On the power of Public-key Functional Encryption with Function Privacy

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

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

Abstract. In the public-key setting, known constructions of *function-private* functional encryption (FPFE) were limited to very restricted classes of functionalities like inner-product [Agrawal *et al.* - PKC 2015]. Moreover, its power has not been well investigated. In this paper, we construct FPFE for general functions and explore its powerful applications (both for general functions and for specific efficient instantiations).
As warmup, 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, Alice can sign a circuit C chosen from some distribution D to get a signature σ and can publish a verification key that allows anybody holding a message m to verify that (1) σ 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 σ hide as much information as possible about C except what can be inferred from knowledge of D.
Then, we show that FPFE can be used to construct in a black-box way functional encryption schemes for randomized functionalities (RFE). Previous constructions of (public-key) RFE relied on iO [Goyal *et al.* - TCC 2015].
As further application, we show that efficient instantiations of FPFE can be used to achieve adaptively-secure CNF/DNF encryption for bounded degree formulae (BoolEnc). Though it was known how to implement BoolEnc from inner-product encryption [Katz *et al.* - EUROCRYPT 2008], as already observed by Katz *et al.* this reduction only works for selective security and completely breaks down for adaptive security. For this application we only need weak assumptions and the resulting adaptively-secure BoolEnc scheme is *efficient.*
Finally, we present a general picture of the relations among all these related primitives. One key observation is that Attribute-based Encryption with function privacy implies FE, a notable fact that sheds light on the importance of the function privacy property for FE.

Keywords. Functional Encryption, Function Privacy, Inner-product Encryption, Obfuscation, Digital Signatures.

1 Introduction

Functional Encryption (FE) [1] is a sophisticated type of encryption that allows to finely control the amount of information that can be revealed by a decryption operation. Progressively, more expressive forms of FE were constructed in a series of works (see, e.g., [2–7]) culminating in the breakthrough of Garg *et al.* [8].

The security notion in these works only take in account the privacy of the *message* but nothing is 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.* [9] for the inner-product functionality [4], subsequently followed by constructions for general functionalities [10]. Boneh *et al.* [11] 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 [12] and the inner-product [13, 14] functionalities.
In the public-key setting, the function can not be hidden completely since the adversary can always try to infer partial information about it using the public key. For this reason, Boneh *et al.* [11] 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 id was chosen from a sufficiently high min-entropy distribution, the adversary should not be able to distinguish the token for id from a token for a uniformly random identity. Agrawal *et al.* [14] consider stronger simulation-based definitions for function privacy but with non-standard simulators (a necessity motivated by broad impossibility results in the area).

It seems that a meaningful simulation-based security notion of public-key function-private functional encryption (FPFE) for some expressive enough class of Boolean circuits would imply virtual black box (VBB) obfuscation for the same class of circuits and thus it seems unachievable even for NC^1 circuits. For such reasons, we stick with the indistinguishability-based (IND-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.* [14] which abstracts the unpredictability property of Boneh *et al.* [11]. 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 out to be very important; indeed it is the key to make such primitives *composable*.

To our knowledge no previous work in literature considered public-key FPFE for more general functionalities, like poly-sized circuits or even NC^1 circuits. This leads to the main questions that we study in this work:

> Can we achieve public-key FPFE for more general functionalities, like at least NC^1 or even all poly-sized circuits, from reasonable assumptions? And what applications and other primitives can we build from it?

Based on the existence of *quasi-siO* proposed by Bitansky, Canetti, Kalai and Paneth [15],[1] we answer *affirmatively* to the first question. The solution we propose is conceptually simple and elegant but we believe that the *key* is in having discovered and identified quasi-siO as the main building block, a relation that was *not* known before in the literature.
Note that quasi-siO is a weakened version of *strong iO* (siO), which guarantees that no efficient adversary can distinguish two *feasible entropy distributions* D_0 or D_1. The weakening lies in the fact that quasi-siO requires the distributions to be *efficiently samplable*.
We answer the second question by mainly demonstrating the implication with respect to functional anonymous signatures, FE for randomized functionalities, and adaptive security for efficient Boolean formulae encryption. Though some of our results can seem basic, this is a due to our recognition of the power of these primitives not studied so far, and some applications we derive from them improve the state of the art in the field or solve known problems. Thus, we deem the simplicity of our approach a positive feature not a shortcoming.
Our results are not only an example of the power and of the applications of FPFE but also and mainly of the power siO/quasi-siO, and in Section 6 we show equivalences between them. We mention that recently the existence of siO was put in contention with the existence of other strong assumptions in [16] and we defer the reader to the discussion after Definition 2 for further details. Anyhow, we point out that all our results can be instantiated assuming only quasi-siO for NC^1 circuits.

1.1 Public-key FPFE based on Quasi-siO

It is worth reminding why existing constructions of FE do not offer any meaningful function privacy. Consider the construction of Garg *et al.* [8] of FE from iO. Therein, the token for a circuit C is an

[1] The name quasi-siO is ours. The authors define a weakening of the their notion of siO (see the following) without explicitly naming it.

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[2], 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. The above analysis motivate us to use siO. If the token was instead a siO of the circuit, it would leak as little 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 FPFE will use the underlying FE scheme FE as a black box and will have identical procedures except that a token for a circuit C will consist of a token of 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 FPFE for C will be a token of 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 little 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.

The modularity of our approach allows to instantiate a FPFE for a class of circuits C assuming only a quasi-siO for the same class of circuits until the class $\mathcal{C}$ is enough expressive, specifically includes at least all NC^1 circuits. Furthermore, it generalizes easily to *multi-inputs FE* (MIFE, in short) [17] 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 2.3 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 scheme 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 FPFE scheme and a token Tok for C of FPFE. 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 Tok and encrypts[3] x to get Ct and evaluates Tok on Ct to get $C(m)$. The correctness of FPFE and its INDFP-Security defined in Section 2.3 imply that such obfuscator is a quasi-siO. This construction also reaffirms that a meaningful simulation-based security notion for FPFE for a class $\mathcal{C}$ would imply VBB obfuscation for $\mathcal{C}$, and thus is unachievable in general. For such reason we stick with an IND-based definition of function privacy.

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

[3] 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 be encrypted in clear. Precisely, according to the definitions from Section 2.3, we only need INDFP-Security and not also IND-Security.

1.2 Functional Anonymous Signatures

As warmup we construct from FPFE a new primitive called *Functional Anonymous Signature* (FAS, in short). Recall that the Naor's transformation[4] allows to transform an identity-based encryption (IBE) scheme [18] in a signature scheme. The idea is that the token for an identity id acts as a signature for it. Such signature can be verified by encrypting the pair (r, 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 IBE, such signature is unforgeable. We generalize this concept to FE and propose what we call FAS. With FAS, a user Alice can sign a Boolean circuit C allowing Bob holding an input m to verify (1) that the signature was issued by Alice and that (2) $C(m) = 1$.
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., $C(m) = 1$.
We foresee FAS to be a very useful primitive in practice, e.g. in the following authenticated policy verification 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 σ 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.
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.[5]
To see why such condition is not too restrictive, consider the above application. In that case, the security of FAS 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.
However, 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 σ hide as much information as possible about C except what can be inferred from knowledge of the distribution from which C is drawn.

[4] Such transformation was first reported in Boneh and Franklin [18].
[5] 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.

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) [19] 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.*[20]. More related primitives are content-concealing signatures and confidential signatures ([21, 22]) 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 2.5 and its construction from ABE with function privacy (FPABE, in short) is presented in Section 4.

We mention that it is possible to construct FAS in a more direct way from quasi-siO, but our aim is also to show equivalences among FAS, quasi-siO and FPFE (see Section 6).

1.3 Functional Encryption for Randomized Functionalities

Goyal *et al.* [23] 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 work (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 satisfy 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 FPMIFE for randomized functionalities (as said before, where the function privacy is restricted to deterministic circuits) with selective form of security. The restriction of selective security can be removed assuming in addition an adaptively-secure MIFE.

The definition of RFE and its security are presented in Section 2.4 and its construction from FPFE is presented in Section 5.

1.4 Efficient adaptively-secure FE for CNF/DNF formulae of bounded degree

Here we assume that the reader is familiar with inner-product encryption (IPE) introduced by Katz *et al.* [4].

Katz *et al.* show how to implement polynomial evaluation from IPE and how to build FE for a subclass of Boolean formulae with a bounded number (at most logarithmic in the security parameter) of variables (BoolEnc). Hereafter, for simplicity we focus on DNF formulae (of bounded degree) and thus we will call such FE scheme DNFEnc. Analogous considerations hold for other classes of Boolean formulae that can be derived from IPE, e.g., CNF formulae.

For instance conjunctions can be handled in the following way. Consider the predicate AND_{I_1,I_2} where $\mathsf{AND}_{I_1,I_2}(x_1,x_2) \stackrel{\triangle}{=} 1$ if both $x_1 = I_1$ and $x_2 = I_2$. Then, we can choose a random $r \leftarrow \mathbb{Z}_p$ (here we assume that the coefficient of the polynomial are over $\mathbb{Z}_p$) and letting the token correspond to the polynomial $p(x_1,x_2) \stackrel{\triangle}{=} r \cdot (x_1 - I_1) + (x_2 - I_2)$. If $\mathsf{AND}_{I_1,I_2}(x_1,x_2) = 1$ then $p(x_1,x_2) = 0$, whereas if $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ then, with all but negligible probability over the choices of r, it will hold that $p(x_1, x_2) \neq 0$. Disjunctions can be implemented by defining a polynomial $p'(x_1, x_2) \stackrel{\triangle}{=} (x_1 - I_1) \cdot (x_2 - I_2)$. It is straightforward that conjunctions and disjunctions can be combined to get DNF formulae but, as the Katz *et al.*'s transform from DNF formulae to polynomials grows super-polynomially in the number of variables, we have to put a bound on it.

As Katz *et al.* observe, in general the token may leak the value of r in which case the adversary will be able to find x_1, x_2 such that $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ yet $p(x_1, x_2) = 0$. Since, however, they consider the "selective" notion of security (where the adversary must commit to x_1, x_2 at the outset of the experiment), this is not a problem in their setting. On the other hand, disjunctions can be handled without issues.

Anyhow, this implies that even adaptively-secure IPE schemes [6] can not be directly employed in this transformation and thus to construct an adaptively-secure DNFEnc. FPFE turns out to be useful in this context: assuming that the underlying IPE satisfies our notion of function privacy, we show that a selectively-secure IPE with function privacy implies an adaptively-secure DNFEnc. The idea is that, being the token function-private, it hides the value r so that the adversary cannot make the reduction to fail. In Appendix B.1 we prove this fact.

We mention that this result does not require a FPFE for general circuits (and it would be an overkill as FPFE for general circuits follows from primitives stronger than iO that is already known to imply adaptively-secure FE for all poly-size Boolean circuits [24]) but it is sufficient to assume a more efficient function-private IPE scheme like the one of [12] based on weaker assumptions.

1.5 Relation between primitives

In Section 6, we present a general picture of the relations among all these related primitives. One key observation is that Attribute-based Encryption with function privacy implies FE, a notable fact that sheds light on the importance of the function privacy property for FE.

Concurrent work. After our work, Arriaga, Barbosa and Farshim [25] generalized FPFE to both different class of samplers and correlated messages in case of point functions. Their techniques and tools for non-correlated distributions share similarities with ours but their focus is not on potential applications as ours. Their generalizations of FPFE to correlated distributions could be useful in the context of Functional Anonymous Signature (see Section 1.2) but the constructions of Arriaga *et al.* only consider *point-functions*, a very restricted functionality. Moreover, for their generalizations Arriaga *et al.* need to assume forms of obfuscation stronger than quasi-siO (for the corresponding functionalities).

2 Definitions

2.1 Preliminaries

In our work, we make use of the following definition inspired by a similar definition from Agrawal *et al.* [13, 14].

Definition 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:

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

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.
In this work, we make use of puncturable pseudorandom functions [26] which are essentially pseudorandom functions (PRFs, in short) that can be defined on all inputs except for a polynomial number of inputs. Due to space constraints we refer the reader to [26] the formal definitions.

Functional Encryption Due to space constraints, we refer the reader to Boneh *et al.* [1] for the standard definitions of FE and its IND-Security.

2.2 Strong and Quasi-strong Indistinguishability Obfuscation

Strong indistinguishability obfuscation has been introduced by Bitansky *et al.* [15]. Their formulation is syntactically different from ours, but as they point out ([27], 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 [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 3 [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.* [15] put forward candidate constructions of siO for NC^1 circuits from variants of semantically secure graded encoding schemes [28]. They assert ([27], 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. We mention that this conjecture has been recently questioned in [16] in which the existence of siO is shown to clash with the existence of Canetti's 1997 AI-DHI assumption.[6]

[6] Notwithstanding it may be that it is the latter assumption to be false; in fact AI-DHI assumption is stated with auxiliary information and such classes of assumptions are subject to pathological counter-examples in which the auxiliary input can be set to an obfuscated circuit to obtain contention.

Anyhow, all our results can be instantiated assuming only quasi-siO for NC^1 circuits. For instance, quasi-siO for NC^1 circuits is sufficient to build FE with function privacy for NC^1 circuits and similarly for the other primitives we build. We stress that even constructions of (public-key) FPFE for NC^1 were not known before. 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 Function-Private Functional Encryption (FPFE)

A FPFE scheme is a FE scheme satisfying IND-Security and the following additional function privacy security notion.

Indistinguishability-based function privacy security. The IND-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 6).

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

1. $\mathcal{C}$ generates $(\mathsf{Mpk}, \mathsf{Msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and runs $\mathcal{A}_0$ on input 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 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 4 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 λ are such that $D_0 \triangleq \{D_{0,\lambda}\}_{\lambda\in\mathbb{N}}$ and $D_1 \triangleq \{D_{1,\lambda}\}_{\lambda\in\mathbb{N}}$ are of a pair of ensembles of *efficiently samplable* feasible entropy distributions[7] 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.

[7] 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 λ 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 λ, 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.

Definition 5 We say that FPFE is *indistinguishability function private secure* (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.[8]

Remark 6 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.

2.4 Functional Encryption for Randomized Functionalities

Goyal *et al.* [23] introduced the concept of FE for randomized functionalities. Like in Komargodski *et al.* [29] in this paper we do not take in account the problem of dishonest decryptors; this problem does not arise only in the context of randomized functionalities, and we think it go 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 refer to the aforementioned papers for details. In this paper we will focus on the functionality of randomized circuits, both randomized 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. Due to space constraints, we refer the reader to Goyal *et al.* for formal definitions of RFE. As our formalization of security for RFE we choose what Goyal *et al.* call "security against key queries after public-key" except that, as discussed before, we do not take in account dishonest decryptors.

2.5 Functional Anonymous Signature

Definition 7 [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* λ.
2. $\mathsf{FAS.Sign}(\mathsf{sk}, \mathsf{C})$, on input a signing key sk for security parameter λ, and a Boolean circuit $C \in C_\lambda$ outputs a *signature* σ of it.
3. $\mathsf{FAS.Verify}(\mathsf{vk}, \sigma, \mathsf{m})$, on input verification key vk for security parameter λ, a signature σ 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 λ, 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:

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

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

[8] 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 $\mathsf{negl}(n)$.

– (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 λ:

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 λ 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 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 λ.

3 Construction of FPFE from quasi-siO

Definition 8 [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 an enough expressive class of circuits $\mathcal{C}$ (at least all NC^1 circuits). We define a FPFE scheme $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}] = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Eval})$ for the class of circuits $\mathcal{C}$.

– $\mathsf{Setup}(1^\lambda)$: output the public-key Mpk and master secret-key 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 $\mathsf{FE.Eval}(\mathsf{Mpk}, \mathsf{Ct}, \mathtt{Tok})$.

It is easy to see that the scheme satisfies correctness assuming the correctness of $\mathsf{qsi}\mathcal{O}$ and FE, and the following theorem holds.

Theorem 9 *If* FE *is* IND*-Secure then* $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}]$ *is* IND*-Secure.*

Due to space constraints, we defer the security reduction of the following theorem to Appendix B.2.

Theorem 10 *If* $\mathsf{qsi}\mathcal{O}$ *is a quasi-siO then* $\mathsf{FPFE}[\mathsf{qsi}\mathcal{O}, \mathsf{FE}]$ *is* INDFP*-Secure.*

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 [17]. That is, if in the above construction we replace 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 of FAS: no adversary, given a token for circuit C can produce another token 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 is function private, the resulting FAS scheme is anonymous as well.

Definition 11 [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 vk and signing key sk to be respectively the public-key and the master secret-key output by the setup of 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 $\perp$.

It is easy to see that the scheme satisfies correctness assuming the correctness of FPABE and is functionally unforgeable.[9] In fact an adversary outputting a forgery that satisfies the winning condition of functional unforgeability, is a valid adversary against the security of FPABE and thus as in the Naor's transformation the forgery can be used to break the security of FPABE. Thus, the following theorem holds.

Theorem 12 *If* FPABE *is* IND*-Secure then* $\mathsf{FAS}[\mathsf{FPABE}]$ *is unforgeable.*

Due to space constraints we defer the proof of the following theorem to Appendix B.3.

Theorem 13 *If* FPABE *is* INDFP*-Secure then* $\mathsf{FAS}[\mathsf{FPABE}]$ *is anonymous.*

5 Construction of RFE from FPFE

Definition 14 [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 [10] $\mathcal{C}'$. We define a RFE scheme $\mathsf{RFE}[\mathsf{F}, \mathsf{FPFE}] =$

[9] 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.

[10] Precisely, in order to obtain theorem 16, 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 (see the following) and that can compute F. In particular, assuming that F can be computed in NC^1 we obtain an RFE scheme for NC^1 from a FPFE scheme for NC^1.

(Setup, KeyGen, Enc, Eval) for the class of randomized Boolean circuits $\mathcal{C} = \{\mathcal{C}_n\}_{n\in\mathbb{N}}$ induced by $\mathcal{C}'$[11] as follows.

- $\mathsf{Setup}(1^\lambda)$: generate the public-key Mpk and the master secret-key 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 Msk for security parameter λ, 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 $C(m||\mathsf{F.Eval}(\mathsf{k}, \mathsf{m}))$.

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

Remark 15 Note that a token decrypts two ciphertexts to the same value. Naively one could think that this makes the scheme subject to an attack: the adversary computes a ciphertext for message m_0 and decrypt it using a token for a circuit C to get $y = C(m_0; r)$ and then compare y with $z = C(m_b; r)$, the result of the decryption of the token on the challenge message encrypting m_b, and claim that $m_b = m_0$ if $y = z$. Anyhow, the equation $y = z$ will be satisfied with overwhelming probability for both $b = 0, 1$. In fact, by the constraint on the circuit C, the probability over the choices of r that $y = C(m_0; r)$ is only negligibly different from the probability that $y = C(m_1; r)$. At the end, not much changes with respect to traditional FE. Therein, an attacker can likewise obtain the result of the decryption of some token for circuit C on m_0, one of the two challenge messages (m_0, m_1), and test it against the decryption of the same token on the challenge ciphertext; but this does not help since $C(m_0) = C(m_1)$. In the case of FPFE the circuit is randomized but the latter equation is relaxed to hold w.v.h.p. probability so that the "attack" is still nullified.

Correctness. It is easy to see that the scheme satisfies correctness assuming the correctness of FPFE and the pseudorandomness of F.

Security reduction.

Theorem 16 If FPFE is IND-Secure and INDFP-Secure, and F is a puncturable pseudorandom function, then RFE[F, FPFE] is INDRFE-Secure.

Proof. 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 INDRFE-Security of RFE[F, FPFE] (here, for sake of simplicity we assume uniform adversaries). Recall that in the INDRFE-Security experiment the adversary $\mathcal{A}$ selects as challenges a pair of messages (m_0, m_1).

- H_0. This corresponds to the INDRFE-Security game in which the challenge ciphertext encrypts the message m_0.

[11] 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).

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

Claim 17 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 λ) 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 INDFP-Security of 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 18 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 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 INDFP-Security of 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 19 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 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 20 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 $\{0,1\}^{m(\lambda)}$, $C(m_0; s_0) = C(m_1; s_1)$ (see also Remark 15). Therefore, the indistinguishability of the two experiments follows from the IND-Security of 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 21 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 22 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 23 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 INDRFE-Security where the challenge encrypts respectively m_0 and m_1. Thus, the indistinguishability of the above hybrid experiments implies that the theorem holds.

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 [24]. 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.
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 [24], 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 it.
In Figure 1, given in Appendix A, we present relations among the primitives studied or discussed in this paper, except for the implication presented in Section 1.4 about IPE. 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.

References

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In *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*, pages 255–275, 2013.
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`.
14. 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.
15. 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.
16. Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Contention in cryptoland: Obfuscation, leakage and UCE. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 542–564, 2016.
17. 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.

18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.
24. Brent Waters. A punctured programming approach to adaptively secure functional encryption. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 678–697, 2015.
25. Afonso Arriaga, Manuel Barbosa, and Pooya Farshim. Private functional encryption: Indistinguishability-based definitions and constructions from obfuscation. Cryptology ePrint Archive, Report 2016/018, 2016. `http://eprint.iacr.org/`.
26. 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.
27. 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.
28. 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.
29. 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.

A Figure

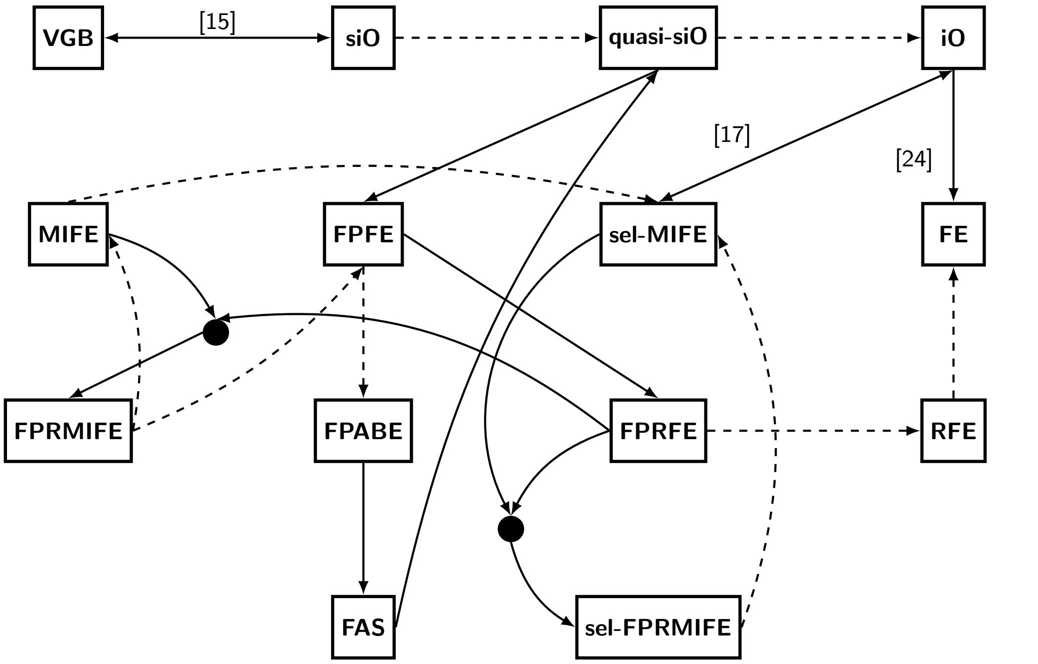

Fig. 1. Relations among primitives studied or discussed in this paper (except for the implication of Section 1.4): A line with arrow from A to B denotes that it is possible to build B from A and lines are annotated with the work where the implication first appeared, or unlabeled if such implication is 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 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.

B Proofs and security reductions

In this section we include the proofs that due to space constraints we did not include in the main body.

B.1 Selectively-secure function-private IPE → adaptively-secure DNFEnc

Now we prove that a selectively-secure function-private IPE implies an adaptively-secure DNFEnc (see Section 1.4).

Proof. For simplicity we will consider only a single conjunction and one token query. At the end we show how to remove such restrictions. Hereafter, we also assume that the reader be familiar with the Katz *et al.*'s transformation presented in Section 1.4.

Observe that the reduction from IPE to DNFEnc fails only in the case that the adversary for the IND-Security against the DNFEnc scheme is able to output as challenge a pair of values (x_1, x_2) such that $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ but $p(x_1, x_2) = 0$. Thus, we have to show that for any PPT adversary the probability that this event occur is negligible in the security parameter. Towards a contradiction we suppose that there exist an adversary $\mathcal{A}$ that during the IND-Security experiment against the DNFEnc scheme is able to output a challenge with such values x_1 and x_2 with some non-negligible probability $\alpha(n)$ (for simplicity hereafter we do not explicitly mention the payload messages).
Assume that for any value of the security parameter n our DNFEnc scheme is parameterized by a prime p_n of length n that induces the field $\mathbb{Z}_{p_n}$ over which the variables are defined. Given a conjunction $\phi \stackrel{\triangle}{=} \mathsf{AND}_{I_1,I_2}$ we call $p_{\phi,r}$ the corresponding polynomial that uses randomness r as specified in Section 1.4 (recall that here for simplicity we restrict the analysis to formulae consisting of a single conjunction) and we call $C_{\phi,r}$ the circuit that evaluates the predicate $C_{\phi,r}(x_1, x_2) = 1$ if and only if $p_{\phi,r}(x_1, x_2) = 0$.
Given a conjunction ϕ consider the pair of ensembles of distributions D_0 and D_1 such that for each value of the security parameter n, $D_{0,n}$ outputs $C^0_{\phi,r}$ for a randomly chosen element r in $\mathbb{Z}_{p_n}$ and $D_{1,n}$ outputs $C^1_{\phi,r}$ for an element r chosen such that its first n bits are set to 0 and the remaining are randomly selected.
It is easy to see that D_0 and D_1 are a pair of feasible entropy distributions (and in addition they are efficiently samplable). In fact, on any input (x_1, x_2) such that $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 1$ the two circuits output by the two distributions give the same answer (i.e., 1) and the adversary can find an input (x_1, x_2) such that $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ but $C^0(x_1, x_2) = 1$ with probability at most $2^{-n} \cdot q$, where $q(\cdot)$ is the number of oracle queries, and such that $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ but $C^1(x_1, x_2) = 1$ with probability at most $2^{-n/2} \cdot q$; thus any adversary with a polynomial number of queries can have only negligible advantage in distinguishing oracle access to the two distribution ensembles.

Now observe that the INDFP-Security guarantees that no PPT adversary can tell apart a token for the circuit C^0 from a token for the circuit C^1. So, we construct an adversary $\mathcal{B}$ against the INDFP-Security of DNFEnc that makes use of the adversary $\mathcal{A}$ against its IND-Security. $\mathcal{B}$ simulates the environment to $\mathcal{B}$ receiving from its challenger either a token for $C^0_{\phi,r}$ chosen by D_0 or a token for $C^1_{\phi,r}$ chosen by D_1. When $\mathcal{A}$ outputs its challenge (x_1, x_2), $\mathcal{B}$ checks 1) that the token evaluates on a ciphertext for this challenge to 1 (thus implying, by the correctness of the polynomial evaluation scheme, that $p_{\phi,r}(x_1, x_2) = 0$), 2) $\mathsf{AND}_{I_1,I_2}(x_1, x_2) = 0$ and if both checks are verified $\mathcal{B}$ computes $r' \stackrel{\triangle}{=} -(x_2 - I_2)/(x_1 - I_1)$ (notice that if condition 1) and 2) are satisfied then it cannot be $x_1 = I_1$) and finally output 0 (indicating a guess for the circuit C^0) if and only if the first $n/2$ bits of p are different from 0; if one of the previous checks is not satisfies $\mathcal{B}$ outputs 1 (indicating a guess for the circuit C^1).

If the circuit was chosen from D_0 then $\mathcal{B}$ outputs 0 with probability at least $\alpha(n)$, up to a negligible factor, whereas if the circuit was chosen from D_1 then $\mathcal{B}$ outputs 0 with probability 0. Therefore, $\mathcal{B}$ can win in the INDFP-Security game against DNFEnc, a contradiction.
For the general case of DNF formulae (instead of formulae consisting of a single conjunction) observe that a DNF formula, when implemented with the Katz *et al.*'s transformation consists in a product of polynomial $p_{\phi,r}$'s (see Section 1.4) in which each term uses different randomness (also recall that there is no randomness introduced for the disjunctions). Thus the general case for DNF formulae follows by a standard hybrid argument. The general case for multiple token queries can be handled by a standard hybrid argument as well.

B.2 Proof of Theorem 10

Proof. Suppose that there exists a legitimate function privacy adversaries $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ breaking the INDFP-Security of FPFE[qsi$\mathcal{O}$, 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 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.[12]).
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 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 FE with security parameter n and randomness r_1 to get the public-key Mpk and master secret-key Msk of FE.
Then, $\mathcal{D}_n$ runs $\mathcal{A}_n$ with randomness r_2 on input Mpk and answers the $\mathcal{A}_n$'s queries using 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 FE.KeyGen(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 qsiO.

B.3 Proof of Theorem 13

PROOF SKETCH. The proof is almost identical to that of theorem 10, 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

[12] 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

anonymity of $\mathsf{FAS}[\mathsf{FPABE}]$. 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 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 10 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 qsiO. $\square$