# On the Power of Public-key Functional Encryption with Function Privacy

Vincenzo Iovino[1] Qiang Tang[1] Karol Żebrowski[2]

[1] University of Luxembourg, {vincenzo.iovino, qiang.tang}@uni.lu
[2] University of Warsaw, kz277580@students.mimuw.edu.pl

## Abstract

In CRYPTO 2014 Bitansky *et al.* introduced a natural strengthening of indistinguishability obfuscation (iO) called *strong iO* (siO) and showed candidate constructions of such primitive from reasonable assumptions. In this paper, assuming quasi-siO, a natural weakening of siO, for a class of circuits $\mathcal{C}$ we construct a *public-key* functional encryption (FE) scheme with *function privacy* (FPFE) for the same class $\mathcal{C}$. In the public-key setting known constructions of FPFE were limited to very restricted classes of functionalities like inner-product [Agrawal *et al.* - PKC 2015] whereas ours can be instantiated for *general* functionalities.

Then, inspired by the Naor's transformation from IBE to signature schemes, we construct from FPFE a natural generalization of a signature scheme endowed with functional properties, that we call *functional anonymous signature* (FAS) scheme. In a FAS (that we show to be equivalent to quasi-siO and FPFE), Alice can sign a circuit $C$ chosen from some distribution $D$ to get a signature $\sigma$ and can publish a verification key that allows anybody holding a message $m$ to verify that (1) $\sigma$ is a valid signature of Alice for *some* (possibly unknown to him) circuit $C$ and (2) $C(m) = 1$. Beyond unforgeability the security of FAS guarantees that the signature $\sigma$ hide as much information as possible about $C$ except what can be inferred from knowledge of $D$. As other application of FPFE, we show that it can be used to construct in a black-box way (without using obfuscation directly) FE for randomized functionalities (RFE). Previous constructions of (public-key) RFE relied on iO [Goyal *et al.* - TCC 2015]. Combining properties of RFE and FAS, we obtain what we call *signed probabilistic programs*, in which Bob can verify that a (possibly hidden to him) probabilistic program $P$ was signed by Alice and run $P$ on any input but can not maliciously choose its random coins.

Furthermore, our constructions of FPFE and RFE naturally generalize to the *multi-inputs* setting. Finally, we present a general picture of the relations among all these related primitives. One of the key points that such implications draw is that Attribute-based Encryption with function privacy implies FE, a notable fact that sheds light on the importance and power of function privacy for FE.

**Keywords:** Functional Encryption, Function Privacy, Obfuscation, Digital Signatures.

# Contents

# 1 Introduction

**Strong indistinguishability obfuscation** In CRYPTO 2014 Bitansky, Canetti, Kalai and Paneth [BCKP14a] introduced a natural strengthening of indistinguishability obfuscation (iO) called *strong iO* (siO) and showed candidate constructions of such primitive for all circuits in $\mathsf{NC}^1$ from variants of semantically secure graded encoding schemes [PST14]. Informally speaking, a siO is secure if no efficient adversary can distinguish whether a circuit was drawn from $D_0$ or $D_1$ where $D_0$ and $D_1$ are a pair of 'feasible' entropy distributions in the sense that no adversary can distinguish oracle access to a circuit drawn from $D_0$ or $D_1$. In this paper, we show several applications of siO. Specifically, for our applications it suffices to consider a natural weakening of siO, already pointed out by Bitansky *et al.*, requiring that the distributions be *efficiently samplable.* We call such notion *quasi-siO.* First, assuming quasi-siO (and one-way functions) we construct functional encryption schemes with function privacy that we describe next.

## 1.1 Functional Encryption with Function Privacy

Functional Encryption (FE, in short) is a sophisticated type of encryption that allows to finely control the amount of information that is revealed by a ciphertext. Progressively more expressive forms of FE were constructed in a series of works (see, e.g., [BDOP04, BW07, KSW08, LOS$^+$10, OT12, Wat12]) culminating in the breakthrough of Garg *et al.* [GGH$^+$13]. The security notion in these works only take in account the privacy of the *message* but nothing was guaranteed for the privacy of the *function.* In the symmetric-key setting, a preliminary study of FE with function privacy was initiated by Shen *et al.* [SSW09] for the inner-product [KSW08] functionality, subsequently followed by constructions for general functionalities [BS15]. Boneh *et al.* [BRS13a] put forward the study of function privacy for FE providing constructions for the Identity-Based Encryption (IBE) functionality, then followed by works that considered the subspace membership [BRS13b] and the inner-product [AAB$^+$13, AAB$^+$15] functionalities. In a public-key setting, the function can not be hidden completely since the adversary can never infer partial information about it using the public-key. For such reason, Boneh *et al.* [BRS13a] consider functions chosen from *high min-entropy* distributions. Precisely, in the context of IBE they propose an IND style real-or-random definition of function privacy, that stipulates that as long as the identity $\mathsf{id}$ was chosen from a sufficiently high min-entropy distribution, the adversary should not be able to distinguish the token for $\mathsf{id}$ from a token for a uniformly random identity. Agrawal *et al.* [AAB$^+$15] also consider stronger simulation-based definitions for function privacy but with non-standard simulators (a necessity motivated by broad impossibility results in the area). A bit of thought shows that a meaningful simulation-based security notion of function private FE (FPFE, in short) for some enough expressive class of Boolean circuits would imply virtual black box obfuscation for the same class of circuits and thus it seems unachievable even for $\mathsf{NC}^1$ circuits. For such reasons, we stick with the indistinguishability-based definition and defer to future works the study of stronger security notions. Specifically, in the case of Boolean circuits, we consider what we call pairs of ensembles of efficiently samplable feasible entropy distributions, a strengthening of a notion defined by Agrawal *et al.* [AAB$^+$15] which abstracts the unpredictability property of Boneh *et al.* [BRS13a]. Formal definition is given in Section 2. Note that we put the constraint that the distributions be efficiently samplable. This is because, in the context of function privacy, as well as for functional anonymous signatures that we will introduce later, users sample the cryptographic objects from efficiently samplable distributions. This subtle difference turns to be very important, indeed it is the key to make such primitives *composable.* In Section 2.2 we discuss the existence of siO and quasi-siO. To our knowledge no previous work in literature considered public-key FPFE for general functionalities. This leads to the main questions that we study in this work:

> Can we achieve public-key FPFE for *all polynomial-sized circuits* from reasonable assumptions? And what applications and other primitives can we build from it?

Our first result answers *affirmatively* to the first question.

**FE with Function Privacy from quasi-siO.** Firs of all, it is worth reminding why existing constructions of FE do not offer any meaningful function privacy. Consider the construction of Garg *et al.* [GGH+13] of FE from iO. Therein, the token for a circuit $C$ is an indistinguishability obfuscation of $C$. One could hope that being the circuit obfuscated it should hide as much information as possible about the circuit. Nevertheless, the form of function privacy here attained is very limited. Specifically, the token for $C$ is indistinguishable from the token for any other *functionally equivalent* circuit $C'$. To show that this is insufficient in many concrete applications, consider the case of circuits implementing point functions. Specifically, for any binary string $x \in \{0,1\}^n$ consider the class of circuits $\mathcal{C}_x$ that contain all circuits $C$ defined so that $C$ on input a binary string $y$ of length $n$ outputs 1 if and only if $y = x$. Then, the class of circuits implementing point functions, let us say restricted to points of length $n$, is the union of all $\mathcal{C}_x$'s for all strings $x$ of length $n$. It is trivial to notice that an iO for this class could just return the value $x$ in clear[1], assuming that this can be done efficiently. That is, the (non necessarily efficient) obfuscator that on input a circuit $C \in \mathcal{C}_x$ for some $x \in \{0,1\}^n$ outputs $x$ in clear (with evaluation procedure associated in the obvious way) is *provably* an iO. Notwithstanding, this obfuscator when plugged in FE does not offer any guarantee of function privacy for these classes of functions. In fact, consider two distributions $D_0$ and $D_1$ over strings in $\{0,1\}^n$ defined so that the first bit in the strings drawn from $D_b$, for $b \in \{0,1\}$ is $b$ and the remaining bits are uniformly and independently chosen. Then, a token for a point $x$ drawn from $D_0$ can be easily distinguished from a token for a point drawn from $D_1$. This is because the obfuscated point leaks $x$ in clear and looking just at the first bit of it, the token can be distinguished. This motivates the use of siO. Indeed, if the token was instead a siO of the circuit, it would leak as few information as possible about the circuit. To the aim of having conceptually simple and general constructions, we construct a FPFE scheme by nesting a generic FE scheme (without function privacy) with a siO. Specifically our FPFE scheme $\mathsf{FPFE}$ will use the underlying FE $\mathsf{FE}$ scheme as black-box and will have identical procedures except that a token for a circuit $C$ will consist of a token of $\mathsf{FE}$ for the circuit $\mathsf{qsi}\mathcal{O}(C)$, where $\mathsf{qsi}\mathcal{O}$ is a quasi-siO: that is, setting $C' = \mathsf{qsi}\mathcal{O}(C)$, a token of $\mathsf{FPFE}$ for $C$ will be a token of $\mathsf{FE}$ for $C'$. Intuitively, even though this token is computed with a non function private scheme, as it is built on the top of circuit obfuscated with quasi-siO, it should leak as few information as possible. In fact, we confirm this intuition providing formal reductions. Note here that the underlying FE scheme guarantees the privacy of the encrypted messages and quasi-siO is only used to add the extra layer of function privacy. Furthermore, the modularity of this approach generalizes easily to *multi-inputs FE* (MIFE, in short) [GGG+14] allowing to construct the first MIFE scheme with function privacy (FPMIFE, in short). The definition of a FPFE scheme and its security are presented in Section 2.4 and its construction from quasi-siO is presented in Section 3.

We observe that the reverse direction also holds. In fact, a quasi-siO $\mathsf{qsi}\mathcal{O}$ for class of circuits $\mathcal{C}$ can be constructed from a FPFE $\mathsf{FPFE}$ for the same class in the following way. For any input $C$ the algorithm $\mathsf{qsi}\mathcal{O}(C)$ outputs the public-key of the system $\mathsf{FPFE}$ and a token $\mathsf{Tok}$ for $C$ of $\mathsf{FPFE}$.

[1] Precisely, we also have to define a corresponding evaluation procedure in the obvious way.

To evaluate such obfuscated circuit on an input $x$, the evaluation algorithm associated with $\mathsf{qsi}\mathcal{O}$ takes as input the public-key and $\mathsf{Tok}$ and encrypts[2] $x$ to get $\mathsf{Ct}$ and evaluates $\mathsf{Tok}$ on $\mathsf{Ct}$ to get $C(m)$. It is easy to see that the correctness of FPFE and our definition of $\mathsf{INDFP}$-Security as defined in Section 2.4 imply that the constructed obfuscator is a quasi-siO. This construction also shows, as said before, that a meaningful simulation-based security notion for FPFE for general circuits would imply VBB obfuscation for general circuits, and thus is unachievable. For such reason we stick to an indistinguishability-based definition of function privacy.

## 1.2 Functional Anonymous Signatures

Recall that the Naor's transformation[3] allows to transform an identity-based encryption (IBE) scheme [BF01] (a special case of FE) in a signature scheme. The idea is that the token for an identity $\mathsf{id}$ (encoded as binary string) acts as a signature for it. Such signature can be verified by encrypting (using the public-key of the scheme) the pair $(r, \mathsf{id})$ for a random string $r$ and testing whether the token (i.e., the signature) evaluated on such ciphertext returns $r$. By the security property of FE, such signature is unforgeable. Suppose that we generalize this idea to FE for general circuits. What would it be the benefit in this case? For instance, this extensions would enable Alice to sign a Boolean circuit $C$ allowing Bob holding an input $m$ to verify that the signature was issued by Alice and $C(m) = 1$. We envision a scenario where the signature of Alice of a circuit $C$ hides $C$ if it is drawn from a feasible entropy distribution. In this case, the intent of Bob is to verify (1) that Alice signed *some* circuit $C$, that is not known to him, and (2) verify that his input $m$ satisfies the circuit, e.g., $C(m) = 1$. Before defining the security of this primitive, that we call *Functional Anonymous Signature* (FAS, in short) scheme, we consider an application scenario.

**Applications of FAS.** FAS can be used to implement an authenticated policy mechanism. Alice, the head of a company, can publish her verification key and with the corresponding secret key can sign an hidden policy $P$ chosen from some known distribution $D$ and send the signature $\sigma$ of $P$ to the server of her company. The secretary of the company, who is assumed to be honest but curious, can grant Bob access to some private document iff the access pattern $m$ held by Bob verifies the signature of Alice, and in particular her hidden policy, i.e., $P(m) = 1$. If the signature is verified by the access pattern of Bob, then the secretary has the guarantee that (1) the policy was signed by Alice and (2) the access pattern of Bob satisfies such policy. Both Bob and the secretary have no information about the policy except what can be inferred from the distribution $D$. Due to the possibility of using universal circuits in FAS, the role of access pattern and policy can be inverted, that is Alice can sign an access pattern and Bob holding a policy can verify whether his policy satisfies her access pattern. It is easy to see that FAS implies traditional signature schemes. Another powerful application related to FAS is given in Section 1.3.

**Security of FAS.** We define FAS with a notion of unforgeability that we call *functional unforgeability*, that suits for most applications of FAS. The notion does not consider as valid the forgery of a circuit more restricted than a circuit for which a signature was seen. That is, it is not considered as a valid forgery if an adversary given a signature of circuit $C$ can sign another circuit $C'$ that computes the same function as $C$ or is more restricted than $C$. To see why such condition is not too restrictive, consider the above application. In that case, the security of FAS

[2] Actually, for this implication to hold we only need "data privacy", i.e., security of the encryptions. In fact, we could assume that the messages are encrypted in clear. Precisely, according to the definitions from Section 2.4, we only need $\mathsf{INDFP}$-Security and not also $\mathsf{IND}$-Security.

[3] Such transformation was first reported in Boneh and Franklin [BF01].

should prevent some unauthorized user to claim that Alice signed a document who authorizes him. This is exactly what the condition states. Note also that being Alice semi-trusted we do not consider a breach of security if she is able to forge a signature for a circuit $C'$ more restricted than the circuit $C$ of which she received a signature from Alice (a circuit $C'$ is said to be more restricted than $C$ if $C'(x) = 1$ implies $C(x) = 1$). Only malicious users have the interest to forge new signatures and in this case their scope is to forge signatures for circuits that authorize them, so a forgery for a more restricted circuit (or a functionally equivalent one) must not be considered a successful attack. Anyway, for other applications such security could not suffice but we show that it is possible to make FAS unforgeable according to the classical notion of unforgeability (i.e., requiring that any PPT adversary can not forge a signature for a circuit $C'$ different (as bit string) from any circuit $C$ for which it saw a signature) just adding a traditional unforgeable scheme on the top of it. Beyond unforgeability, we require *anonymity*, namely that a signature $\sigma$ hide as much information as possible about $C$ except what can be inferred from knowledge of the distribution from which $C$ is drawn. FPFE fits perfectly in the picture, and in fact we show that it implies FAS in a black-box way. Specifically, we show how to extend the Naor's transformation to construct FAS for a class of circuits $\mathcal{C}$ from Attribute-based Encryption (ABE, in short) [GPSW06] with function privacy, a weaker notion of FPFE, for the same class $\mathcal{C}$. We remark that despite of the name, FAS does not share much similarities with functional signatures as defined by Goldwasser *et al.*[BGI14]. More related primitives are content-concealing signatures and confidential signatures ([Can97, DFM$^+$10]) that can be viewed as a weak form of FAS schemes without functional capabilities (or alternatively for the class of equality predicates).

The definition of FAS and its security are presented in Section 2.6 and its construction from ABE with function privacy (FPABE, in short) is presented in Section 4.

### 1.3 Functional Encryption for Randomized Functionalities (and SPP)

Goyal *et al.* [GJKS15] put forward the first construction of FE supporting *randomized* circuits. In this setting, the challenge is to guarantee that the circuit be evaluated on fresh randomness that can not be maliciously chosen. A tentative solution to the problem would be to include the seed of a pseudo-random function in the token. Unfortunately, this approach fails since the token is not guaranteed to hide the function that the circuit is supposed to compute. This leaves open the possibility that this basic idea could work assuming a FE whose token hides the function (i.e., with function privacy), and in fact we are able to confirm this intuition by showing a black-box construction of FE for randomized circuits (RFE, in short) from FPFE for (deterministic) circuits. We adopt an indistinguishability-based security for RFE, but unlike Goyal *et al.* we do not take in account the problem of dishonest encryptors that goes beyond the scope of our paper (and concerns not only RFE but FE and FPE as well). Our construction of RFE also preserves the function privacy of the underlying FPFE and thus satisfies the standard notion of function privacy where the adversary can ask distributions of *deterministic* circuits. We call this notion FPRFE. We believe that it also satisfies a form of function privacy extended in a natural way to support randomized circuits, but we did not investigate the details. Our construction of RFE can be easily extended to the multi-inputs setting, resulting in the first construction, assuming only quasi-siO, of a Public-key FPMIFE for randomized functionalities (as said before, where the function privacy is restricted to deterministic circuits) with selective form of indistinguishability security. The restriction of selective security can be removed assuming in addition an adaptively indistinguishability secure MIFE.

The definition of RFE and its security are presented in Section 2.5 and its construction from

FPFE is presented in Section 5.

**Signed probabilistic programs.** As FPABE implies FAS, FPRFE[4] implies the powerful primitive of *signed probabilistic programs* (SPP, in short) in which Alice can sign a probabilistic program $P$ that can be publicly verified and executed under the following guarantees. Bob with the verification key of Alice can check that a (possibly hidden to him) program $P$ was truly signed by her and Alice has the guarantee that Bob can execute $P$ on any input but he can not choose the random coins of $P$ (that look indistinguishable from random to Bob on any input). Furthermore, the signature also hides the program similarly to how FAS does. However, note that it would even make sense to define a weaker notion in which the signature contains the program in *clear*. Such weaker primitive can be constructed from RFE (without function privacy). We do not present formal definitions and further details of SPP but they can derived extending the properties FAS to randomized circuits as in RFE.

# 2 Definitions

## 2.1 Preliminaries

A *negligible* function $\mathsf{negl}(k)$ is a function that is smaller than the inverse of any polynomial in $k$. If $D$ is a probability distribution, the writing "$x \leftarrow D$" means that $x$ is chosen according to $D$. If $D$ is a finite set, the writing "$x \leftarrow D$" means that $x$ is chosen according to uniform probability on $D$. If $q > 0$ is an integer then $[q]$ denotes the set $\{1, \ldots, q\}$. If $B$ is an algorithm and $A$ is an algorithm with access to an oracle then $A^B$ denotes the execution of $A$ with oracle access to $B$. If $a$ and $b$ are arbitrary strings, then $a||b$ denotes the string representing their delimited concatenation. We assume a standard binary encoding for circuits, so for ease of exposition we define functions with binary inputs and invoke them with circuits as inputs instead of their binary encodings. In this paper, we will mostly work in the non-uniform model of computation. We say that $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ is a non-uniform family of PPT algorithm if there exists a polynomial $p(\cdot)$ such that for any $n \in \mathbb{N}$ machine $\mathcal{A}_n$ has size bounded by $p(n)$ and the running-time of $\mathcal{A}_n$ on any input $x$ of length $n$ is bounded by $p(n)$. Note that an algorithm $\mathcal{A}_n$ in the family $\mathcal{A}$ is only required to work for inputs of size $n$ but sometimes with a slight abuse of notation we give to $\mathcal{A}_n$ in input the security parameter $n$ and some other input of size polynomial related to $n$: for example we write $\mathcal{A}_n(1^n, \mathsf{Mpk})$ when we actually mean $\mathcal{A}_{n+|\mathsf{Mpk}|}(1^n, \mathsf{Mpk})$. With a slight abuse of notation, we write that a statement holds for all security parameters $\lambda$, whereas we actually mean that this has to hold only *for sufficiently large* values of $\lambda$. In our work, we make use of the following definition inspired by a similar definition from Agrawal *et al.* [AAB$^+$13, AAB$^+$15].

**Definition 2.1** [Pair of Ensembles of Feasible Entropy Distributions]. Let $D_0 = \{D_{0,n}\}_{n\in\mathbb{N}}$ and $D_1 = \{D_{1,n}\}_{n\in\mathbb{N}}$ be two ensembles of distributions over a class of circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ where any $n \in \mathbb{N}, \mathcal{C}_n$ contains circuits of the same size. Then, we say that $D_0$ and $D_1$ are a pair of ensembles of feasible entropy distributions, if for all non-uniform families of (possibly inefficient) algorithms $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ making a polynomial number of queries to its oracle (i.e., the adversaries are *semi-bounded*), it holds that:

$$\left|\Pr_{C\leftarrow D_0}\left[\mathcal{A}_n^{C(\cdot)}(1^n, 1^{|C|}) = 1\right] - \Pr_{C\leftarrow D_1}\left[\mathcal{A}_n^{C(\cdot)}(1^n, 1^{|C|}) = 1\right]\right| \leq \mathsf{negl}(n) .$$

[4] Precisely, for such implications to hold we just need to assume ABE for randomized circuits.

Note that in the above definition we do not require that the distributions be *efficiently samplable* but for all our applications we will put such additional constraint. So we will talk about a pair of ensembles of efficiently samplable feasible entropy distributions with the obvious meaning.

**Puncturable Pseudorandom Functions.** In this work, we make use of puncturable pseudorandom functions [SW14] which are essentially pseudorandom functions (PRFs, in short) that can be defined on all inputs except for a polynomial number of inputs.

**Definition 2.2** A puncturable family of PRFs $\mathsf{F}$ is given by a triple of Turing Machines $(\mathsf{F.Key}, \mathsf{F.Puncture}, \mathsf{F.Eval})$, and a pair of computable functions $n(\cdot)$ and $m(\cdot)$, satisfying the following conditions:

- Functionality preserved under puncturing: For every set $S \subset \{0,1\}^{n(\lambda)}$, for all $x \in \{0,1\}^{n(\lambda)}$ where $x \notin S$, we have that:
$\Pr\left[\, \mathsf{F.Eval}(\mathsf{K},\mathsf{x}) = \mathsf{F.Eval}(\mathsf{K_S},\mathsf{x}) : \mathsf{K} \leftarrow \mathsf{F.Key}(1^{\lambda}), \mathsf{K_S} = \mathsf{F.Puncture}(\mathsf{K},\mathsf{S}) \,\right] = 1.$

- Pseudorandom at punctured points: For every non-uniform family of PPT adversaries $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$, for every set $S \subset \{0,1\}^{n(\lambda)}$ consider an experiment where $K \leftarrow \mathsf{F.Key}(1^{\lambda})$ and $K_S = \mathsf{F.Puncture}(\mathsf{K},\mathsf{S})$. Then there exists a negligible function $\mathsf{negl}(\cdot)$ such that for any $\lambda \in \mathbb{N}$ such that we have:

$$\left|\Pr\left[\, \mathcal{A}_\lambda(K_S, S, \mathsf{F.Eval}(\mathsf{K},\mathsf{S})) = 1 \,\right] - \Pr\left[\, \mathcal{A}_\lambda(K_S, S, U_{m(\lambda)\cdot|S|}) = 1 \,\right]\right| \leq \mathsf{negl}(\lambda) \ .$$

  where $\mathsf{F.Eval}(\mathsf{K},\mathsf{S})$ denotes the concatenation of $(\mathsf{F.Eval}(\mathsf{K},\mathsf{x_1}), \ldots, \mathsf{F.Eval}(\mathsf{K},\mathsf{x_k}))$ where $S = \{x_1, \ldots, x_k\}$ is the enumeration of the elements of $S$ in lexicographic order, and $U_\ell$ denotes the uniform distribution over $\ell$ bits. For ease of notation, we write $\mathsf{F}(\mathsf{K},\mathsf{x})$ to represent $\mathsf{F.Eval}(\mathsf{K},\mathsf{x})$. We also represent the punctured key $\mathsf{F.Puncture}(\mathsf{K},\mathsf{S})$ by $K(S)$.

## 2.2 Strong and Quasi-strong Indistinguishability Obfuscation

Strong indistinguishability obfuscation has been introduced by Bitansky *et al.* [BCKP14a]. Their formulation is syntactically different from ours, but as they point out ([BCKP14b], p. 4) it is equivalent to ours. Thus, without loss of generality we adopt the following formulation as it is more suitable for our scopes.

**Definition 2.3** [Strong Indistinguishability Obfuscators for Circuits] A uniform PPT machine $\mathsf{si}\mathcal{O}$ is called a strong indistinguishability obfuscator (siO, in short) for a circuit family $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$, if the following conditions are satisfied:

- Correctness: $\forall n, \forall C \in \mathcal{C}_n, \forall x \in \{0,1\}^\star$ we have $\Pr\left[\, C'(x) = C(x) : C' \leftarrow \mathsf{si}\mathcal{O}(1^n, C) \,\right] = 1$.

- Strong indistinguishability: For all pairs of ensembles of feasible entropy distributions $D_0 = \{D_{0,n}\}_{n\in\mathbb{N}}$ and $D_1 = \{D_{1,n}\}_{n\in\mathbb{N}}$ over a class of Boolean circuits $\mathcal{C}' = \{\mathcal{C}'_n\}_{n\in\mathbb{N}} \subset \mathcal{C}$ where for any $n \in \mathbb{N}$ the set $\mathcal{C}'_n$ contains circuits of the *same* size, for any non-uniform family of PPT distinguishers $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that the following holds: For all $n \in \mathbb{N}$, we have that $|\Pr_{C\leftarrow D_{0,n}}\left[\, \mathcal{D}_n(1^n, 1^{|C|}, \mathsf{si}\mathcal{O}(1^n, C)) = 1 \,\right] - \Pr_{C\leftarrow D_{1,n}}\left[\, \mathcal{D}_n(1^n, 1^{|C|}, \mathsf{si}\mathcal{O}(1^n, C)) = 1 \,\right]| \leq \mathsf{negl}(n)$.

Bitansky *et al.* also hint the following weakening of siO (as they do not explicitly assign a name to the primitive, the new name is ours).

**Definition 2.4** [Quasi-strong indistinguishability Obfuscators for Circuits] A quasi-strong indistinguishability obfuscator (quasi-siO, in short) for a circuit family $\mathcal{C}$ is defined analogously to siO except that the strong indistinguishability condition is weakened with the quasi-strong indistinguishability condition that is identical to the former except that it is required that the ensembles of distributions be ensembles of *efficiently samplable* distributions.

**On the existence of siO and quasi-siO.** Bitansky *et al.* [BCKP14a] put forward candidate constructions of siO for $\mathsf{NC}^1$ circuits from variants of semantically secure graded encoding schemes [PST14]. Anyway, as they point out ([BCKP14b], p. 5) "existing candidates of indistinguishability obfuscation for all circuits may also be considered as candidates for siO for all circuits". Motivated by this conjecture, in this paper we assume the existence of siO for all circuits. Moreover, all our results can be weakened assuming only siO for $\mathsf{NC}^1$ circuits. For instance, siO for $\mathsf{NC}^1$ circuits is sufficient to build FE with function privacy for $\mathsf{NC}^1$ circuits and similarly for the other primitives we build. We stress that even constructions of (public-key) FPFE for $\mathsf{NC}^1$ were not known before. All such positive results for siO imply corresponding positive results for quasi-siO. Furthermore, Bitansky *et al.* point out that quasi-siO follows from even a weakening of their notion of semantically secure graded encoding schemes.

## 2.3 Functional Encryption

Functional encryption schemes are encryption schemes for which the owner of the master secret can compute restricted keys, called *tokens*, that allow to compute a *functionality* on the plaintext associated with a ciphertext. We start by defining the notion of a functionality.

**Definition 2.5** [Functionality] A *functionality* $F$ is a function $F : K \times M \to \Sigma$ where $K$ is the *key space*, $M$ is the *message space* and $\Sigma$ is the *output space*.

In this work, we consider the following functionality.

**Definition 2.6** [$\mathsf{Circuit}$ Functionality] The $\mathsf{Circuit}$ functionality has key space $K = \cup_n K_n$ with $K_n$ equals to the set of all polynomial-sized Boolean circuits $C$ with $n$ input and output wires. The message space $M = \cup_n M_n$ with $M_n$ equal to the set $\{0, 1\}^n$. For $k \in K$ and $m \in M$, we have $\mathsf{Circuit}(C, m) = C(m)$. Note that this also implicitly defines the output space $\Sigma \triangleq \cup_n \Sigma_n \triangleq \cup_n \{0, 1\}^n$.

**Definition 2.7** [$\mathsf{NC}^1$ Functionality] Such functionality is defined similarly to the functionality $\mathsf{Circuit}$ except that for any $n \in \mathbb{N}$ the key space $K_n$ equals the set of all circuits in $\mathsf{NC}^1$ with $n$ input and output wires.

The above two definitions can be generalized to any class of Boolean circuits in the obvious way. That is, we will sometimes talk of a class of circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$. In this case, for ease of exposition, we assume that any circuit $C \in \mathcal{C}_n$ has $n$ inputs and output wires, except when explicitly specified. This can be easily generalized to the cost of intoducing slightly more complicated notation.

**Definition 2.8** [$\mathsf{ABECircuit}$ and $\mathsf{ABENC}^1$ Functionality] The $\mathsf{ABECircuit}$ functionality has key space $K = \cup_n K_n$ with $K_n$ equals to the set of all polynomial-sized Boolean circuits $C$ with $n$ input and output wires. The message space $X = M \times I = \cup_n (M_n \times I_n)$ where $M$ is the *payload* space and $I$ is the *index* space, with both $M_n$ and $I_n$ equals to the set $\{0, 1\}^n$. For $k \in K$ and

$x = (m, \mathsf{ind}) \in M$, we have $\mathsf{ABECircuit}(C, (m, \mathsf{ind})) = m$ if $C(\mathsf{ind}) = 1$ or $\perp$ otherwise. Note that here we assume that instead the circuits have one bit output. Note that this also implicitly defines the output space $\Sigma$ in the obvious way. Analogously we define $\mathsf{ABENC}^1$ in the obvious way setting for any $n \in \mathbb{N}$ the key space $K_n$ equals to the set of all circuits in $\mathsf{NC}^1$ with $n$ input and output wires.

**Definition 2.9** [Functional Encryption Scheme for Circuit] A *functional encryption* scheme for the Circuit functionality defined over $(K_n, M_n)$ is a tuple $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of 4 algorithms with the following syntax:

$\mathsf{Setup}(1^\lambda, 1^n)$: on input the security parameter $\lambda$ and the length $n$ of the Boolean input supported by the scheme, outputs *public* and *master secret* keys $(\mathsf{Pk}, \mathsf{Msk})$;

$\mathsf{KeyGen}(\mathsf{Msk}, k)$: on input a master secret key Msk and *n-input Boolean circuit* $C \in K_n$, outputs *token* $\mathtt{Tok}_C$;

$\mathsf{Enc}(\mathsf{Pk}, m)$: on input public key Pk and *n-bit Boolean string* $m \in M_n$, outputs *ciphertext* Ct;

$\mathsf{Dec}(\mathsf{Pk}, \mathtt{Tok}_k, \mathsf{Ct})$: outputs a string $y$.

**Correctness.** We require that for all $n$ for all $C \in K_n$ and $m \in M_n$, and for all $(\mathsf{Pk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^n)$, $\mathtt{Tok}_C \leftarrow \mathsf{KeyGen}(\mathsf{Msk}, C)$ and $\mathsf{Ct} \leftarrow \mathsf{Enc}(\mathsf{Pk}, x)$, then

$$\mathsf{Dec}(\mathsf{Pk}, \mathtt{Tok}_C, \mathsf{Ct}) = \mathsf{Circuit}(C, x) = C(x)$$

with probability $1 - \mathsf{negl}(\lambda)$ for some negligible function $\mathsf{negl}(\cdot)$.

The above definition of a FE for Circuit extends easily to other functionalities. Thus, in general we speak about of a FE for functionality $F$ (e.g., $\mathsf{NC}^1$ circuits, or an arbitrary class of circuits $\mathcal{C}$, etc.) with the obvious meaning. Of particular interest is the following special type of FE.

**Definition 2.10** [Attribute-based Encryption][GPSW06] We denote by Attribute-based Encryption (ABE, in short) for Circuit (resp. for $\mathsf{NC}^1$) a FE for functionality ABECircuit (resp. functionality $\mathsf{ABENC}^1$). More generally, we speak about an ABE for functionality $F$ (see Boneh *et al.* [BSW11] for formal definitions) with the obvious meaning.

In Appendix A we recall the standard definition of indistinguishability-security for functional encryption.

## 2.4 Function Private Functional Encryption

A Function Private Functional Encryption (FPFE, in short) scheme is a FE scheme satisfying IND-Security and the following additional function privacy security notion.

**Indistinguishability-based function privacy security.** The indistinguishability-based function privacy notion of security for a functional encryption scheme
$\mathsf{FPFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ is formalized by means of the following game $\mathsf{INDFP}^{\mathsf{FPFE}}_{\mathcal{A}}$ between an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and a *challenger* $\mathcal{C}$. Below, we present the definition for only one function; it is easy to see the definition extends naturally for multiple functions (see remark 2.13).

$\mathsf{INDFP}^{\mathsf{FPFE}}_{\mathcal{A}}(1^\lambda)$

1. $\mathcal{C}$ generates $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and runs $\mathcal{A}_0$ on input $\mathsf{Mpk}$;
2. $\mathcal{A}_0$ submits queries for Boolean circuits $C_i \in \mathcal{C}_\lambda$ for $i = 1, \ldots, q_1$ and, for each such query, $\mathcal{C}$ computes $\mathtt{Tok}_i = \mathsf{KeyGen}(\mathsf{Msk}, C_i)$ and sends it to $\mathcal{A}_0$.

   When $\mathcal{A}_0$ stops, it outputs two *challenge distributions* $D_{0,\lambda}, D_{1,\lambda}$ over $\mathcal{C}_\lambda$ and its internal state $\mathtt{st}$.
3. $\mathcal{C}$ picks $b \in \{0, 1\}$ at random, picks a circuit $C$ according to distribution $D_{b,\lambda}$, and computes the *challenge token* $\mathsf{Tok} = \mathsf{KeyGen}(\mathsf{Msk}, C)$ and sends $\mathsf{Tok}$ to $\mathcal{A}_1$ that resumes its computation from state $\mathtt{st}$.
4. $\mathcal{A}_1$ submits queries for circuits $C_i \in \mathcal{C}_\lambda$ for $i = q_1 + 1, \ldots, q$ and, for each such query, $\mathcal{C}$ computes $\mathtt{Tok}_i = \mathsf{KeyGen}(\mathsf{Msk}, C_i)$ and sends it to $\mathcal{A}_1$.
5. When $\mathcal{A}_1$ stops, it outputs $b'$.
6. **Output:** if $b = b'$ then output 1 else output 0.

The advantage of adversary $\mathcal{A}$ in the above game is defined as

$$\mathsf{Adv}^{\mathsf{FPFE},\mathsf{INDFP}}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathsf{INDFP}^{\mathsf{FPFE}}_{\mathcal{A}}(1^\lambda) = 1] - 1/2|.$$

Note that we did not put any non-trivial constraint on the above game. In fact, any PPT could trivially win in it. As in Agrawal *et al.* we need to restrict the class of adversaries to what are called *legitimate function privacy* adversaries.

**Definition 2.11** A non-uniform family of PPT algorithms $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ is called a *legitimate function privacy* adversary against a FPFE scheme for a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda \in \mathbb{N}}$ if all pairs of distributions $D_{0,\lambda}$ and $D_{1,\lambda}$ output by $\mathcal{A}_\lambda$ in the above game for security parameter $\lambda$ are such that $D_0 \stackrel{\triangle}{=} \{D_{0,\lambda}\}_{\lambda \in \mathbb{N}}$ and $D_1 \stackrel{\triangle}{=} \{D_{1,\lambda}\}_{\lambda \in \mathbb{N}}$ are of a pair of ensembles of *efficiently samplable* feasible entropy distributions[5] over a circuit class $\mathcal{C}' = \{\mathcal{C}'_\lambda\}_{\lambda \in \mathbb{N}}$ where for any $\lambda \in \mathbb{N}, \mathcal{C}'_\lambda$ contains circuits of the *same* size.

**Definition 2.12** We say that $\mathsf{FPFE}$ is *indistinguishability function private secure* ($\mathsf{INDFP}$-Secure, for short) if every legitimate function privacy adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ have at most negligible advantage in the above game.[6]

**Remark 2.13** We defined the security for a challenge consisting of only one function. It is easy to observe that this one-function definition implies a corresponding many-functions definition. Nevertheless, note that this holds because we assume that the distributions output by the adversary be efficiently samplable, that is a natural requirement in this context. For a different definition where the adversary is allowed to output general distributions, this implication could not hold.

[5] Note that the adversary is randomized so that the distributions could depend on its randomness. Thus, the interpretation here is that *all* pairs of sequences $(D_{0,\lambda}, D_{1,\lambda})_{\lambda \in \mathbb{N}}$, formed putting for any $\lambda$ some pair of distributions $D_{0,\lambda}$ and $D_{1,\lambda}$ that it is a *possible* (i.e., such that the adversary outputs them with non-zero probability) output of the adversary in the experiment for parameter $\lambda$, is a pair of ensembles of efficiently samplable feasible entropy distributions. Note that Agrawal *et al.* do not explicitly expand on this detail. Same considerations hold for later definition of FAS legitimate adversaries.

[6] Hereafter, we say that a family of algorithms $\mathcal{B} = \{\mathcal{B}_n\}_{n \in \mathbb{N}}$ has negligible advantage in a experiment if there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $n \in \mathbb{N}$ the advantage of $\mathcal{B}_n$ in the experiment is at most $\mathsf{negl}(n)$.

### 2.5 Functional Encryption for Randomized Functionalities

Goyal *et al.* [GJKS15] introduced the concept of FE for randomized functionalities. Like in Komargodski *et al.* [KSY15] in this paper we do not take in account the problem of dishonest decryptors, as this problem does not arise only in the context of randomized functionalities, and thus we think it goes beyond the scope of our paper. A FE for randomized functionalities (RFE, in short) has the same syntax of a FE scheme for deterministic functionalities, with the obvious change that the functionality takes two inputs, the message and the randomness. We defer to the aforementioned papers for details. In this paper we will focus on the functionality of randomized circuits, both randomized $\mathsf{NC}^1$ circuits and general randomized poly-size circuits, defined in an anologous way to the deterministic case except that such circuits also take a random string as second input. Whereas the syntax of a RFE is almost identical to the deterministic setting, the correctness and the security instead are changed.

**Definition 2.14** [Correctness of RFE] A RFE scheme $\mathsf{RFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for a randomized class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ is correct if for every sufficiently large $\lambda \in \mathbb{N}$, for every polynomial $n = n(\lambda)$, for every sequence of $n$ functions $C_1, \ldots, C_n \in \mathcal{C}_\lambda$, and every sequence of $n$ messages $m_1, \ldots, m_n \in \{0,1\}^n$, the following two distributions are computationally indistinguishable:

1. **Real:** $\{\mathsf{RFE.Eval}(\mathsf{Ct}_i, \mathtt{Tok}_j)\}_{i\in[n],j\in[n]}$, where:

   - $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{RFE.Setup}(1^\lambda)$:
   - $\mathsf{Ct}_i \leftarrow \mathsf{RFE.Enc}(\mathsf{Mpk}, x_i)$ for $i \in [n]$;
   - $\mathtt{Tok}_j \leftarrow \mathsf{RFE.KeyGen}(\mathsf{Msk}, C_j)$ for $j \in [n]$;

2. **Ideal:** $\{C_j(x_i; r_{i,j})\}$, where $r_{i,j} \leftarrow \{0,1\}^\lambda$.

In the above lines, the values $r_{i,j}$'s represent the randomness used by the circuits. For ease of notation, we assume that such random strings have length $\lambda$ though it is easy to generalize it to the cost of introducing a slightly more complicated notation.

**Indistinguishability-based security for RFE.** As our formalization of security we choose what Goyal *et al.* call "security against key queries after public-key" except that, as said before, we do not take in account the problem of dishonest decryptors. The indistinguishability-based security for a RFE scheme
$\mathsf{RFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for a class of circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_\lambda$ is formalized by means of the following game $\mathsf{INDRFE}^{\mathsf{RFE}}_{\mathcal{A}}$ between an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and a *challenger* $\mathcal{C}$. Below, we present the definition for only one message; it is easy to see the definition can be extended naturally for multiple messages and, as observed by Goyal *et al.*, a RFE scheme satisfying the definition with single message also satisfies the definition with multiple messages.

```
$\mathsf{INDRFE}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda)$
    1. $(x_0, x_1, \mathtt{st}) \leftarrow \mathcal{A}_1(1^\lambda)$;
    2. $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{RFE.Setup}(1^\lambda)$:
    3. $b \leftarrow \{0, 1\}$;
    4. $\mathsf{Ct}^\star \leftarrow \mathsf{Enc}(\mathsf{Mpk}, x_b)$;
    5. $b' \leftarrow \mathcal{A}_2^{\mathsf{RFE.KeyGen}(\mathsf{Msk},\cdot)}(\mathsf{Mpk}, \mathsf{Ct}^\star, \mathtt{st})$;
    6. Output: if $x_0$ and $x_0, x_1 \in \{0, 1\}^\lambda$, the queries are for circuits in $\mathcal{C}_n$ and $b = b'$
       then output 1 else output 0.
```

A generic PPT adversary could easily win in the above game. Thus, as in Goyal *et al.* we need to put a restriction on the adversary to make the security requirement non-trivial.

**Definition 2.15** We say that an algorithm $\mathcal{A}$ is a *legitimate RFE* adversary if in the above experiment the following holds. Let $\mathsf{Mpk}^\lambda$ be any public-key given output in the experiment for parameter $\lambda$ with non-zero probability, let $\mathtt{st}^\lambda$ be any state output by $\mathcal{A}_1$ during the experiment for parameter $\lambda$ with non-zero probability, and let $S^\lambda = (C_i)_{i \in [q]}$ denote any list of $q(\lambda)$ oracle queries consisting of randomized circuits made, with non-zero probability, by $\mathcal{A}_2$ to its oracle during the experiment for parameter $\lambda$ on input $\mathsf{Mpk}^\lambda$ and $\mathtt{st}^\lambda$. Then, the two ensembles of distributions $(\mathsf{Mpk}^\lambda, \mathtt{st}^\lambda, (C_i(x_0; r))_{i \in S^\lambda})_{\lambda \in \mathbb{N}}$ and $(\mathsf{Mpk}^\lambda, \mathtt{st}^\lambda, (C_i(x_1; r))_{i \in S^\lambda})_{\lambda \in \mathbb{N}}$ are statistically indistinguishable (i.e., there exists a negligible function $\mathsf{negl}(\cdot)$ such that for any randomized function $f$ the probability that $f$ can distinguish them is $\mathsf{negl}(\lambda)$ where the probability is taken over the choices of $r \in \{0, 1\}^\lambda$).

The advantage of adversary $\mathcal{A}$ in the above game is defined as

$$\mathsf{Adv}^{\mathsf{RFE},\mathsf{INDRFE}}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathsf{INDRFE}^{\mathsf{RFE}}_{\mathcal{A}}(1^\lambda) = 1] - 1/2|.$$

**Definition 2.16** We say that $\mathsf{RFE}$ is *indistinguishability secure* ($\mathsf{INDRFE}$-Secure, for short) if all non-uniform families of PPT legitimate RFE adversaries $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ have at most negligible advantage in the above game.

## 2.6 Functional Anonymous Signature

**Definition 2.17** [Functional Anonymous Signature Schemes] A *functional anonymous signature* (FAS, in short) scheme for a class of circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n \in \mathbb{N}}$, where for each $n \in \mathbb{N}$ and any $C \in \mathcal{C}_n$ has $n$ input wires and one output wire, is a tuple of PPT algorithms $\mathsf{FAS} = (\mathsf{FAS.Setup}, \mathsf{FAS.Sign}, \mathsf{FAS.Verify})$ with the following syntax:

1. $\mathsf{FAS.Setup}(1^\lambda)$ outputs a pair consisting of a *verification* and *signing* key $(\mathsf{vk}, \mathsf{sk})$ for *security parameter* $\lambda$.

2. $\mathsf{FAS.Sign}(\mathsf{sk}, \mathsf{C})$, on input a signing key $\mathsf{sk}$ for security parameter $\lambda$, and a Boolean circuit $C \in C_\lambda$ outputs a *signature* $\sigma$ of it.

3. $\mathsf{FAS.Verify}(\mathsf{vk}, \sigma, \mathsf{m})$, on input verification key $\mathsf{vk}$ for security parameter $\lambda$, a signature $\sigma$ for some (possibly unknown) circuit $C \in \mathcal{C}_\lambda$, and message $m \in \{0, 1\}^\lambda$ outputs 1 or $\perp$;

We require the following correctness requirement on a FAS:

- (Correctness): For all security parameter $\lambda$, all circuits $C \in \mathcal{C}_n$, all $m \in \{0,1\}^\lambda$ such that $C(m) = 1$, there exists a negligible probability $\mathsf{negl}(\cdot)$ sucht that it holds:

$$\Pr\Big[\mathsf{Verify}(\mathsf{vk}, \mathsf{m}, \sigma) = 1 : (\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda), \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{C})\Big] \leq 1 - \mathsf{negl}(\lambda).$$

For the security, we require the two following security properties:

- (Functional Unforgeability): Our notion of unforgeability, that suits for most applicatios of FAS, does not consider as valid the forgery of a circuit more restricted than a circuit for which a signature was seen. Formally, we require that any non-uniform family of PPT algorithms $\mathcal{A}$ wins in the following game with probability negligible in $\lambda$:

  1. $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{FAS.Setup}(1^\lambda)$;
  2. $(C, \sigma) \leftarrow \mathcal{A}^{\mathsf{FAS.Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk})$;
  3. **Winning Condition:** $\mathcal{A}$ wins iff $C \in \mathcal{C}_\lambda$ and there exists $m \in \{0,1\}^\lambda$ such that $\mathsf{FAS.Verify}(\mathsf{vk}, \mathsf{m}, \sigma) = 1$ and for any circuit $C'$ for which $\mathcal{A}$ asked an oracle query it holds that $C'(m) = 0$.

  Later, we will show how to make a FAS unforgeable according to the classical notion just adding a traditional signature scheme on the top.

- (Anonymity): Consider the following game between a challenger and an adversary $\mathcal{A}$.

  1. $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{FAS.Setup}(1^\lambda)$;
  2. $(D_0, D_1, \mathtt{st}) \leftarrow \mathcal{A}^{\mathsf{FAS.Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk})$;
  3. $b \leftarrow \{0, 1\}$;
  4. $C \leftarrow D_b$:
  5. $\sigma \leftarrow \mathsf{FAS.Sign}(\mathsf{sk}, \mathsf{C})$;
  6. $b' = \mathcal{A}(\mathtt{st}, \sigma)$;
  7. **Output:** $\mathcal{A}$ wins iff $b' = b$.

  A non-uniform family of PPT algorithms $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{N}}$ is called a *legitimate FAS* adversary against a FAS scheme for a class of Boolean circuits $\mathcal{C} = \{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if all pairs of distributions $D_{0,\lambda}$ and $D_{1,\lambda}$ output by $\mathcal{A}$ in the above game for security parameter $\lambda$ are such that $D_0 \overset{\triangle}{=} \{D_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $D_1 \overset{\triangle}{=} \{D_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ are a pair of ensembles of *efficiently samplable* feasible entropy distributions over a circuit class $\mathcal{C}' = \{\mathcal{C}'_\lambda\}_{\lambda\in\mathbb{N}} \subset \mathcal{C}$ where for any $\lambda \in \mathbb{N}, \mathcal{C}'_\lambda$ contains circuits of the *same* size. We require that all legitimate FAS adversaries can win in the above game with probability at most negligible in $\lambda$.

# 3 Construction of FPFE from quasi-siO

**Definition 3.1** [quasi-siO-Based Construction]
Let $\mathsf{qsi}\mathcal{O}$ be a quasi-siO and $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.Enc}, \mathsf{FE.KeyGen}, \mathsf{FE.Eval})$ be a FE scheme, both for a class of circuits $\mathcal{C}$.

We define a FPFE functional encryption scheme
$\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}] = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for the class of circuits $\mathcal{C}$ as follows.

- $\mathsf{Setup}(1^\lambda)$: output the public-key $\mathsf{Mpk}$ and master secret-key $\mathsf{Msk}$ computed, respectively, as the public-key and the master secret-key output by $\mathsf{FE.Setup}(1^\lambda)$.
- $\mathsf{Enc}(\mathsf{Mpk}, m)$: output $\mathsf{Ct} \leftarrow \mathsf{FE.Enc}(\mathsf{Mpk}, m)$.
- $\mathsf{KeyGen}(\mathsf{Msk}, C$: output the token $\mathsf{FE.KeyGen}(\mathsf{Msk}, \mathsf{qsi}\mathcal{O}(C))$.
- $\mathsf{Eval}(\mathsf{Mpk}, \mathsf{Ct}, \mathtt{Tok})$: output $\mathsf{FE.Eval}(\mathsf{Mpk}, \mathsf{Ct}, \mathtt{Tok})$.

**Correctness.** It is easy to see that the scheme satisfies correctness assuming the correctness of $\mathsf{qsi}\mathcal{O}$ and $\mathsf{FE}$.
**IND-Security.** It is trivial to observe that the following theorem holds.

**Theorem 3.2** *If* $\mathsf{FE}$ *is* IND-*Secure then* $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}]$ *is* IND-*Secure.*

**INDFP-Security.**

**Theorem 3.3** *If* $\mathsf{qsi}\mathcal{O}$ *is a quasi-siO then* $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}]$ *is* INDFP-*Secure.*

**Proof:** Suppose that there exists a legitimate function privacy adversaries $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ breaking the INDFP-Security of $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}]$. Specifically, suppose that there exists a non-negligible function $p(\cdot)$ such that for any $n \in \mathbb{N}$, $\mathcal{A}_n$ wins in the INDFP-Security parameterized by $n$ with advantage $\geq p(n)$. Thus, by an averaging argument, for any $n \in \mathbb{N}$ there exist two distributions $D_{0,n}$ and $D_{1,n}$ and random strings $r_1, r_2 \in \{0,1\}^\star$ (to be defined later) such that in the the security experiment (for parameter $n$) executed with random strings $r_1, r_2$, $\mathcal{A}_n$ outputs such distributions as challenge distributions with non-zero probability and under the occurrence of such event $\mathcal{A}_n$ has advantage $p(n)$. Precisely, $r_1$ is used to compute the public-key and the master secret-key with which the token queries can be answered (w.l.o.g., we can assume that $\mathsf{KeyGen}$ is deterministic) and $r_2$ is used to run the adversary until the challenge query (that is, after the challenge query other randomness will be used and $r_1$ and $r_2$ determine the behavior of $\mathcal{A}_n$ until that point but not after.[7]). Then, from the fact that $\mathcal{A}$ is a legitimate function privacy adversary it follows that the ensembles $D_0 = \{D_{0,n}\}_{n\in\mathbb{N}}$ and $D_1 = \{D_{1,n}\}_{n\in\mathbb{N}}$ are a pair of ensembles of feasible entropy distributions and thus it is straightforward to construct a family of non-uniform distinguishers $\mathcal{D} = \{\mathcal{D}_n\}$ breaking the security of qsiO as follows. Specifically, $\mathcal{D}_n$ has embedded the random strings $r_1, r_2$ (that have size polynomial in $n$) and takes as input the obfuscated circuit $C'$ that is a computed as $\mathsf{qsi}\mathcal{O}(C)$ where the circuit $C$ is drawn from either $D_{0,n}$ or $D_{1,n}$. $\mathcal{D}_n$ runs the setup of $\mathsf{FE}$ with security parameter $n$ and randomness $r_1$ to

[7] Recall that there are two ways to define probabilistic algorithms. One is to feed them with a random string, and one is to give them access to an oracle that returns random bits. Here we can adopt the latter convention and in this case we mean that the oracle uses the bits of $r_2$ to answer the queries until the challenge phase, and after that the oracle returns uniformly and independently chosen bits. Furthermore, note that $r_2$ is not used to answer the challenge query: indeed, as it will be specified later, the randomness used to answer it is chosen by the challenger of quasi-siO and thus it will be not known to the distinguisher.

get the public-key $\mathsf{Mpk}$ and master secret-key $\mathsf{Msk}$ of $\mathsf{FE}$. Then, $\mathcal{D}_n$ runs $\mathcal{A}_n$ with randomness $r_2$ on input $\mathsf{Mpk}$ and answers the $\mathcal{A}_n$'s queries using $\mathsf{Msk}$. Then, by construction of $r_1$ and $r_2$, $\mathcal{A}_n$ outputs as challenge distributions $D_{0,n}$ and $D_{1,n}$. $\mathcal{D}_n$ answers the challenge query returning to $\mathcal{A}_n$ the token $\mathsf{FE.KeyGen}(\mathsf{Msk}, C')$ and then continues the execution of $\mathcal{A}_n$ as before. At the end $\mathcal{D}_n$ outputs what $\mathcal{A}_n$ outputs. It is easy to see that the advantage of $\mathcal{D}_n$ in distinguishing whether the input was an obfuscation of a circuit drawn from $D_{0,n}$ or $D_{1,n}$ is $p(n)$ (note here that the probability is also taken over the choices of the randomness used to compute $C'$ that is not known to $\mathcal{D}_n$). Then, we conclude that $\mathcal{D}$ along with the ensembles of distributions $D_0 = \{D_{0,n}\}_{n\in\mathbb{N}}$ and $D_1 = \{D_{1,n}\}_{n\in\mathbb{N}}$ contradicts the security of $\mathsf{qsiO}$. ▌

**Extensions to multi-inputs FE with function privacy.** A nice property enjoyed by our construction is that it easily extends to the multi-inputs setting [GGG+14]. That is, if in the above construction we replace $\mathsf{FE}$ with a multi-inputs FE, the resulting scheme is a function private multi-inputs functional encryption scheme (where the security is naturally generalized to the multi-inputs setting).

## 4 Construction of FAS from FPABE

**Overview.** The construction extends the Naor's transformation from IBE to (traditional) signature schemes. Specifically a token for a circuit $C$ computed with the ABE system acts as a signature for $C$. The security of the ABE system guarantees the unforgeability as required by FAS. In fact, no adversary, given a token for circuit $C$ can produce another token, and thus a valid forgery, for another circuit that would enable to distinguish the encryption of two ciphertexts computed with an attribute $x$ such that $C(x) = 0$. If in addition the ABE system satisfies function privacy, the resulting FAS scheme is anonymous as well.

**Definition 4.1** [FPFE-Based Construction] Let $\mathsf{FPABE} = (\mathsf{FPABE.Setup}, \mathsf{FPABE.Enc}, \mathsf{FPABE.KeyGen}, \mathsf{FPABE.Eval})$ be a FPABE scheme for the class of Boolean circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$.

We define a FAS scheme
$\mathsf{FAS}[\mathsf{FPABE}] = (\mathsf{FAS.Setup}, \mathsf{FAS.Sign}, \mathsf{FAS.Verify})$ for $\mathcal{C}$ as follows.

- $\mathsf{FAS.Setup}(1^\lambda)$: set verification key $\mathsf{vk}$ and signing key $\mathsf{sk}$ to be respectively the public-key and the master secret-key output by the setup of $\mathsf{FPABE}$ .
- $\mathsf{FAS.Sign}(\mathsf{sk}, \mathsf{C})$: output $\sigma \leftarrow \mathsf{FPABE.KeyGen}(\mathsf{sk}, \mathsf{C})$.
- $\mathsf{FAS.Verify}(\mathsf{vk}, \sigma, \mathsf{x})$: choose random value $r \leftarrow \{0,1\}^\lambda$, encrypt $\mathsf{Ct} \leftarrow \mathsf{FPABE.Enc}(\mathsf{vk}, (r, x))$ and compute $r' \leftarrow \mathsf{FPABE.Eval}(\mathsf{vk}, \mathsf{Ct}, \sigma)$. If $r' = r$ then output 1 otherwise output $\perp$.

**Correctness.** It is easy to see that the scheme satisfies correctness assuming the correctness of $\mathsf{FPABE}$.

**Functionally Unforgeability.** It is trivial to observe that $\mathsf{FAS}[\mathsf{FPABE}]$ is functionally unforgeable. In fact an adversary outputting a forgery that satisfies the winning condition of functional unforgeability, is a valid adversary against the security of $\mathsf{FPABE}$ and thus as in the Naor's transformation the forgery can be used to break the security of $\mathsf{FPABE}$. Thus, the following theorem holds.

**Theorem 4.2** *If* $\mathsf{FPABE}$ *is* $\mathsf{IND}$*-Secure then* $\mathsf{FAS}[\mathsf{FPABE}]$ *is unforgeable.*

**Anonymity.**

**Theorem 4.3** *If* $\mathsf{FPABE}$ *is* $\mathsf{INDFP}$*-Secure then* $\mathsf{FAS[FPABE]}$ *is anonymous.*

Proof Sketch. The proof is almost identical to that of theorem 3.3, thus we omit full details. Suppose that there exists a family of non-uniform PPT adversaries $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ breaking the anonymity of $\mathsf{FAS[FPABE]}$. Then, it is easy to construct a family of non-uniform PPT adversaries $\mathcal{B} = \{\mathcal{B}_n\}_{n\in\mathbb{N}}$ breaking the security of $\mathsf{FPABE}$. Being $\mathcal{A}$ a legitimate FAS adversary, we can construct $\mathcal{B}_n$ identical to the distinguisher $\mathcal{D}_n$ in the proof of theorem 3.3 except in the way that $\mathcal{B}_n$ has to simulates the view to $\mathcal{A}$ and construct the challenge. This is also straightforward. Then, we conclude that $\mathcal{B}$ contradicts the security of $\mathsf{qsiO}$. □

**Adding unforgeability to FAS.** It is easy to make the above scheme even secure according to the traditional notion of unforgeability. It is sufficient to use a traditional unforgeable signature scheme and signing the token with such scheme. The resulting scheme will be unforgeable (according to the traditional notion) as well.

## 5 Construction of RFE from FPFE

**Definition 5.1** [FPFE-Based Construction]
Let $\mathsf{F} = (\mathsf{F.Key}, \mathsf{F.Puncture}, \mathsf{F.Eval})$ be a puncturable pseudorandom function and $\mathsf{FPFE} = (\mathsf{FPFE.Setup}, \mathsf{FPFE.Enc}, \mathsf{FPFE.KeyGen}, \mathsf{FPFE.Eval})$ be a FPFE scheme, both for a sufficiently expressive class of (deterministic) Boolean circuits $\mathcal{C}'$ to be specified later in Remark 5.10).

We define a RFE functional encryption scheme
$\mathsf{RFE[F, FPFE]} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for the class of randomized Boolean circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ induced by $\mathcal{C}'$[8] as follows.

- $\mathsf{Setup}(1^\lambda)$: generate the public-key $\mathsf{Mpk}$ and the master secret-key $\mathsf{Msk}$ computed, respectively, as the public-key and the master secret-key output by $\mathsf{FPFE.Setup}(1^\lambda)$.
- $\mathsf{Enc}(\mathsf{Mpk}, m)$: output $\mathsf{Ct} \leftarrow \mathsf{FPFE.Enc}(\mathsf{Mpk}, m)$.
- $\mathsf{KeyGen}(\mathsf{Msk}, C)$: on input a master secret-key $\mathsf{Msk}$ for security parameter $\lambda$, a Boolean randomized circuit $C \in \mathcal{C}_\lambda$ with input of length $n$ and randomness of length $n$, compute $k \leftarrow \mathsf{F.Key}(1^\lambda)$ and output the token $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, C[k]))$ for the following deterministic Boolean circuit $C[k] \in \mathcal{C}'_{2\lambda}$.

```
Circuit $C[k](m)$
1. Pad with circuits $U[C, k, m_0, m_1, s_0, s_1]$ and $U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1]$;
2.     return $C(m||\mathsf{F.Eval(k, m)})$.
```

- $\mathsf{Eval}(\mathsf{Mpk}, \mathsf{Ct}, \mathtt{Tok})$: output $\mathsf{FPFE.Eval}(\mathsf{Mpk}, \mathsf{Ct}, \mathtt{Tok})$.

**Correctness.** It is easy to see that the scheme satisfies correctness assuming the correctness of $\mathsf{FPFE}$ and the pseudorandomness of $\mathsf{F}$.

[8] Here, we mean that for any $n \in \mathbb{N}$ and for any randomized circuit $C \in \mathcal{C}_n$ with inputs of length $n$ and randomness of length $n$ we define $C$ to be the corresponding deterministic circuit $C' \in \mathcal{C}'_{2n}$ with inputs of length $2n$ defined in the obvious way (i.e., defined so that the two circuits when viewed as circuits with inputs of length $2n$ have the same description).

**Security reduction.** We reduce the security of our RFE scheme to that of the underlying primitives (FPFE and puncturable pseudorandom functions) via a series of hybrid experiments against a PPT legitimate RFE adversary $\mathcal{A}$ attacking the $\mathsf{INDRFE}$-Security of $\mathsf{RFE}[\mathsf{F},\mathsf{FPFE}]$ (here, for sake of simplicity we assume uniform adversaries). Recall that in the $\mathsf{INDRFE}$-Security experiment the adversary $\mathcal{A}$ selects as challenges a pair of messages $(m_0, m_1)$.

- $H_0$. This corresponds to the $\mathsf{INDRFE}$-Security game in which the challenge ciphertext encrypts the message $m_0$.

- $H_1$. This experiment is identical to $H_0$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k, m_0, m_1, s_0, s_1])$ where $s_b = \mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b})$ for $b \in \{0, 1\}$ and $U[C, k, m_0, m_1, s_0, s_1]$ is the following deterministic circuit:

```
**Circuit** $U[C, k, m_0, m_1, s_0, s_1](m)$
Pad with circuits $C[k]$ and $U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1]$;
if $m = m_0$ return $C(m||s_0)$;
else if $m = m_1$ return $C(m||s_1)$;
    otherwise return $C(m||\mathsf{F.Eval}(\mathsf{k}, \mathsf{m}))$.
```

**Claim 5.2** Indistinguishability of $H_1$ from $H_0$. First, we assume that the adversary asks only one token query. The general case follows from a standard hybrid argument. Note that the two circuits $C[k]$ and $U[C, k, m_0, m_1, s_0, s_1]$ compute the same function. In fact, on input $m = m_b$ for $b \in \{0, 1\}$ the first circuit computes $C(m_b||\mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b}))$ and the second circuit computes $C(m_b||s_b)$ that, by construction of $s_b$, equals $C(m_b||\mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b}))$. For any other input $m \neq m_0, m_1$, by construction, the two circuits output the same value as well. Then, consider the two ensembles (parameterized by the security parameter $\lambda$) of distributions $D_0$ and $D_1$ defined so to output with probability 1, respectively, the circuit $C[k]$ and the circuit $U[C, k, m_0, m_1, s_0, s_1]$. It is straightforward to notice that such pair of ensembles of distributions is feasible, thus the claim follows from the $\mathsf{INDFP}$-Security of $\mathsf{FPFE}$.

- $H_2$. This experiment is identical to $H_1$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1])$ where $s_b = \mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b})$ for $b \in \{0, 1\}$ as before but $k(\{m_0, m_1\}) = \mathsf{F.Puncture}(\mathsf{k}, \{\mathsf{m_0}, \mathsf{m_1}\})$ and $U[C, k(\{m_0, m_1\}, m_0, m_1, s_0, s_1]$ is identical to $U[C, k, m_0, m_1, s_0, s_1]$ except for the constant $k(\{m_0, m_1\}$ instead of $k$.

**Claim 5.3** Indistinguishability of $H_2$ from $H_1$. First, we assume that the adversary asks only one token query. The general case follows from a standard hybrid argument. Note that the two circuits $U[C, k, m_0, m_1, s_0, s_1]$ and $U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1]$ differ only for the constant values $k$ and $k(\{m_0, m_1\})$. By the fact that $\mathsf{F}$ preserves the functionality at points different from the punctured points, and by construction of the two circuits and of $s_0$ and $s_1$, the two circuits compute the same function. Thus, as argued above, the claim follows from the $\mathsf{INDFP}$-Security of $\mathsf{FPFE}$.

- $H_3$. This experiment is identical to $H_2$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1])$ where $s_0$ and $s_1$ are randomly and independently chosen in $\{0,1\}^{m(\lambda)}$, and $k = \mathsf{F.Key}(1^\lambda)$ and $k(\{m_0, m_1\}) = \mathsf{F.Puncture}(\mathsf{k}, \{\mathsf{m_0}, \mathsf{m_1}\})$ are as in the previous experiments.

**Claim 5.4** Indistinguishability of $H_3$ from $H_2$. First, we assume that the adversary asks only one token query. The general case follows from a standard hybrid argument. The indistinguishability of the two experiments follows from the pseudorandomness of $\mathsf{F}$ at the punctured points $m_0$ and $m_1$.

- $H_4$. This experiment is identical to $H_3$ except that the challenge ciphertext is computed as encryption of $m_1$.

  **Claim 5.5** Indistinguishability of $H_4$ from $H_3$. First, we notice what follows. Any token for randomized circuit $C$ for which $\mathcal{A}$ asked a query is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1])$ where $s_0$ and $s_1$ are randomly and independently chosen in $\{0,1\}^{m(\lambda)}$ and $k(\{m_0, m_1\}) = \mathsf{F.Puncture}(\mathsf{k}, \{\mathsf{m_0}, \mathsf{m_1}\})$ (for $k$ computed as $k \leftarrow \mathsf{F.Key}(1^\lambda)$). By construction we have $U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1](m_0) \stackrel{\triangle}{=} C(m_0; s_0)$ and $U[C, k(\{m_0, m_1\}), m_0, m_1, s](m_1) \stackrel{\triangle}{=} C(m_1; s_1)$. By the requirement that $\mathcal{A}$ is a legitimate RFE adversary, it follows that $\mathcal{A}$ only asks queries for circuits $C$ such that $C(m_0; s)$ is statistically indistinguishable from $C(m_1; s)$ where the probability is taken over the choices of $s$ and thus the above equations imply that with all except negligible probability over the choices of $s_0$ and $s_1$ in $\{0,1\}^{m(\lambda)}$, $C(m_0; s_0) = C(m_1; s_1)$. Therefore, the indistinguishability of the two experiments follows from the $\mathsf{IND}$-Security of $\mathsf{FPFE}$.

- $H_5$. This experiment is identical to $H_4$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k(\{m_0, m_1\}), m_0, m_1, s_0, s_1])$ where $s_b$ for $b \in \{0,1\}$ is computed as $\mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b})$, and $k = \mathsf{F.Key}(1^\lambda)$ and $k(\{m_0, m_1\}) = \mathsf{F.Puncture}(\mathsf{k}, \{\mathsf{m_0}, \mathsf{m_1}\})$ are as in the previous experiments.

  **Claim 5.6** Indistinguishability of $H_5$ from $H_4$. The indistinguishability of the two experiments is symmetrical to that of $H_3$ from $H_2$.

- $H_6$. This experiment is identical to $H_5$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, U[C, k, m_0, m_1, s_0, s_1])$ where $s_b$ for $b \in \{0,1\}$ is computed as $\mathsf{F.Eval}(\mathsf{k}, \mathsf{m_b})$ and $k = \mathsf{F.Key}(1^\lambda)$ as in the previous experiments.

  **Claim 5.7** Indistinguishability of $H_6$ from $H_5$. The indistinguishability of the two experiments is symmetrical to that of $H_2$ from $H_1$.

- $H_7$. This experiment is identical to $H_6$ except that any token for randomized circuit $C$ is computed as $\mathsf{FPFE.KeyGen}(\mathsf{Msk}, C[k])$ where $k = \mathsf{F.Key}(1^\lambda)$ as in the previous experiments.

  **Claim 5.8** Indistinguishability of $H_7$ from $H_6$. The indistinguishability of the two experiments is symmetrical to that of $H_1$ from $H_0$.

Note that experiments $H_0$ and $H_7$ correspond to the experiments of $\mathsf{INDRFE}$-Security where the challenge encrypts respectively $m_0$ and $m_1$.

Thus, the indistinguishability of the above hybrid experiments implies the following theorem (see also remark 5.10).

**Theorem 5.9** If $\mathsf{FPFE}$ is $\mathsf{IND}$-Secure and $\mathsf{INDFP}$-Secure, and $\mathsf{F}$ is a puncturable pseudorandom function, then $\mathsf{RFE}[\mathsf{F}, \mathsf{FPFE}]$ is $\mathsf{INDRFE}$-Secure.

**Remark 5.10** Note that in order to obtain theorem 5.9, the minimal class of circuits $\mathcal{C}'$ must be sufficiently expressive to contain all circuits that can compute the "transformed" circuits used in the security proof and that can compute $\mathsf{F}$. In particular, assuming that $\mathsf{F}$ can be computed in $\mathsf{NC}^1$ we obtain an RFE scheme for $\mathsf{NC}^1$ from a FPFE scheme for $\mathsf{NC}^1$.

**Extensions to multi-inputs RFE and RFE with function privacy** A nice property enjoyed by our construction is that it easily extends to the multi-inputs setting [GGG$^+$14]. That is, if in the above construction we replace $\mathsf{FPFE}$ with a multi-inputs FPFE, the resulting scheme is a multi-inputs functional encryption scheme for randomized functionalities (where the security is naturally generalized to the multi-inputs setting). Moreover, the above construction preserves function privacy, i.e., $\mathsf{RFE}[\mathsf{F}, \mathsf{FPFE}]$ is function private as well (FPRFE), under the standard notion of $\mathsf{INDFP}$-Security for *deterministic* circuits, i.e., the adversary against function privacy can only ask distributions of deterministic circuits. It seems that our construction could be also proven to satisfy a form of function privacy extended in a natural way to support randomized circuits but we did not investigate the details.

Precisely, we have the following theorem.

**Theorem 5.11** *Assuming the existence of quasi-siO, there exists a selectively indistinguishability secure multi-inputs FE with function privacy (as said before, here we refer to the standard notion of function privacy for deterministic circuits in which the adversary against function privacy can only submit a pair of distributions over* deterministic *circuits) for randomized functionalities. Furthermore, the restriction of selective security can be removed assuming in addition an adaptively indistinguishability secure MIFE.*

**Proof:** This follows from the fact that quasi-siO implies iO that in turn implies selectively indistinguishability secure multi-inputs FE via [GGG$^+$14]. Then, multi-inputs FE combined with quasi-siO implies multi-inputs FE with function privacy that in turn implies multi-inputs FE with function privacy for randomized functionalities. Assuming in addition adaptively indistinguishability secure MIFE it is easy to verify the second part of the theorem. ▌

# 6 Relation between Primitives

It is easy to see that quasi-siO implies iO that in turn is known to imply (along with one-way functions) FE [Wat14]. Thus, quasi-siO implies FPFE. Moreover, FAS can be used to construct a quasi-siO as follows. An obfuscation of circuit $C$ will consist of a signature for $C$ and the verification key of the FAS scheme, and to evaluate the obfuscated circuit on an input $x$, just run the verification algorithm of FAS with input the verification key, the signature and the message $m$. From the anonymity of FAS, such obfuscator is easily seen to be a quasi-siO. Note that this implication does not assume FAS with any kind of unforgeability. Since FPFE implies FPABE, that in turn implies FAS, we have that FAS, FPFE and quasi-siO are *equivalent* primitives (i.e., they imply each other). (Furthermore, these implication would also hold assuming selectively secure variants of FPFE, FPABE and FAS). The equivalence also extends to FPRFE and SPP. One of the key points highlighted by our results is that FPABE implies quasi-siO and thus iO that in turn (assuming in addition one-way functions) implies FE [Wat14], a notable fact that sheds light on the importance and power of function privacy for FE. Indeed, even though ABE is not known to imply FE, our results show that the additional property of function privacy suffices for such scope. In Figure 1 we present relations among the primitives studied or discussed in this paper. Note that we are not aware of any work in the literature that explicitly claims a construction of MIFE with adaptive indistinguishable-security, so in the figure we do not put any implication from some primitive to MIFE.

## 7 Acknowledgments

We thank Nir Bitansky for answering questions concerning siO.

## References

[AAB$^+$13] Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. Function private functional encryption and property preserving encryption : New definitions and positive results. Cryptology ePrint Archive, 2013. `http://eprint.iacr.org/2013/744`.

[AAB$^+$15] Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. On the practical security of inner product functional encryption. In *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, pages 777–798, 2015.

[BCKP14a] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 108–125, 2014.

[BCKP14b] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. IACR Cryptology ePrint Archive, 2014. `http://eprint.iacr.org/2014/554/20140805:181558`. Note that we refer to the version posted on 14 August 2014.

[BDOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, pages 501–519, 2014.

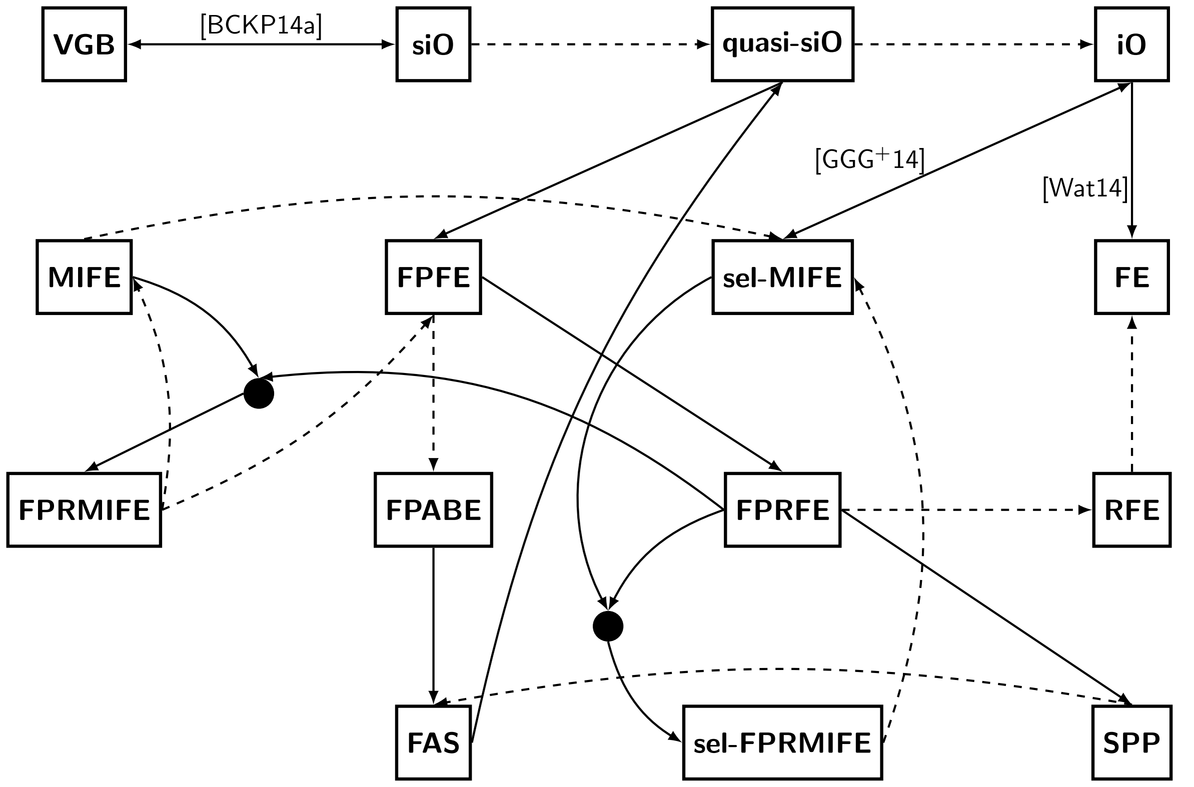

Figure 1: **Relations among primitives studied or discussed in this paper:** A line with arrow going from A to B denotes that it is possible to build B from A and lines are annotated with the work where the implication first appeared, or unlabeled if such implication is discussed in this paper. A line from A to B with arrows at both ends denotes that it is possible to build A from B and vice-versa. A dashed line denotes a trivial implication. Two lines coming respectively from A and B with arrow directed in a circled black box with an outgoing line with arrow directed to box C means that it is possible to build C assuming both A and B (e.g., FPRFE and MIFE imply FPRMIFE). All the implications in the figure are valid if the above primitives are assumed for a sufficiently expressive class of Boolean circuits, for concreteness $\mathsf{NC}^1$. All the primitives related to FE are assumed to be in the public-key model. For the implication from iO to FE as well as for the implication from quasi-siO to FPFE we also need to assume one-way functions. sel-MIFE denotes a selectively indistinguishability-secure MIFE and analogously sel-FPRMIFE. For FE and MIFE we assume adaptive indistinguishability-security. For the security of FPFE, FAS and RFE see Section 2. FPRFE denotes a RFE scheme with a standard form of function privacy for deterministic circuits (see Section 5) and FPRMIFE denotes a FPRFE scheme that is in addition multi-inputs.

[BRS13a] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 461–478. Springer, 2013.

[BRS13b] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 255–275. Springer, 2013.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 306–324, 2015.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Germany.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Berlin, Germany.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Berlin, Germany.

[DFM$^+$10] Alexander W. Dent, Marc Fischlin, Mark Manulis, Martijn Stam, and Dominique Schröder. Confidential signatures and deterministic signcryption. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 462–479, Paris, France, May 26–28, 2010. Springer, Berlin, Germany.

[GGG$^+$14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 578–602. Springer, 2014.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013.

[GJKS15] Vipul Goyal, Abhishek Jain, Venkata Koppula, and Amit Sahai. Functional encryption for randomized functionalities. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 325–351, 2015.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press. Available as Cryptology ePrint Archive Report 2006/309.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[KSY15] Ilan Komargodski, Gil Segev, and Eylon Yogev. Functional encryption for randomized functionalities in the private-key setting from minimal assumptions. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 352–377, 2015.

[LOS$^+$10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 591–608, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.

[PST14] Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation from semantically-secure multilinear encodings. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 500–517, 2014.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Berlin, Germany, March 15–17, 2009.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *Symposium on Theory of Computing Conference, STOC'14, New York, NY, USA, 31 May-3 June, 2014*, pages 475–484, 2014.

[Wat12] Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 218–235, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.

[Wat14] Brent Waters. A punctured programming approach to adaptively secure functional encryption. IACR Cryptology ePrint Archive, 2014. `http://eprint.iacr.org/2014/588`.

# A Security of FE

The indistinguishability-based notion of security for functional encryption scheme $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for functionality $F$ defined over $(K, M)$ is formalized by means of the following game $\mathsf{IND}^{\mathsf{FE}}_{\mathcal{A}}$ between an adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ and a *challenger* $\mathcal{C}$. Below, we present the definition for only one message; it is easy to see the definition extends naturally for multiple messages.

$\mathsf{IND}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda)$

1. $\mathcal{C}$ generates $(\mathsf{Pk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and runs $\mathcal{A}_0$ on input $\mathsf{Pk}$;
2. $\mathcal{A}_0$ submits queries for keys $k_i \in K$ for $i = 1, \ldots, q_1$ and, for each such query, $\mathcal{C}$ computes $\mathtt{Tok}_i = \mathsf{KeyGen}(\mathsf{Msk}, k_i)$ and sends it to $\mathcal{A}_0$.

   When $\mathcal{A}_0$ stops, it outputs two *challenge plaintexts* $m_0, m_1 \in M$ satisfying $|m_0| = |m_1|$ and its internal state $\mathtt{st}$.
3. $\mathcal{C}$ picks $b \in \{0, 1\}$ at random, computes the *challenge ciphertext* $\mathsf{Ct} = \mathsf{Enc}(\mathsf{Pk}, m_b)$ and sends $\mathsf{Ct}$ to $\mathcal{A}_1$ that resumes its computation from state $\mathtt{st}$.
4. $\mathcal{A}_1$ submits queries for keys $k_i \in K$ for $i = q_1 + 1, \ldots, q$ and, for each such query, $\mathcal{C}$ computes $\mathtt{Tok}_i = \mathsf{KeyGen}(\mathsf{Msk}, k_i)$ and sends it to $\mathcal{A}_1$.
5. When $\mathcal{A}_1$ stops, it outputs $b'$.
6. **Output:** if $b = b'$, $m_0$ and $m_1$ are of the same length, and $F(k_i, m_0) = F(k_i, m_1)$ for $i = 1 \ldots, q$, then output 1 else output 0.

The advantage of adversary $\mathcal{A}$ in the above game is defined as

$$\mathsf{Adv}^{\mathsf{FE},\mathsf{IND}}_{\mathcal{A}}(1^\lambda) = |\mathrm{Prob}[\mathsf{IND}^{\mathsf{FE}}_{\mathcal{A}}(1^\lambda) = 1] - 1/2|.$$

**Definition A.1** We say that $\mathsf{FE}$ is *indistinguishably secure* ($\mathsf{IND}$-Ssecure, for short) if all non-uniform families of PPT adversaries $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ have at most negligible advantage in the above game.