\lambda)$

and $\Pr\left[\mathcal{H}^{(11)}(\lambda)=1\right]=1/2$, we observe that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T} &\overset{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Expt}^{\mathsf{aMPRF}}_{\Pi,\mathcal{F},\mathcal{A},T}(\lambda)=1\right]-\frac{1}{2}\right| \\
&= \left|\Pr\left[\mathcal{H}^{(0)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(11)}(\lambda)=1\right]\right| \\
&\leq \left|\Pr\left[\mathcal{H}^{(0)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(1)}(\lambda)=1\right]\right| \\
&\quad+\left|\Pr\left[\mathcal{H}^{(1)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(2,1)}(\lambda)=1\right]\right| \\
&\quad+\sum_{j=2}^{3}\sum_{i=1}^{T_2}\left|\Pr\left[\mathcal{H}^{(j,i)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(j,i+1)}(\lambda)=1\right]\right| \\
&\quad+\sum_{i=1}^{T_1}\sum_{j=4}^{9}\left|\Pr\left[\mathcal{H}^{(j,i)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(j+1,i)}(\lambda)=1\right]\right| \\
&\quad+\sum_{i=1}^{T_1}\left|\Pr\left[\mathcal{H}^{(10,i)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(4,i+1)}(\lambda)=1\right]\right| \\
&\quad+\left|\Pr\left[\mathcal{H}^{(4,T+1)}(\lambda)=1\right]-\Pr\left[\mathcal{H}^{(11)}(\lambda)=1\right]\right| \\
&\leq (T_1(\lambda)+T_2(\lambda))\cdot\Delta(\lambda)+\mathsf{neg}(\lambda) \\
&= T(\lambda)\cdot\Delta(\lambda)+\mathsf{neg}(\lambda).
\end{aligned}$$

■

Acknowledgments

We thank Zvika Brakerski for various insightful discussions.

References

1. Agrawal, S., Agrawal, S., Badrinarayanan, S., Kumarasubramanian, A., Prabhakaran, M., Sahai, A.: Function private functional encryption and property preserving encryption: New definitions and positive results. Cryptology ePrint Archive, Report 2013/744 (2013)
2. Agrawal, S., Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption: New perspectives and lower bounds. In: Advances in Cryptology – CRYPTO '13. pp. 500–518 (2013)
3. Alwen, J., Barbosa, M., Farshim, P., Gennaro, R., Gordon, S.D., Tessaro, S., Wilson, D.A.: On the relationship between functional encryption, obfuscation, and fully homomorphic encryption. In: Proceedings of the 14th International Conference on Cryptography and Coding - IMACC. pp. 65–84 (2013)
4. Ananth, P., Boneh, D., Garg, S., Sahai, A., Zhandry, M.: Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689 (2013)
5. Ananth, P., Brakerski, Z., Segev, G., Vaikuntanathan, V.: The trojan method in functional encryption: From selective to adaptive security, generically. Cryptology ePrint Archive, Report 2014/917 (2014)

6. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S.P., Yang, K.: On the (im)possibility of obfuscating programs. Journal of the ACM 59(2), 6 (2012)
7. Bellare, M., O'Neill, A.: Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In: Proceedings of the 12th International Conference on Cryptology and Network Security. pp. 218–234 (2013)
8. Boneh, D., Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM Journal on Computing 32(3), 586–615 (2003), preliminary version in *Advances in Cryptology – CRYPTO '01*, pages 213–229, 2001
9. Boneh, D., Raghunathan, A., Segev, G.: Function-private identity-based encryption: Hiding the function in functional encryption. In: Advances in Cryptology – CRYPTO '13. pp. 461–478 (2013)
10. Boneh, D., Raghunathan, A., Segev, G.: Function-private subspace-membership encryption and its applications. In: Advances in Cryptology – ASIACRYPT '13. pp. 255–275 (2013)
11. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Proceedings of the 8th Theory of Cryptography Conference. pp. 253–273 (2011)
12. Boneh, D., Waters, B.: Constrained pseudorandom functions and their applications. In: Advances in Cryptology - ASIACRYPT '13. pp. 280–300 (2013)
13. Boyle, E., Chung, K., Pass, R.: On extractability obfuscation. In: Proceedings of the 11th Theory of Cryptography Conference. pp. 52–73 (2014)
14. Boyle, E., Goldwasser, S., Ivan, I.: Functional signatures and pseudorandom functions. In: Proceedings of the 17th International Conference on Practice and Theory in Public-Key Cryptography. pp. 501–519 (2014)
15. Brakerski, Z., Segev, G.: Function-private functional encryption in the private-key setting. Cryptology ePrint Archive, Report 2014/550 (2014)
16. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Proceedings of the 8th IMA International Conference on Cryptography and Coding. pp. 360–363 (2001)
17. De Caro, A., Iovino, V., Jain, A., O'Neill, A., Paneth, O., Persiano, G.: On the achievability of simulation-based security for functional encryption. In: Advances in Cryptology – CRYPTO '13. pp. 519–535 (2013)
18. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science. pp. 40–49 (2013)
19. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Fully secure functional encryption without obfuscation. Cryptology ePrint Archive, Report 2014/666 (2014)
20. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. Journal of the ACM 33(4), 792–807 (1986)
21. Goldwasser, S., Gordon, S.D., Goyal, V., Jain, A., Katz, J., Liu, F.H., Sahai, A., Shi, E., Zhou, H.S.: Multi-input functional encryption. In: Advances in Cryptology – EUROCRYPT '14. pp. 578–602 (2014)
22. Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Proceedings of the 45th Annual ACM Symposium on Theory of Computing. pp. 555–564 (2013)
23. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Functional encryption with bounded collusions via multi-party computation. In: Advances in Cryptology – CRYPTO '12. pp. 162–179 (2012)

24. Goyal, V., Jain, A., Koppula, V., Sahai, A.: Functional encryption for randomized functionalities. Cryptology ePrint Archive, Report 2013/729 (2013), to appear in TCC 2015
25. Kiayias, A., Papadopoulos, S., Triandopoulos, N., Zacharias, T.: Delegatable pseudorandom functions and applications. In: Proceedings of the 20th Annual ACM Conference on Computer and Communications Security. pp. 669–684 (2013)
26. Komargodski, I., Segev, G., Yogev, E.: Functional encryption for randomized functionalities in the private-key setting from minimal assumptions. Cryptology ePrint Archive, Report 2014/868 (2014)
27. O'Neill, A.: Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556 (2010)
28. Sahai, A., Waters, B.: Slides on functional encryption. Available at `http://www.cs.utexas.edu/~bwaters/presentations/files/functional.ppt` (2008)
29. Sahai, A., Waters, B.: How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing. pp. 475–484 (2014)
30. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in Cryptology – CRYPTO '84. pp. 47–53 (1984)
31. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Proceedings of the 6th Theory of Cryptography Conference. pp. 457–473 (2009)
32. Waters, B.: A punctured programming approach to adaptively secure functional encryption. Cryptology ePrint Archive, Report 2014/588 (2014)