On the Semantic Security of Functional Encryption Schemes

Manuel Barbosa[1] and Pooya Farshim[2]

[1] HASLab – INESC TEC and Universidade do Minho, Portugal
[2] Department of Computer Science, Darmstadt University of Technology, Germany
mbb@di.uminho.pt, farshim@cased.de

Abstract. Functional encryption (FE) is a powerful cryptographic primitive that generalizes many asymmetric encryption systems proposed in recent years. Syntax and security definitions for general FE were recently proposed by Boneh, Sahai, and Waters (BSW) (TCC 2011) and independently by O'Neill (ePrint 2010/556). In this paper we revisit these definitions, identify several shortcomings in them, and propose a new definitional approach that overcomes these limitations. Our definitions display good compositionality properties and allow us to obtain new feasibility and impossibility results for adaptive token-extraction attack scenarios that shed further light on the potential reach of general FE for practical applications. The main contributions of the paper are the following:

- We show that the BSW definition of semantic security fails to reject intuitively insecure FE schemes where a ciphertext leaks more about an encrypted message than that which can be recovered from an image under the supported functionality. Our definition (as O'Neill's) does not suffer from this problem.
- We introduce an orthogonal notion of *setup security* that rejects all FE schemes where the master secret key may give unwanted power to the TA, allowing the recovery of extra information from images under the supported functionality. We prove FE schemes supporting *all-or-nothing* functionalities are intrinsically setup-secure and further show that many well-known functionalities *are* all-or-nothing.
- We extend the equivalence result of O'Neill between indistinguishability and semantic security to restricted *adaptive* token-extraction attacks (the standard notion of security for, e.g., IBE and ABE schemes). We establish that this equivalence holds for the large class of all-or-nothing functionalities. Conversely, we show that the proof technique used to establish this equivalence cannot be applied to schemes supporting a one-way function.
- We show that the notable *inner-product* functionality introduced by Katz, Sahai, and Waters (EUROCRYPT 2008) can be used to encode a one-way function under the Small Integer Solution (SIS) problem, and hence natural approaches to prove its (restricted) adaptive security fail. This complements the equivalence result of O'Neill for the non-adaptive case, and leaves open the question of proving the semantic security of existing inner-product encryption schemes.

Keywords. Functional encryption, Semantic security, Indistinguishability, Preimage samplability, Adaptive token extraction model, Inner-product encryption, Small integer solution.

1 Introduction

Functional encryption (FE) is a public-key primitive that generalizes many encryption systems, including public-key encryption, identity-based encryption, searchable encryption, attribute-based encryption, and all other variants of predicate encryption systems [BSW11]. In such a system, each decryption key TK_f (called a *token*) is associated with a function f (which may be viewed as a circuit). When a token holder runs the decryption algorithm on a ciphertext encrypting a message m, it recovers the image $f(\mathsf{m})$. A trusted authority holding a master secret key is responsible for issuing tokens. This allows the TA to control which users can recover which images from encrypted data. Realizing such a powerful primitive for complex functionalities could revolutionize information security applications in a way that is comparable to the notable case of fully homomorphic encryption. For example, one might encrypt a data set (e.g., a set of photographs) and share it in a cloud environment; the ability to issue tokens would then allow a user to award access to the results of different computations over the data set (e.g., partially blurred images). Interestingly, very recent developments in this area indicate that this may indeed be within our reach. For example, a concrete realization of functional encryption for arbitrary functionalities has been recently proposed in [GVW12], as well as functional encryption for regular languages in [Wat12].

The intuitive security requirement for a functional encryption scheme is that no information should leak from a ciphertext bar that which can be recovered via legitimately obtained decryption tokens. As for other encryption primitives, there are various ways in which this intuition can be formalized. In the case of public-key encryption, for example, the two standard formalizations are semantic security and ciphertext indistinguishability.

Semantic security is a simulation-based definition that measures to what extent real-world attacks on a cryptosystem can be replicated in an ideal world. Roughly speaking, in the real world an adversary is given a ciphertext c encrypting a message m sampled from a distribution $\mathcal{M}$; its goal is to recover some information v about the encrypted message. In the ideal world, a simulator must replicate the output of the adversary in the real world but now without having access to the ciphertext. (We include a variant by Goldreich [Gol04] in Appendix B.) Clearly, if one can show that such a simulator exists, then the ciphertext in the real world must be (computationally) hiding all information about the encrypted message and the encryption scheme can be considered secure.

Although semantic security is an intuitive and desirable security goal, it may be cumbersome to work with due to its simulation-based formalization. Luckily, by the seminal equivalence result of Goldwasser and Micali [GM84] indistinguishability-based notions (e.g., $\mathsf{IND\text{-}CPA}$) have proved to offer a convenient path to establishing the semantic security of public-key encryption schemes. The same approach was adopted for virtually all other primitives that can be classified as particular cases of functional encryption. Indeed, for all these primitives, the standard notions of security are indistinguishability-based ones and semantic security is not considered explicitly. In hindsight, the implicit assumption in all of this prior work was that, similarly to public-key encryption, an equivalence between indistinguishability and an intuitive security notion akin to semantic security would also hold for these primitives.

Somewhat surprisingly, in independent works, Boneh, Sahai, and Waters [BSW11] (BSW from here on) and O'Neill [O'N10] have shown that this is *not* the case for functional encryption schemes supporting complex functionalities. Indeed, both works demonstrated limitations in the indistinguishability-based notion for functional encryption and proposed strictly stronger semantic security notions to overcome these problems. These results highlight the importance of converging to a definition of semantic security for functional encryption that can be adopted as a de facto standard by the community. However, the definitional approaches adopted in both works are significantly different and the relation between the two is not well understood. In particular, it is not clear whether there are fundamental differences between the two so as to determine which one of them should be favored in detriment of the other.

The goal of this paper is to change this state of affairs. We analyze the positive and negative aspects of the definitions laid down by BSW and O'Neill and find that both approaches have strengths that should be preserved, and yet they also have weaknesses that should be reconsidered. We propose a new balanced set of definitions that incorporates the results of this analysis.

Analysis of previous definitions. Boneh, Sahai, and Waters [BSW11] provide an elegant generalization of the syntax of FE schemes, aiming to capture concrete primitives such as Identity-Based Encryption (IBE), Attribute-Based Encryption (ABE), and Predicate Encryption (PE) as particular cases. The authors propose a natural indistinguishability-based security definition for this new primitive, but then present a counterexample scheme showing that this notion of security is inadequate for general functionalities: the scheme is intuitively insecure, but can be proven $\mathsf{IND\text{-}CPA}$-secure. A notion of semantic security using black-box simulators is then proposed to address this problem, which is argued to exclude intuitively insecure schemes such as those presented in the aforementioned counterexample. The paper concludes with a series of feasibility results. On the positive side, the authors show that it is possible to construct semantically secure schemes supporting PE in the random-oracle model starting from any indistinguishable FE scheme supporting such functionalities. On the negative side, BSW show that semantically secure schemes do not exist even for simple functionalities such as IBE. This result hinges on the adversary's capability to perform adaptive token-extraction queries that corrupt, a posteriori, the security of the challenge ciphertext; this is a stronger attack model than that which is captured by the standard notions of security for primitives such as IBE, ABE, or PE.

Perhaps surprisingly, in this work we show that the BSW definition is too *weak* in the sense that it also fails to exclude some intuitively insecure schemes. The problem resides in the fact that the ideal-world simulator is allowed to control the generation of the global parameters for the functional encryption scheme. We show that this renders the simulator unreasonably more powerful than the real-world adversary, allowing the simulator to retain trapdoor information that permits recovering information from images $f(\mathsf{m})$, which otherwise would be hidden and cannot be recovered in the real world. Based on this observation, we present a counterexample scheme supporting a trapdoor one-way permutation (TDP) that is clearly intuitively insecure, yet can be proven BSW semantically secure.

Independently of Boneh et al., O'Neill [O'N10] proposed an alternative definitional approach to functional encryption schemes for general functionalities. The author presents alternative syntax, correctness, and indistinguishability-based security notions that are conceptually close to those in [BSW11], but proposes a significantly different semantic security definition. The paper then discusses the feasibility of achieving semantic security, by first presenting a separation to the indistinguishability notion, and then introducing a simple property for supported functionalities, called *preimage samplability*, under which the two notions are equivalent for non-adaptive token-extraction attacks. The fact that functionalities such as IBE and inner-product encryption [KSW08] are shown to be preimage samplable provide positive results for semantically secure FE schemes supporting such functionalities.

Analyzing the semantic security model proposed by O'Neill we find that it does not suffer from the same problem we identified for the BSW definition. Indeed, the ideal-world simulator in O'Neill's definition must work with honestly sampled global parameters, which means that it is unable to "cheat" by keeping extra trapdoor information. Nevertheless, we present other counterexamples for which the intuitive notion of security is not at all clear, but which can be proven secure under O'Neill's model. As an example, consider a functional encryption scheme supporting a TDP that reveals the corresponding trapdoor along with the decryption token. Should such a scheme be considered intuitively insecure? The distinction here is that information is leaked via tokens, rather than by the ciphertext, which means that it is hard to say such a scheme should be rejected by a semantic security definition. On the other hand, one can argue that a security definition for functional encryption should reject schemes that fail to preserve the security properties of the supported functionalities. Finally, as the author acknowledges [O'N10], O'Neill's notion of semantic security does not suitably deal with adaptive token-extraction attacks.

Main contributions. We now detail our main contributions.

Syntax. We start in Section 2 by tailoring the syntax and correctness definitions of functional encryption so as to capture the standard definitions for primitives such as IBE, ABE, PE, etc., as particular cases. This was not strictly the case with previous approaches. In particular, we identify a notion of *full correctness*, which maps to the notions adopted in [O'N10,BSW11], and imposes that the decryption operation explicitly returns a failure symbol when the functionality is undefined for a particular input value.

Indistinguishability. We modify the notion of *intentional leakage* [BSW11] to the slightly different concept of *potential leakage* in Section 3. This allows us to dissociate syntactic aspects (e.g., we do not need to include a special empty token in the syntax of the primitive) from the security aspects of an FE scheme. Potential leakage captures the general restrictions that must be in place to ensure that various security models exclude attacks on functional encryption schemes based on information that the scheme is not designed to conceal, e.g., the length of messages or the identity of the receivers. Through this notion we are able to define indistinguishability-based security as a natural generalization of the equivalent notions for standard primitives, and automatically get feasibility results that do not require transforming the original schemes (which strictly speaking was not the case in previous works).

Semantic security. Having identified a number of limitations of the semantic security models proposed by BSW and O'Neill (Section 4), in Section 5 we propose a notion of semantic security that incorporates features from the definitions by BSW, and also by O'Neill. Again, our goal is to faithfully generalize the definitions of semantic security for primitives like PKE [Gol04] and IBE [ACG^{+}06]. We observe that full adaptive token extraction models are not typically considered in such schemes, and so we propose a *restricted adaptive* token-extraction attack model. The restriction we impose intuitively prevents an attacker from obtaining decryption tokens that would allow it to trivially corrupt an encrypted ciphertext a posteriori, in the style of non-committing encryption [Nie02]. Put another way, our semantic security definition permits specifying the message distributions from which encrypted messages may be drawn, along with matching restrictions on the tokens that can be issued by the TA a posteriori, in order to provide FE security guarantees in a more flexible usage scenario. Using this strategy we circumvent impossibility results for unrestricted token extractions [BSW11]. Finally, we show that our semantic security definition displays a desirable composition property: security against single-message attacks implies security against multi-message attacks, even under restricted adaptive token-extraction attacks, thereby allowing us to present our results in the simpler single-message scenario.

Setup security. Our definition of semantic security preserves the resilience of O'Neill's definition in rejecting schemes that leak information to the adversary via the ciphertext. However, like all previous definitions, it does not provide any safeguards against leakage via decryption tokens or the master secret key. We therefore go on to introduce a new notion of *setup security* which enforces that tokens (or more strongly the setup procedure of the system) do not release any privileged information that might enable token holders or the trusted authority to obtain information which would otherwise be hidden by images values (Section 6). We show that setup security excludes all intuitively insecure schemes that we consider in the paper, while being inclusive enough to enable positive results for existing FE schemes. More precisely, we show that functionalities admitting a *conditional preimage sampling* procedure have an intrinsically secure setup procedure. We show PKE and IBE schemes, and more generally FE schemes supporting *all-or-nothing* functionalities, where an image either entirely reveals the encrypted message or nothing at all, are conditionally preimage samplable.

Adaptive equivalence. In Section 7 we present some positive feasibility results for our proposed notion of semantic security. There we extend O'Neill's results for non-adaptive token-extraction attacks and propose a variant of preimage samplability (PS) that enables us to obtain an equivalence between $\mathsf{IND\text{-}CPA}$-secure and semantically secure functional encryption under *restricted* adaptive token extraction. Moreover, our requirement is *weaker* than that of O'Neill if we are only interested in the non-adaptive token extraction scenario. Finally, we show that conditional preimage samplability (as defined to establish setup security) also implies our stronger notion of preimage samplability. We immediately get that indistinguishability-based security is equivalent to semantic security under restricted adaptive token-extraction attacks for all-or-nothing functionalities. This gives a wide range of positive results for (multi-message) semantically secure functional encryption that extends previous known results.

Inner products. We conclude the paper in Section 8 by presenting negative results for inner-product encryption (IPE). These results bring a twist to our extension of O'Neill's work: it is *not* the case that all the equivalences between semantic security and indistinguishability established by O'Neill for non-adaptive token extractions carry over to our restricted adaptive scenario. Concretely, we show that although inner-product encryption is proven by O'Neill to satisfy the preimage sampling property [O'N10], this functionality is provably *not* preimage samplable under the more restrictive PS notion that we introduce. Technically we show that, for certain parameterizations of the inner-product functionality, a successful preimage sampler can be used to break the Small Integer Solution (SIS) problem. This leaves open the question of proving the semantic security of existing inner-product encryption schemes under restricted adaptive token-extraction attacks.

2 Syntax and Correctness

We start by fixing the notion that we will be using throughout the paper. We write $x \leftarrow y$ for assigning value y to variable x. We write $x \leftarrow_{\$} X$ for sampling x from set X uniformly at random. If $\mathcal{A}$ is a probabilistic algorithm we write $y \leftarrow_{\$} \mathcal{A}(x_1, \ldots, x_n)$ for the action of running $\mathcal{A}$ on inputs $x_1, \ldots, x_n$ with random coins chosen uniformly at random, and assigning the result to y. We use ":" for appending to a list. For a random variable X, we denote by $[X]$ the support of X, i.e., the set of all values that X takes with nonzero probability. PPT as usual stands for probabilistic polynomial-time. All algorithms are PPT unless stated otherwise. We say $\nu(\lambda)$ is negligible if $|\nu(\lambda)| \in \lambda^{-\omega(1)}$. We use bold font, e.g., $\mathbf{m}$, to denote a vector, and slightly abuse notation when applying a function to each element of a vector, writing $f(\mathbf{m})$ when we mean $(f(\mathsf{m}_1), \ldots, f(\mathsf{m}_n))$, for $\mathbf{m} = (\mathsf{m}_1, \ldots, \mathsf{m}_n)$. We use $[X]_i$ for the ith component of X, and $[X]_i^j$ for the ith through jth components. We denote the ℓ_2 norm of a vector $\boldsymbol{x}$ by $\|\boldsymbol{x}\|_2$.

2.1 Functional encryption

Syntax. We now define the syntax for a functional encryption (FE) scheme, where the function space may, in general, *depend* on the public parameters of the system; see the discussion below. Such a scheme is specified by four PPT algorithms as follows.

1. $\mathsf{Setup}(1^\lambda)$: This is the setup algorithm. On input a security parameter 1^λ, it outputs a master secret key Msk and a master public key Mpk. Implicitly included in Mpk are a function/circuit space description FunSp and a message space MsgSp. The function space FunSp consists of descriptions of circuits $f : \mathsf{MsgSp} \to \mathsf{MsgSp} \cup \{\perp\}$.
2. $\mathsf{TKGen}(f, \mathsf{Msk})$: This is the token-generation algorithm. On input a function f and a master secret key Msk, it outputs a token TK for f.
3. $\mathsf{Enc}(\mathsf{m}, \mathsf{Mpk})$: This is the encryption algorithm. On input a message m and the master public key Mpk, it outputs a ciphertext c.
4. $\mathsf{Dec}(\mathsf{c}, \mathsf{TK})$: This is the deterministic decryption algorithm. On input a ciphertext c and a token TK, it outputs a message $\mathsf{m} \in \mathsf{MsgSp}$ or the special failure symbol $\perp$.

Correctness. The special symbol $\perp$ in the co-domain of functions accounts for functions that may be undefined on parts of their domain, or for which we do not expect the cryptosystem to behave correctly. Accordingly, we call an FE scheme *correct* if, for all $\lambda \in \mathbb{N}$, all $(\mathsf{Mpk}, \mathsf{Msk}) \in [\mathsf{Setup}(1^\lambda)]$, all $\mathsf{m} \in \mathsf{MsgSp}(\mathsf{Mpk})$, all $\mathsf{c} \in [\mathsf{Enc}(\mathsf{m}, \mathsf{Mpk})]$, all $f \in \mathsf{FunSp}(\mathsf{Mpk})$, and all $\mathsf{TK} \in [\mathsf{TKGen}(f, \mathsf{Msk})]$, we have that $f(\mathsf{m}) \neq \perp \implies \mathsf{Dec}(\mathsf{c}, \mathsf{TK}) = f(\mathsf{m})$. We call an FE scheme *fully correct* when the $f(\mathsf{m}) \neq \perp$ restriction is removed. In other words, the decryption algorithm must also return $\perp$ whenever $f(\mathsf{m}) = \perp$. The definition of correctness can be weakened in several ways. For instance one can relax the correctness condition so it holds only with an overwhelming probability over the choices of random coins for various algorithms. One may consider a game-based notion of correctness where a computationally bounded adversaries, with(out) access to the master secret key, aims to violate the system's functionality by outputting a malicious m and/or function f. All of our results carry over to these scenarios, and we do not consider them for the sake of simplicity.

Function spaces. We say an FE scheme FE supports f if for any set of random coins used in the setup procedure we have that $f \in \mathsf{FunSp}$. We say an FE scheme has a *fixed* function space if for each value of the security parameter, the set of functions that the scheme supports is fixed and independent of the coins of the setup procedure.

Comparison with the previous definitions. There are two differences to the definition in [O'N10]. First, O'Neill stipulates that the function space is indexed by the security parameter, yet it is fixed and independent of the setup algorithm. However, for a number of primitives such as inner-product encryption and attribute-based encryption, the function space may depend on the parameters generated by the setup algorithm. Second, we do not require correctness to hold when the function evaluates to $\perp$. As we will see, this weaker correctness notion allows us to see standard PKE, IBE, and other schemes as particular cases of functional encryption. Strictly speaking, this was not possible with the definition in [O'N10]. A correct FE scheme according to [O'N10] can be written as a fully correct scheme in our syntax, and vice versa.

One presentational difference between the definition in [BSW11] and that of ours is that we treat functions explicitly whereas BSW define a general functionality $F(K, \cdot)$ indexed by keys K. This difference is inconsequential as the description of a function can be interpreted as a key. In both definitions various spaces can depend on the public parameters.[3] Furthermore, we do not rely on a special empty key to model leakage of side information such as plaintext length. We will deal with this issue when defining the security of an FE scheme later on. Finally, our definition of correctness differs from BSW in the same way as it differs from that of O'Neill's.

2.2 Instantiations

We now show that a number of standard cryptographic primitives can be seen as special cases of the functional encryption syntax and correctness conditions as defined above. Other primitives such as attributed-based encryption, predicate encryption, and inner-product encryption can be captured in a similar way.

Public-key encryption. Given a correct PKE scheme, we can recast it as a correct FE scheme supporting the (fixed) function space consisting of the identity function $\mathsf{id}(\mathsf{m}) := \mathsf{m}$ only. Conversely, given a correct FE scheme supporting the identity function, we can recast it as a correct PKE scheme. The identity function is fully defined and hence correctness and full correctness coincide.

Identity-based encryption. Given a correct IBE scheme supporting identities $\mathsf{ID} \in \mathsf{IDSp}(\mathsf{Mpk})$, we may recast it as a *correct* FE scheme supporting the functions

$$f_{\mathsf{ID}^*}(\mathsf{m}, \mathsf{ID}) := \begin{cases} (\mathsf{m}, \mathsf{ID}) & \text{if } \mathsf{ID}^* = \mathsf{ID}\,; \\ \perp & \text{otherwise,} \end{cases}$$

for ID^* in the identify space $\mathsf{IDSp}(\mathsf{Mpk})$. The converse transformation also holds.

Weakly robust IBE. Note that the above correspondence assumes correctness on both sides. For full correctness, we obtain a primitive that closely resembles a *weakly robust* IBE scheme [ABN10]. In such an IBE scheme, it is infeasible to find a message and two distinct identities such that a random encryption of the message under one identity decrypts successfully to a non-$\perp$ value under the second identity with non-negligible probability. Indeed, a fully correct FE scheme as above gives a weakly robust IBE scheme. Conversely, a weakly robust IBE gives rise to an FE scheme for which full correctness for the above functionality holds with overwhelming probability.

Hidden-vector encryption. Boneh and Waters [BW07] proposed a hidden-vector encryption (HVE) system. In such a system, the message space comprises pairs $(\mathsf{m}, \boldsymbol{x})$, where $\boldsymbol{x}$ is a vector of n elements in $\{0, 1\}$ and the function space is indexed by a vector $\boldsymbol{y}$ of n elements in $\{\star\} \cup \{0, 1\}$, where we refer to $\star$ as the wildcard character. More precisely, the function space is given by

$$f_{\boldsymbol{y}}(\mathsf{m}, \boldsymbol{x}) := \begin{cases} \mathsf{m} & \text{if } \mathsf{Match}(\boldsymbol{x}, \boldsymbol{y}); \\ \perp & \text{otherwise,} \end{cases}$$

[3] This is somewhat implicit in [BSW11] and as we shall see it has implications towards the security of a scheme.

where $\mathsf{Match}(\boldsymbol{x},\boldsymbol{y})$ represents predicate $\boldsymbol{y}_i = \star \ \vee \ \boldsymbol{x}_i = \boldsymbol{y}_i$, for all $1 \le i \le n$. Similarly to above, with the caveat that the standard HVE definitions impose full correctness, any HVE scheme gives a fully correct FE scheme supporting the above function space under our syntactic definitional approach, and vice-versa.

3 Indistinguishability

To formalize various security definitions, we will be using a language similar to the code-based game playing [BR06]. We omit the **Initialize** and **Finalize** procedures, and define an explicit interaction between a game Game and an adversary $\mathcal{A}$. We use $\mathcal{A}^{\mathbf{O}}$ as a short hand for an adversary $\mathcal{A}$ that has access to all the oracles defined in a game. In each game we restrict attention to legitimate adversaries, a condition that is defined specifically for each game.

We define an indistinguishability-based notion of security that is tailored to capture different gradings of security for the same functionality. Indeed, the game is parameterized by a PPT relation R that defines the admissible set of challenge queries. This relation generalizes the restriction of choosing challenge messages with the same length in public-key encryption. By requiring $\mathsf{R}(\mathsf{m}_0,\mathsf{m}_1)$ to hold in the challenge query, one acknowledges that challenge queries that violate this restriction *may* lead to a (trivial) break. This approach decouples security concerns from the correctness of the scheme. We refer to the relation R as the *potential leakage relation.*

Definition 1 (IND-CPA Security). *Let game* $\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{A}}$ *be as defined in Figure 1. The* IND-CPA *security of an FE scheme relative to potential leakage relation* R*, requires the advantage of any adversary* $\mathcal{A} = (\mathcal{A}_1,\mathcal{A}_2)$ *to be negligible, when this is defined as*

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda) \Rightarrow \mathsf{T}\right] - 1.$$

```
Game $\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda)$:
  $b \leftarrow\$ \{0,1\}$; $\mathsf{TKList} \leftarrow [\,]$
  $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
  $b' \leftarrow\$ \mathcal{A}^{\mathbf{O}}(\mathsf{Mpk})$
  Return $(b' = b)$

oracle LR$(\mathsf{m}_0,\mathsf{m}_1)$:
  $\mathsf{c} \leftarrow\$ \mathsf{Enc}(\mathsf{m}_b,\mathsf{Mpk})$
  Return $\mathsf{c}$

oracle Token$(f)$:
  $\mathsf{TK} \leftarrow\$ \mathsf{TKGen}(f,\mathsf{Msk})$
  $\mathsf{TKList} \leftarrow f : \mathsf{TKList}$
  Return $\mathsf{TK}$
```

Fig. 1. Game defining the IND-CPA security of an FE scheme. An adversary is legitimate if: 1) it calls **LR** once and with a pair $(\mathsf{m}_0,\mathsf{m}_1)$ such that $\mathsf{R}(\mathsf{m}_0,\mathsf{m}_1)$ holds; 2) for all $f \in \mathsf{TKList}$ have $f(\mathsf{m}_0) = f(\mathsf{m}_1)$; and 3) in the token non-adaptive model, it does not call **Token** after calling **LR**.

Potential leakage. We now discuss how the potential leakage relation should be defined for the example instantiations we presented in the previous section, so that the resulting notion of security exactly matches the standard notions of security for those cryptographic primitives. A list is given in Table 1.

PKE	$\mathsf{R}(\mathsf{m}_0,\mathsf{m}_1) := \lvert\mathsf{m}_0\rvert = \lvert\mathsf{m}_1\rvert$
IBE	$\mathsf{R}((\mathsf{m}_0,\mathsf{ID}_0),(\mathsf{m}_1,\mathsf{ID}_1)) := \lvert\mathsf{m}_0\rvert = \lvert\mathsf{m}_1\rvert \wedge \mathsf{ID}_0 = \mathsf{ID}_1$
Anon. IBE	$\mathsf{R}((\mathsf{m}_0,\mathsf{ID}_0),(\mathsf{m}_1,\mathsf{ID}_1)) := \lvert\mathsf{m}_0\rvert = \lvert\mathsf{m}_1\rvert$
HVE	$\mathsf{R}((\mathsf{m}_0,\boldsymbol{x}_0),(\mathsf{m}_1,\boldsymbol{x}_1)) := \lvert\mathsf{m}_0\rvert = \lvert\mathsf{m}_1\rvert$
FE [O'N10]	$\mathsf{R}(\mathsf{m}_0,\mathsf{m}_1) := \lvert\mathsf{m}_0\rvert = \lvert\mathsf{m}_1\rvert$
FE [BSW11]	$\mathsf{R}(\mathsf{m}_0,\mathsf{m}_1) := \epsilon(\mathsf{m}_0) = \epsilon(\mathsf{m}_1)$

Table 1. Examples of the potential leakage relation.

Note the distinction between the different IBE flavors. In IBE, identities may be leaked and hence we must exclude challenge queries where two different identities are provided. This restriction is removed in anonymous IBE. The indistinguishability notion of security for weakly robust IBE (unlike the correctness requirement) is identical to that for IBE schemes.

Security. Not only can we relate the syntax of functional encryption schemes to that of existing primitives but, under the appropriate potential leakage relations in the table above, we can also reduce IND-CPA security of an FE scheme to an existing primitive and vice versa. Our choices therefore lead to a notion of functional encryption scheme that is indeed a generalization of existing cryptographic primitives.

Relation with O'Neill's definition. In [O'N10] the implicit potential leakage relation is fixed to be the equality of the message lengths, i.e., $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1) := (|\mathsf{m}_0| = |\mathsf{m}_1|)$. Although this is a natural choice, the resulting security definition fails to generalize those for IBE schemes (be it anonymous or non-anonymous). Indeed, non-anonymous IBE schemes cannot be captured under this definition as the functions take the identity as an input. For anonymous IBE schemes, the restriction artificially imposes that the sum total of lengths of the messages and identities queried to the left-or-right oracle to be equal. Our choice for the potential leakage relation deals with these issues in a cleaner way and is closer in sprit to the approach taken in [BSW11]. It is straightforward to see that a feasibility result under O'Neill's definitional choices leads to a feasibility result in our setting with a fixed function space, full correctness, and with respect to the length equality relation. The converse also holds.

Relation with BSW. Boneh, Sahai, and Waters [BSW11] define a special empty key (function) ϵ that is aimed at capturing information about encrypted messages that might be publicly recoverable from ciphertexts (typically including the message length). Formally, the authors assume that the token for this empty key is itself empty and, by the correctness requirement for the scheme, it follows that any party can run the decryption algorithm on a ciphertext to recover this so-called *intentionally leaked* information. However, this requirement implies that the standard syntax definitions for PKE, IBE, and other primitives do not naturally generalize to functional encryption and deviates from our goal. In BSW, for example, it is stated that an IBE should attach the message length and target identity to the ciphertext to strictly meet this requirement. This in turn forces us to reason about the implications of such transformations, e.g., for the scheme's security.[4] Our understanding is that the definitional choice corresponding to the empty key is mostly motivated to allow for an elegant definition of indistinguishability-based security. We, however, believe that our approach via relation R separates security issues from syntactic and correctness issues, while still maintaining the flexibility of the BSW definition. More formally, if the potential leakage relation is defined to be $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1) := (\epsilon(\mathsf{m}_0) = \epsilon(\mathsf{m}_1))$, queries that allow adversaries to exploit the empty token are excluded.

Given the discussion above, we can translate between feasibility result for the BSW definition of function encryption and our definition. Any scheme that is secure under the BSW definition yields a fully correct and secure scheme under our definition, when one introduces the appropriate syntactic changes and adopts a potential leakage relation that excludes challenge queries that could allow adversaries to explore the empty token. A conversion in the other direction implies transforming the scheme so that it explicitly leaks information through the empty token (e.g., attaching extra information to the ciphertexts, or adding a special token to the public parameters that leaks this information). The resulting scheme should not only be fully correct under our definition but also leak information that induces challenge query restrictions in the BSW model that match the original potential leakage relation. In this case, a security result in our model, yields a BSW-secure scheme.

[4] For example, for CCA security to be preserved, the decryption algorithm must also be changed with additional checks.

4 Limitations of the Models by BSW and O'Neill

It is well known that the standard notions of indistinguishability and semantic security for public-key encryption are equivalent [Gol04]. Indeed, the initial motivation for indistinguishability was to allow for a simpler strategy to establish that an encryption scheme achieves semantic security. However, a closer look at the indistinguishability notion of security for FE schemes reveals that a similar strategy is not valid for general functionalities. The problem is as follows [BSW11]. For some functionalities the restriction on the **LR** oracle imposing that $f(\mathsf{m}_0) = f(\mathsf{m}_1)$ can prevent the adversary from simultaneously extracting the token for f and launching a meaningful chosen-plaintext attack. To see this, consider any function f for which it is infeasible to find collisions. For example, if the function is injective, it is clear that any adversary that extracts the token for this function will be prevented from calling the challenge oracle on anything other than $\mathsf{m}_0 = \mathsf{m}_1$. However, in this case, the adversary will have no chance of winning the IND-CPA game.

4.1 The BSW model

Boneh et al. go on to turn this observation into a concrete functional encryption scheme supporting a one-way permutation that is intuitively insecure, but can be easily shown to satisfy the IND-CPA security definition. Roughly speaking, in this scheme one encrypts m under a standard PKE scheme. The token for the one-way permutation function f is the secret key for the PKE. Upon decryption, one first recovers m and then computes $f(\mathsf{m})$. The scheme is clearly correct. However, since $f(\mathsf{m})$ hides m computationally, the functional encryption scheme is not guaranteeing that the decryptor learns no more about the encrypted message than that which is leaked by $f(\mathsf{m})$. On the other hand, one can easily show that this FE scheme is IND-CPA-secure if the underlying PKE is itself IND-CPA-secure: if an adversary extracts the token for f, which is a permutation, then it is bound to calling the challenge oracle on $\mathsf{m}_0 = \mathsf{m}_1$; if it does not extract the token, then a simple reduction shows that it is attacking the PKE scheme.

Given the inadequacy of indistinguishability, Boneh et al. gave a semantic security model for FE schemes in the spirit of the semantic security definition given for PKE schemes [Gol04]. We recall this definition here.

Definition 2 (BSW Semantic Security). *The semantic security of an FE scheme requires that there exists a (black-box) simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ such that for any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and all distinguishers $\mathcal{D}$ the advantage*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) := \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] ,$$

is negligible, where games $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A},\mathcal{D}}$ and $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}$ are shown in Figure 2. Here, ϵ denotes the intentional leakage function corresponding to the empty token.

```
Game $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A},\mathcal{D}}(\lambda)$:
    $\mathsf{TKList} \leftarrow [\,]$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
    $(\mathbf{m}, \mathsf{st}) \leftarrow\$ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $\mathbf{c} \leftarrow\$ \mathsf{Enc}(\mathbf{m}, \mathsf{Mpk})$
    $v \leftarrow\$ \mathcal{A}_2^{\mathbf{Token}}(\mathsf{Mpk}, \mathbf{c}, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathbf{m}, \mathsf{st}, v, \mathsf{TKList})$
    Return $\mathcal{D}(\mathsf{trace})$

oracle Token($f$):
    $\mathsf{TK} \leftarrow\$ \mathsf{TKGen}(f, \mathsf{Msk})$
    $\mathsf{TKList} \leftarrow f : \mathsf{TKList}$
    Return $\mathsf{TK}$

Game $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)$:
    $\mathsf{List} \leftarrow [\,]$
    $(\mathsf{Mpk}, \tau) \leftarrow\$ \mathcal{S}_1(1^\lambda)$
    $(\mathbf{m}, \mathsf{st}) \leftarrow\$ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $v \leftarrow\$ \mathcal{S}_3^{\mathbf{Eval}, \mathcal{A}_2^{\circ}(\mathsf{Mpk},\cdot,\mathsf{st})}(\epsilon(\mathbf{m}), \tau)$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathbf{m}, \mathsf{st}, v, \mathsf{List})$
    Return $\mathcal{D}(\mathsf{trace})$

oracle Token($f$):
    $(\mathsf{TK}, \tau) \leftarrow\$ \mathcal{S}_2(f, \tau)$
    Return $\mathsf{TK}$

oracle Eval($f$):
    $\mathsf{List} \leftarrow f : \mathsf{List}$
    Return $f(\mathbf{m})$
```

Fig. 2. Games defining the BSW semantic security of an FE scheme [BSW11]. In the ideal game, notation $\mathcal{S}_3^{\mathcal{A}_2^{\circ}(\mathsf{Mpk},\cdot,\mathsf{st})}$ denotes that $\mathcal{S}_3$ has black-box access to $\mathcal{A}_2$, to which it must pass a ciphertext, and for which it must simulate a **Token** oracle.

We defer a discussion of the technical details of the definition to a subsequent section when we present an alternative. Nevertheless, we highlight at this moment the intuition behind the definition, as this is central to our main result in this section:

> **(BSW Intuition)** *The information leaked by a ciphertext and a set of decryption tokens is no more than that leaked by an equivalent set of images to a party who controls the generation of system parameters.*

Boneh et al. show that the above counterexample scheme for a one-way permutation f is not semantically secure, thereby providing evidence that semantic security is the correct notion of security for functional encryption. However, this raises the question if the semantic security model of BSW itself is adequate. We answer this question in the negative by using a similar argument to that given above for the inadequacy of indistinguishability: we demonstrate an intuitively insecure scheme that can be proven secure under the BSW model. Our result hinges on the fact that the ideal-world experiment allows the simulator to control the generation of system parameters and hence the description of the function space. This gives it an advantage over the real-world adversary, as it may be able keep trapdoor information that allows it to recover extra information from the images obtained from the **Eval** oracle.

THE COUNTEREXAMPLE. We restrict ourselves to the non-adaptive token extraction model so as not to fall within the range of impossibility results established in [BSW11]. (This only strengthens our argument.) In other words, we disallow $\mathcal{A}_2$ from querying the **Token** oracle. We note that our argument also goes through for the weaker definition of semantic security that is used in [BSW11, Definition 5] to present a (stronger) impossibility result when adaptive token-extraction queries are disallowed.

Consider a functional encryption scheme constructed from a public-key encryption scheme PKE and a one-way trapdoor permutation TDP as defined in Figure 3. The scheme provides the expected functionality by encrypting an input message under a standard PKE, and simply evaluating the TDP upon decryption. The token corresponding to the TDP is simply the PKE secret key. The trapdoor for the TDP is *not* kept as part of the secret parameters, and to make the point clearer one should think of it as being "destroyed" upon generation. Consider also that the intentional leakage for this scheme is defined as $|\mathsf{m}|$. The scheme is clearly correct.

```
alg. $\mathsf{Setup}(1^\lambda)$:
  $(f, f^{-1}) \leftarrow\!\$\ \mathsf{TDP.Gen}(1^\lambda)$
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\$\ \mathsf{PK.Gen}(1^\lambda)$
  $\mathsf{Msk} \leftarrow \mathsf{sk}$
  $\mathsf{FunSp} := \{f\}$
  $\mathsf{Mpk} \leftarrow (\mathsf{pk}, \mathsf{FunSp})$
  Return $(\mathsf{Msk}, \mathsf{Mpk})$
```

```
alg. $\mathsf{Enc}(\mathsf{m}, \mathsf{Mpk})$:
  $\mathsf{pk} \leftarrow \mathsf{Mpk}$
  $\mathsf{c} \leftarrow\!\$\ \mathsf{PK.Enc}(\mathsf{m}, \mathsf{pk})$
  Return $(\mathsf{c}, |\mathsf{m}|)$
```

```
alg. $\mathsf{TKGen}(f, \mathsf{Msk})$:
  $\mathsf{sk} \leftarrow \mathsf{Msk}$
  Return $\mathsf{sk}$
alg. $\mathsf{Dec}((\mathsf{c}, |\mathsf{m}|), \mathsf{TK}_f)$:
  $\mathsf{sk} \leftarrow \mathsf{TK}_f$
  $\mathsf{m} \leftarrow \mathsf{PK.Dec}(\mathsf{c}, \mathsf{sk})$
  Return $f(\mathsf{m})$
```

Fig. 3. An intuitively insecure but BSW-secure FE scheme from a PKE scheme and a family of TDPs.

Following the same reasoning as in the previous counterexample, this scheme leaks too much information to a decryptor holding a token for f: it will learn m, whereas only the image under the TDP should be leaked. Now consider the following BSW simulator $(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$:

- $\mathcal{S}_1$ executes $\mathsf{Setup}(1^\lambda)$, but now it keeps the trapdoor for the TDP in its state. It returns Mpk and $\tau = (\mathsf{Mpk}, \mathsf{Msk}, \mathsf{td})$.
- $\mathcal{S}_2$ answers the token-extraction queries, using Msk stored in its state, and updates the state to record the token-extraction queries placed by the adversary. Note that since only one functionality is supported, it is either the case that the adversary makes no token-extraction queries, or it extracts the token for f.

- $\mathcal{S}_3$ first checks if the adversary extracted the token for f. If not: 1) it generates a random message vector of appropriate size; 2) it constructs a ciphertext $\mathbf{c}$ encrypting the message vector under Mpk; and 3) it calls the adversary oracle on $\mathbf{c}$, and outputs whatever $\mathcal{A}_2$ returns. Otherwise: 1) it calls **Eval** to obtain $f(\mathbf{m})$; 2) it uses the trapdoor in its state to recover $\mathbf{m}$ (note that the result is guaranteed to coincide with the original encrypted message given that the TDP is injective); 3) it constructs a ciphertext $\mathbf{c}$ encrypting that message vector under Mpk; and 4) it calls the adversary oracle on $\mathbf{c}$ and outputs whatever $\mathcal{A}_2$ returns.

It is easy to show that simulator $\mathcal{S}$ always succeeds in simulating $\mathcal{A}$'s output, as long as the underlying PKE is IND-CPA-secure. If the adversary extracts the token for f, then the simulator is able reconstruct a perfect simulation of the ciphertext in the real game. On the other hand, if the adversary does not extract the token, then any adversary/distinguisher pair that detect the change in the ciphertext distribution can be used to break the IND-CPA security of the underlying PKE scheme.

Remark. The counterexample above is *not* based on the definition of the function space itself, but rather on the fact that an entity who controls parameter generation can keep trapdoor information on it. To make this point clearer, in Appendix A we present another version of this counterexample where we show that the problem remains even for functional encryption schemes where the function space is fixed and independent of the global parameters: we take a BSW semantically secure functional encryption scheme supporting all circuits in a family of trapdoor permutations, and then convert it into a scheme that is still semantically secure, but intuitively insecure. This means that, somewhat counter-intuitively, the BSW definition can also be inadequate for FE schemes with a fixed function space.

4.2 O'Neill's model and its potential shortcomings

We recall the semantic security model of O'Neill. As acknowledged by the author, the definition in [O'N10] fails to adequately capture adaptive token-extraction attacks, and hence we restrict attention to the non-adaptive scenario.

Definition 3 (O'Neill's Semantic Security). *Let games* $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A}}$ *and* $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S}}$ *be as shown in Figure 4. The semantic security of an FE scheme requires that for any adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$*, there exists a simulator* $\mathcal{S}$ *such that the advantage function below is negligible.*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathcal{A},\mathcal{S}}(\lambda) := \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A}}(\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S}}(\lambda) \Rightarrow \mathsf{T}\right]$$

```
Game $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathcal{A}}(\lambda)$:
    $\mathsf{TKList} \leftarrow [\,]$
    $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
    $\mathsf{st} \leftarrow\!\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $(\mathsf{m}, t) \leftarrow\!\!\$\ \mathcal{A}_2(\mathsf{Mpk},\mathsf{st})$
    $\mathsf{c} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{m},\mathsf{Mpk})$
    $t' \leftarrow\!\!\$\ \mathcal{A}_3(\mathsf{c},\mathsf{st})$
    Return $(t = t')$
```

```
oracle Token($f$):
    $\mathsf{TK} \leftarrow\!\!\$\ \mathsf{TKGen}(f,\mathsf{Msk})$
    $\mathsf{TKList} \leftarrow (f,\mathsf{TK}) : \mathsf{TKList}$
    Return $\mathsf{TK}$
```

```
Game $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathcal{A},\mathcal{S}}(\lambda)$:
    $\mathsf{TKList} \leftarrow [\,]$; $\mathsf{List} \leftarrow [\,]$
    $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
    $\mathsf{st} \leftarrow\!\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $(\mathsf{m}, t) \leftarrow\!\!\$\ \mathcal{A}_2(\mathsf{Mpk},\mathsf{st})$
    $\mathsf{List} \leftarrow [f(\mathsf{m})\ :\ (f,\cdot) \in \mathsf{TKList}]$
    $t' \leftarrow\!\!\$\ \mathcal{S}(\mathsf{Mpk},\mathsf{List},\mathsf{TKList},\mathsf{st})$
    Return $(t = t')$
```

```
oracle Token($f$):
    $\mathsf{TK} \leftarrow\!\!\$\ \mathsf{TKGen}(f,\mathsf{Msk})$
    $\mathsf{TKList} \leftarrow (f,\mathsf{TK}) : \mathsf{TKList}$
    Return $\mathsf{TK}$
```

Fig. 4. Games defining O'Neill's semantic security for an FE scheme in the non-adaptive token-extraction model [O'N10].

Once again we defer a discussion of the technical aspects of the definition to the next section, where we discuss the differences to our own definition. However, it is important at this point to highlight the intuition behind O'Neill's definition:

(O'Neill's Intuition) *The information leaked by a ciphertext and a set of decryption tokens is no more than that leaked by an equivalent set of images and tokens.*

There is a fundamental difference between this definition and that of BSW: the simulator is no longer in control of the generation of systems parameters. In return, a token-extraction oracle is provided in the ideal game. This means that the same strategy we presented above to argue for the inadequacy of the BSW definition does *not* apply directly to the definition by O'Neill. Nevertheless, other potential problems remain that we discuss next.

Potentially insecure scheme 1. We modify the BSW counterexample scheme as follows. The setup procedure is similar to before, except that the trapdoor for the randomly chosen permutation f is no longer destroyed but kept in the master secret key of the system. The token-generation algorithm is modified so that the token for the TDP contains the trapdoor (as well as the secret key for the PKE scheme). The encryption and decryption routines are as before. This scheme can be proven secure under O'Neill's definition: although the simulator can no longer generate the trapdoor information itself, this piece of information will become available once the adversary extracts the token for f. It is unclear if this scheme is intuitively insecure as the ciphertext does not leak any information beyond that leaked by images *and tokens*.

Potentially insecure scheme 2. Consider the following trivial construction of an FE scheme supporting its own encryption circuit. Take a PKE scheme and set the message space of the FE scheme to be (m, r) pairs. Take a PKE keys $(\mathsf{sk}, \mathsf{pk})$ and set the master secret key to be sk and the master public key to be pk. To functionally encrypt m re-encrypts under pk the ciphertext c resulting from $\mathsf{Enc}(\mathsf{m}, \mathsf{pk}; r)$. The decryption token is simply sk and decryption recovers and outputs c. This construction is clearly correct and it can be shown to be semantically secure under the previous semantic security definitions. It is also unclear whether it should be classified as insecure. On the one hand, it is hard to argue that it is intuitively insecure. This is because the function is evaluated on the sender's side *and* encrypted under a secure encryption scheme. Furthermore, the decryptor is the legitimate holder of the decryption key, and hence from its perspective there is no security property associated with the evaluated function. However, one can also consider things from the perspective of the encryptor, e.g., she might expect that token holders recover nothing but a ciphertext, but this is clearly not the case.

5 The New Semantic Security Model

In this section we propose new a definition of semantic security that avoids the above problems while simultaneously achieving several other goals of interest. Our definition is designed to be strong enough to exclude the clearly intuitively insecure counterexample we presented for the BSW definition and capture adaptive token extractions, without being infeasible to achieve due to its excessive strength. Furthermore, the definition should be compatible with the standard approaches to define the semantic security of PKE and IBE schemes. To this end, we directly start by giving the details of the definition, and justify our choices momentarily.

Definition 4 (Semantic Security). *Let games* $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}$ *and* $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}$ *be as shown in Figure 5. The semantic security of an FE scheme relative to potential leakage relation* R *requires that for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a legitimate PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for all PPT distinguishers* $\mathcal{D}$ *the following advantage function is negligible.*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) := \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right]$$

```
Game $\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}(\lambda)$:
  $\mathsf{FuncList} \leftarrow [\,]$
  $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
  $(\mathcal{M}, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
  $(\mathsf{m}, h, t) \leftarrow\!\!\$\ \mathcal{M}$
  $\mathsf{c} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{m}, \mathsf{Mpk})$
  $v \leftarrow\!\!\$\ \mathcal{A}_2^{\mathbf{Token}}(\mathsf{c}, h, \mathsf{st})$
  $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
  Return $\mathcal{D}(\mathsf{trace}, v)$
```

```
oracle Token($f$):
  $\mathsf{TK} \leftarrow\!\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
  $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
  Return $\mathsf{TK}$
```

```
Game $\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}(\lambda)$:
  $\mathsf{FuncList} \leftarrow [\,]$; $\mathsf{m} \leftarrow \perp$
  $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
  $(\mathcal{M}, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{S}_1^{\mathbf{Eval}}(\mathsf{Mpk})$
  $(\mathsf{m}, h, t) \leftarrow\!\!\$\ \mathcal{M}$
  $\mathsf{ImgList} \leftarrow [f(\mathsf{m}) : f \in \mathsf{FuncList}]$
  $v \leftarrow\!\!\$\ \mathcal{S}_2^{\mathbf{Eval}}(\mathsf{ImgList}, h, \mathsf{st})$
  $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
  Return $\mathcal{D}(\mathsf{trace}, v)$
```

```
oracle Eval($f$):
  $\mathsf{TK} \leftarrow\!\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
  $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
  Return $(\mathsf{TK}, f(\mathsf{m}))$
```

Fig. 5. Games defining the semantic security of an FE scheme under restricted adaptive token-extraction attacks. An adversary is legitimate if: 1) $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ holds for every pair of messages in $[\mathcal{M}]_1$; 2) for all second-stage **Token** queries f, we have that $f(\mathsf{m}_0) = f(\mathsf{m}_1)$ for all $\mathsf{m}_0, \mathsf{m}_1 \in [\mathcal{M}]_1$; and 3) in the token non-adaptive model, $\mathcal{A}_2$ and $\mathcal{S}_2$ do not call **Token** and **Eval** respectively.

The intuition behind the definition is as in the previous definitional approaches: an adversary should learn no more about an encrypted message than that which is explicitly revealed by the functions associated to the decryption tokens that it obtains. To this end, we require the existence of a simulator that does not have access to the ciphertext, but only to the images of the encrypted message under the same set of functions. This simulator is bound to producing an output that essentially looks like that produced by the adversary in the real world, which implies the ciphertext indeed reveals no extra information. More in detail, the simulator must emulate $\mathcal{A}_1$'s output and produce an output v that matches the information recovered by $\mathcal{A}_2$ from the ciphertext. However, the simulator is denied access to the ciphertext, and is bound to obtaining a set of images that matches those recovered by the real-world adversary via its **Token** oracle (this last restriction is imposed by including $\mathsf{FuncList}$ in trace). Like in the indistinguishability model, the potential leakage relation can be used to exclude trivial attacks whereby the real-world adversary would obtain information trivially leaked by the ciphertext (whereas this would not be available in the ideal world). Finally, we observe that the token-extraction queries performed by the adversary in the second stage are restricted to functions that are constant over the support of the message distribution. This allows us to generalize the feasibility results that are well known for particular instances of functional encryption, namely IBE schemes. For this reason, we call this model semantic security under *restricted adaptive token-extraction attacks.*

We now justify various definitional choices that we have made in the semantic security definition above.

Free simulators. In contrast to the previous definitional approaches, the simulators have a more general form in our definitions (modulo the generation of parameters; see below). More precisely, compared to O'Neill's definition, we have replaced adversary $\mathcal{A}_1$ with a simulator. This was done for the sake of clarity of the definition, and also brings the definition closer to the standard approach to defining semantic security for public-key encryption. The BSW model assumes a black-box simulator of special form.

Parameters and oracles. We do not allow the simulator to control the generation of global parameters. We provide the simulator with the master public key and give it access to decryption tokens, and hence our definitional choice here is closer to that of O'Neill. Our choice here is motivated by the principle that the simulator should not be more powerful than the adversary. This restriction, for example, forbids the simulator from tweaking the function space of the scheme, and hence the BSW counterexample that we discussed above can no longer be shown semantically secure (any simulator that succeeds with non-negligible probability in outputting $v = \mathsf{m}$ will contradict the security of the TDP). A token-extraction oracle should be provided to the simulator to compensate for the fact it no longer has access to the master secret key. Note also that in order to bind the simulator to work with the provided master public key, we must include it in the trace.

Distinguishers. In our definition, the outcomes of both the real-world and ideal-world experiments are obtained by passing the guessed information v and a full trace of the execution to a distinguisher $\mathcal{D}$. The trace includes the master public key, the list of extracted functions, the message distribution, and auxiliary information t about the challenge message that we call a *hint*. By requiring that the traces are indistinguishable and including $\mathsf{FuncList}$ in the trace, we meaningfully capture the restrictions on second-stage queries placed by the simulator, which was an acknowledged problem in [O'N10]. On the other hand, adversary $\mathcal{A}_3$ in O'Neill's definition is trying to guess an value t that is chosen by $\mathcal{A}_2$ along with the challenge message. Consequently, the $t = t'$ test at the end of the game can be seen as a canonical distinguisher, getting hint information t, consistently with our approach. For this reason, this choice strengthens our definition with respect to O'Neill's and is similar to the approach in [BSW11]. Indeed, the hint t plays a similar role to the state of the adversary in the BSW definition: it is provided to the distinguisher but hidden from the black-box simulator. Finally, this feature is fundamental to ensure that our definition composes from single-message to multi-message scenarios, even under *restricted adaptive token-extraction attacks* (see below for a formal statement of this result). This justifies that our presentation of the definition for the single-message scenario is without loss of generality.

Message distribution. In the BSW definition $\mathcal{A}_1$ directly outputs the message vector, whereas in our definition the adversary outputs a message distribution. There are two reasons for this. First, in our definition the simulator obtains leakage about the challenge message via the message distribution that ensures a balance with the leakage information available to the adversary via the challenge ciphertext (this balance is established via the restriction imposed by the potential leakage relation on the support of the message distribution output by a legitimate adversary). The same effect is achieved in BSW by explicitly providing the intentional leakage to the adversary, which implies the incorporation of the empty token into the syntax of functional encryption. Second, by introducing this change, we are able to establish equivalence to the indistinguishability notion for a larger class of schemes, even under restricted adaptive token-extraction attacks.

The message distribution in our definition does not output only a message, but also history information h that is given both to the adversary and the simulator, and an additional hint t that is provided only to the distinguisher. As we mentioned above, this is fundamental to obtain a composition theorem. Furthermore, it can be also related to the BSW model as follows. The BSW adversary can keep privileged information in its state, akin to a powerful history function $h(\mathsf{m}, \mathsf{Coins}(\mathcal{M}))$ that can take the coins used in sampling the message itself [BDPR98]. This corresponds to the history information h in our definition.

Our definition differs from O'Neill's in a similar way: we have replaced adversary $\mathcal{A}_2$ with a message distribution. We note that O'Neill's definition seems closer to our choice, as the description and the inputs of $\mathcal{A}_2$ can be seen as hardwired into $\mathcal{M}$. Conversely, we may view the state information as the description of a message distribution that $\mathcal{A}_2$ runs. Furthermore, as mentioned above, $\mathcal{A}_2$ outputs a value t that is fed to a canonical distinguisher that simply checks $t = t'$. However, a critical extension in comparison to O'Neill's model is that we provide history information h to both the adversary and the simulator.

Adaptive token queries. It is shown in [BSW11] that unrestricted token-extraction queries in the second stage lead to instantiability problems akin to those encountered in non-committing encryption [Nie02]. We therefore tailor our definition to exclude second-stage token-extraction queries under which such problems may arise. Intuitively, we rule out scenarios in which the trusted authority issues tokens that might allow an adversary to trivially corrupt, a posteriori, the security of past encryptions. We will see that for many FE scenarios this yields a natural definition of semantic security in the sense that it is equivalent to the indistinguishability-based notion for large classes of circuits, even under restricted adaptive token attacks. We emphasize that this attack model generalizes those adopted for standard primitives, including IBE, and ABE schemes [Gol04,ACG$^+$06]. As in the previous definitional approaches, we include the extracted functions in the trace so that the simulator is forced to make similar queries in the ideal game.

We summarize the features of our new definition: 1) free simulators, 2) honest parameter generation (as in O'Neill), 3) use of general distinguishers (closer to BSW), 4) message generation via a message distribution (closer to O'Neill), 5) history information (closer to BSW), and 6) hint for the distinguisher (present in both O'Neill and BSW).

Composition. Observe that the IND-CPA definition can be shown to compose from single to multiple **LR** queries (i.e., from a single-message to a multi-message attack scenario) using a standard hybrid argument [BDPR98]. One of the crucial features that our semantic security definition enjoys is that it also composes. Below we show that a multi-message variant of our definition where the message distribution outputs a *vector* of messages (see Appendix D for the details) is equivalent to the single-message version above.

Theorem 1 (Composition). *Let* FE *be a functional encryption scheme that is semantically secure under the (single-message) definition in Figure 5. Suppose that there is a polynomial* poly *such that for any single-message adversary $\mathcal{A}$, there is a semantic security simulator $\mathcal{S}[\mathcal{A}]$ such that*

$$\mathsf{Time}_{\mathcal{S}[\mathcal{A}]}(\lambda) \leq \mathsf{Time}_{\mathcal{A}}(\lambda) + \mathsf{Time}_{\mathcal{M}}(\lambda) + \mathsf{poly}(\lambda) \quad \textit{and} \quad \mathsf{Time}_{\mathcal{S}[\mathcal{M}]}(\lambda) \leq \mathsf{Time}_{\mathcal{M}}(\lambda) + \mathsf{poly}(\lambda)\,,$$

where $\mathcal{M}$ is the distribution output by $\mathcal{A}$ and $\mathcal{S}[\mathcal{M}]$ is the simulated message distribution. Then for every real-world multi-message PPT adversary $\mathcal{A}'$, there exist a real-world single-message PPT adversary $\mathcal{A}$ and a multi-message PPT simulator $\mathcal{S}'$ such that for any distinguisher $\mathcal{D}'$, there exists a distinguisher $\mathcal{D}$ for which

$$\mathbf{Adv}^{\mathsf{m\text{-}ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A}',\mathcal{S}',\mathcal{D}'}(\lambda) \leq \mathbf{Q}(\lambda) \cdot \mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda)\,,$$

where $\mathcal{S}$ is the simulator implied by the single-message semantic security and $\mathbf{Q}(\lambda)$ is an upper bound on the number of messages that the message distribution algorithms outputs.

Proof (Overview). We give an overview of the proof here and leave the details to Appendix D. The proof for the non-adaptive case is essentially a simulation-based hybrid argument. Consider the attack models where in the ith hybrid, for $i = 0, \ldots, q$, the adversary has access to $(q-i)$ ciphertexts and i image lists. In each step we change a ciphertext to the corresponding image list. We show that for any adversary in the ith hybrid model, there is an adversary in the $(i+1)$st hybrid that does equally well. To see this, note that the adversary in the ith hybrid can be viewed as a *single-message* real-world adversary that also receives some extra auxiliary information consisting of ciphertexts and image lists. By the semantic security guarantees of the scheme we may replace this adversary by an equally good one that only gets the image list for the replaced ciphertext. This concludes the proof as the qth hybrid corresponds to the ideal-world multi-message semantic security game. Note that the running time of the final simulator in the ideal game, which recursively depends on the previous simulators, stays polynomial if the condition given in the theorem is satisfied.

In the case of restricted adaptive token-extraction attacks, the argument is slightly more intricate, since part of the auxiliary information received by the adversary in the ith hybrid cannot be generated in the first stage of the adversary that we construct against single-message security (this is because some of the images will only be defined on second-stage token-extraction queries). However, under restricted adaptive token-extraction queries, it is possible to simulate the extra images on the fly directly from the description of the message distribution and the extracted token. □

Remark. The restriction on the running time of the single-message simulator can, for black-box simulators, be intuitively interpreted as an imposition that the simulator runs the adversary at most once. We will return to this later in the paper, when we construct black-box simulators for a large class of FE schemes.

COMPOSITION OF BSW AND O'NEILL. We do not know if the BSW and O'Neill's models compose. In the case of O'Neill's definition, it seems that the model lacks the necessary features to allow a proof technique similar to that used for Theorem 1. More in detail, O'Neill's definition does not allow modeling of message-dependent history information, and this is often necessary for composition of simulation-based security models [DDN00,Gol04].

6 Setup Security

In the previous section we defined a security model that (among other things) excludes the clearly intuitively insecure TDP-based counterexample from Section 4.1 that was problematic for the BSW semantic security model. In this respect, our security model has the same advantages as those of O'Neill. Unfortunately, similarly to O'Neill's model, our definition fails to exclude the counterexamples from Section 4.2 (because the simulator also has access to decryption tokens). This raises the question of whether our model can be strengthened further so these schemes are also ruled out. One direct approach to achieve this would be to further restrict the simulator by denying it access to tokens (as well as the master secret key) in the ideal world. We present this model in Appendix C and show that it is infeasible to achieve (essentially because a correct simulation of tokens would imply breaking the functional encryption scheme one is trying to prove semantically secure). We therefore take a different approach.

The first observation we make is that our definition accepts these counterexamples because indeed they do *not* leak information through the ciphertext. In fact, leakage is enabled by the combined information provided by *images* and decryption tokens (or more generally the master secret key). Intuitively, once a trapdoor for a TDP is provided to a token holder, the TDP circuit essentially becomes an efficiently invertible encoding function from messages onto images, offering no (intuitive) security whatsoever. From the point of view of the semantic security definition, where the aim is to exclude schemes where ciphertexts leak more information than that which is leaked by images, these counterexample schemes should therefore be considered secure.

However, it is still a reasonable security goal to expect that tokens do not help a token holder to extract information from images that would otherwise be hidden by the functionality. We therefore consider the stronger setting where the master secret key is required not to compromise the security properties of the supported functionalities. Informally, this means that even the trusted authority holding the master secret key should not be able to hold any "trapdoor" information on the functionalities supported by the functional encryption scheme.

A NEW NOTION OF SECURITY. Our approach to formalizing this security notion, which we call *setup security*, is as follows. We consider an attack scenario where an adversary is given the master secret key Msk and adaptively interacts with an evaluation oracle for the functionality in order to construct a trace that contains the master public key, a list of functions, a message distribution, a list of images, and a function func that models leaked information. This trace establishes the adversary's claim as to his ability to extract information from the images, which is specified by the leakage function. The FE scheme will then be setup-secure if a simulator given only the trace and the same set of images provided to the adversary, i.e., without having access to the master secret key, can extract essentially the same information.

Definition 5 (Setup Security). *The setup security of an FE scheme relative to potential leakage relation* R *requires that for any adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a simulator* $\mathcal{S}$ *such that the advantage function*

$$\mathbf{Adv}^{\mathsf{setsec}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S}}(\lambda) := 2 \cdot \Pr\left[\mathsf{SetSec}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S}}(\lambda) \Rightarrow \mathsf{T}\right] - 1$$

is negligible, where game $\mathsf{SetSec}_{\mathsf{FE},\mathsf{R},\mathcal{A}}$ *is shown in Figure 6.*

```
Game $\mathsf{SetSec}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S}}(\lambda)$:
    $\mathsf{FuncList} \leftarrow [\,]$; $b \leftarrow\$ \{0,1\}$
    $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M},\mathsf{st}) \leftarrow\$ \mathcal{A}_1(\mathsf{Msk},\mathsf{Mpk})$
    $\mathsf{m} \leftarrow\$ \mathcal{M}$
    $(\mathsf{func}, v_0) \leftarrow\$ \mathcal{A}_2^{\mathbf{Eval}}(\mathsf{st})$
    $\mathsf{ImgList} \leftarrow [f(\mathsf{m}) : f \in \mathsf{FuncList}]$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk},\mathcal{M},\mathsf{FuncList},\mathsf{ImgList},\mathsf{func})$
    $v_1 \leftarrow\$ \mathcal{S}(\mathsf{trace})$
    Return $(v_b = \mathsf{func}(\mathsf{m}))$

oracle $\mathbf{Eval}(f)$:
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $f(\mathsf{m})$
```

Fig. 6. Game defining the setup security of an FE scheme. An adversary is legitimate if $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ holds for every pair of messages in $[\mathcal{M}]$.

It is easy to see that all the potentially insecure TDP-based counterexamples that we have introduced are excluded under this definition. Indeed, let $\mathcal{A}_1$ be the adversary that outputs the uniform message distribution and places a single query to the **Eval** oracle containing the trapdoor permutation. Algorithm $\mathcal{A}_2$ is defined to invert the image using the trapdoor contained in the master secret key. The leakage function func is set to be the identity function. Then, any simulator for this adversary would lead to a successful inverter for the TDP. The counterexample involving double encryption is treated similarly.

Setup security via conditional preimage sampling. We show that the above definition (unlike "strong" semantic security; see Appendix C) allows natural classes of functionalities to be proven secure. To this end, we introduce a notion of *conditional preimage samplability.* Roughly speaking, this asserts that given a message distribution $\mathcal{M}$ and a list of functions $[f_i]_{i=1}^n$, it is possible to efficiently sample from $\mathcal{M}$ when this is conditioned on a set of images $[f_i(\mathsf{m})]_{i=1}^n$ for some $\mathsf{m} \in [\mathcal{M}]$. The actual definition is slightly more complex, as we need to deal with possible adversarial adaptiveness (hence the inclusion of the **Eval** oracle) as well as define indistinguishability of conditional distributions in a meaningful way (this is achieved via the third stage of the adversary $\mathcal{A}_3$).

Definition 6 (Conditional Preimage Samplability). *We say a functional encryption scheme supports a conditionally preimage samplable (CPS) functionality relative to potential leakage relation* R *if, for any PPT adversary* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$*, there exists a PPT sampler* Samp *such that the advantage*

$$\mathbf{Adv}^{\mathsf{cps}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp}}(\lambda) := 2 \cdot \Pr\left[\mathsf{CPS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp}}(\lambda) \Rightarrow \mathsf{T}\right] - 1$$

is negligible, where game $\mathsf{CPS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp}}$ *is shown in Figure 7.*

```
Game $\mathsf{CPS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp}}(\lambda)$:
    $\mathsf{FuncList} \leftarrow [\,]$; $b \leftarrow\$ \{0,1\}$
    $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M},\mathsf{st}) \leftarrow\$ \mathcal{A}_1(\mathsf{Msk},\mathsf{Mpk})$
    $\mathsf{m}_1 \leftarrow\$ \mathcal{M}$
    $\mathsf{st} \leftarrow\$ \mathcal{A}_2^{\mathbf{Eval}}(\mathsf{st})$
    $\mathsf{ImgList} \leftarrow [f(\mathsf{m}_1) : f \in \mathsf{FuncList}]$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk},\mathcal{M},\mathsf{FuncList},\mathsf{ImgList})$
    $\mathsf{m}_0 \leftarrow\$ \mathsf{Samp}(\mathsf{trace})$
    $b' \leftarrow\$ \mathcal{A}_3(\mathsf{m}_b,\mathsf{st})$
    Return $(b' = b)$

oracle $\mathbf{Eval}(f)$:
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $f(\mathsf{m}_1)$
```

Fig. 7. Game defining the conditional preimage samplability of an FE scheme. An adversary is legitimate if $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ holds for every pair of messages in $[\mathcal{M}]$. A sampler is legitimate if it outputs $\mathsf{m}_0 \in [\mathcal{M}]$.

The following theorem shows that conditional preimage samplability is a sufficient condition for setup security.

Theorem 2 (CPS $\Rightarrow$ Secure Setup). *Any functional encryption supporting a CPS functionality relative to potential leakage relation* R *is setup-secure with respect to* R.

Proof. The proof is straightforward. We build a setup security simulator $\mathcal{S}$ using the CPS sampler Samp as follows: run Samp on trace to get m_0 and then output $\mathsf{func}(\mathsf{m}_0)$. We show that this simulator is successful whenever Samp is. Suppose that there exists a setup security adversary $(\mathcal{A}_1, \mathcal{A}_2)$ that succeeds against $\mathcal{S}$. We first observe that any setup security adversary can be converted into the first two stages $(\mathcal{A}'_1, \mathcal{A}'_2)$ of a CPS adversary: just use $\mathcal{A}'_1 = \mathcal{A}_1$ without change and let $\mathcal{A}'_2$ run $\mathcal{A}_2$ and output $\mathsf{st} := (\mathsf{func}, v_0)$. We construct the third stage of a CPS attacker as follows: algorithm $\mathcal{A}'_3$ simply recovers (func, v) from the input state and outputs 1 if $\mathsf{func}(\mathsf{m}_b) = v$ and 0 otherwise. It is clear that the CPS adversary $(\mathcal{A}'_1, \mathcal{A}'_2, \mathcal{A}'_3)$ will have the same advantage as the setup security adversary $(\mathcal{A}_1, \mathcal{A}_2)$. However, we know that there can be no CPS adversary that succeeds against Samp, which means that $(\mathcal{A}_1, \mathcal{A}_2)$ cannot be successful. This concludes the proof. □

Remark. We note that weaker versions of both the definitions of setup security and conditional preimage samplability could be considered where the adversary does not get to choose the list of extracted functions adaptively. This could be achieved by denying $\mathcal{A}_2$ access to **Eval** and giving this to $\mathcal{A}_1$ instead ($\mathcal{A}_2$ would simply get a corresponding list of images). We did not consider this because we can achieve the stronger versions of the definitions presented above. Looking ahead, we note that such a non-adaptive version of collision samplability would still imply the notion of restricted preimage samplability that we will introduce in the next section.

Concrete setup-secure schemes. We now look at concrete functionalities and show that setup security is already achieved by many existing functional encryption schemes. We begin by defining a broad class of functionalities where an image either entirely reveals the encrypted message, or nothing at all.

Definition 7 (All-or-Nothing Functionality). *We say a functional encryption scheme supports an all-or-nothing (ANOT) functionality if for all* $\lambda \in \mathbb{N}$, *all* $(\mathsf{Mpk}, \mathsf{Msk}) \in [\mathsf{Setup}(1^\lambda)]$, *all* $f \in \mathsf{FunSp}$, *and all* $\mathsf{m} \in \mathsf{MsgSp}$ *we have that* $f(\mathsf{m}) \in \{\mathsf{m}, \bot\}$.

As an example, consider predicate encryption systems [BSW11] where the message space is partitioned into pairs $\mathsf{m} = (\mathsf{x}, \mathsf{idx})$. Here, x is a hidden payload and idx is extra information that determines which tokens can be used to recover x from a ciphertext encrypting m. More precisely, each secret key is associated with a predicate P, and the payload can be recovered whenever $P(\mathsf{idx}) = \mathsf{T}$. Formally,

$$f_P(\mathsf{x}, \mathsf{idx}) := \begin{cases} (\mathsf{x}, \mathsf{idx}) & \text{if } P(\mathsf{idx}) = \mathsf{T}\,; \\ \bot & \text{otherwise.} \end{cases}$$

Observe that we include idx in the output of the functionality when the predicate evaluates to T, thereby rendering the functionality all-or-nothing. It is easy to see that PKE and IBE schemes are examples of ANOT functional encryption schemes. For PKE schemes this is obvious, since the functionality is the identity function and the index space is empty. For IBE schemes, observe that whenever the output of the functionality is not $\bot$, the identity (i.e., the index) is also implicitly leaked by the functionality, thereby revealing the full message. Furthermore, this also includes variants of inner-product encryption, hidden vector encryption, etc., where a successful decryption operation explicitly reveals the encrypted index.

Our first positive result for setup security is given by the following theorem.

Theorem 3 (ANOT $\Rightarrow$ CPS). *Any ANOT functionality is conditionally preimage samplable in* expected *polynomial time (for any potential leakage relation).*

The intuition behind the proof of this theorem is as follows. Since the functionality is all-or-nothing, there are two possible trace outcomes. In the first, the adversary queries a function which acts as the identity map on the message sampled from $\mathcal{M}$. In this case the sampler can simply return the message. Second, it could be the case that all the image values are $\perp$. In this case the sampler will sample a message from $\mathcal{M}$ and checks if it maps to $\perp$ under all functions given to it in the trace. If this is the case, it outputs the message, else it retries. The number of retries, conditioned on a given trace, will depend on the probability that the image list is all $\perp$. However, the overall expected number of retries can be shown to be 1. We note that this is a form of rejection sampling, which guarantees that the output of the sampler is correctly distributed. The details of the proof may be found in Appendix G, where we also discuss why this sampler cannot be converted into a *strict* PPT black-box sampler by truncating the execution time.

Combining Theorems 2 and 3 we obtain the following corollary.

Corollary 1. *Any functional encryption scheme supporting an ANOT functionality has a secure setup procedure with respect to expected polynomial-time simulators and arbitrary potential leakage relations.*

Public-index predicate encryption. We now show that there exists a large class of FE schemes for which we can construct a strict PPT conditional preimage sampler. Intuitively, such schemes leak more information about encrypted messages which, when provided to the sampler, allows this stronger result to go through (in our framework, this is captured by the potential leakage relation, which ensures that there is a balance between the leakage trivially accessible to adversaries and that available to simulators and samplers).

Definition 8 (Jointly All-or-Nothing Functionality). *We say a functional encryption scheme supports a jointly all-or-nothing (JNOT) functionality relative to potential leakage relation* R *if, for all* $\lambda \in \mathbb{N}$, *all* $(\mathsf{Msk}, \mathsf{Mpk}) \in [\mathsf{Setup}(1^\lambda)]$, *all subsets* $\mathcal{F} \subseteq \mathsf{FunSp}$, *all message distributions* $\mathcal{M}$ *where* $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1) = \mathsf{T}$ *for all* $\mathsf{m}_0, \mathsf{m}_1 \in [\mathcal{M}]$, *we have that the following holds*

$$\forall \mathsf{m} \in [\mathcal{M}], \exists f \in \mathcal{F}, f(\mathsf{m}) = \mathsf{m} \quad \vee \quad \forall \mathsf{m} \in [\mathcal{M}], \forall f \in \mathcal{F}, f(\mathsf{m}) = \perp .$$

Roughly speaking, in this definition the all-or-nothing property is no longer formulated over a single message, but over a class of admissible message distributions defined by the potential leakage relation. Concretely, R constrains the support of the message distribution $\mathcal{M}$ in such a way that, for any subset of functions $\mathcal{F}$ extracted from the function space, the list of images will be guaranteed to, either totally reveal m, or to information theoretically preserve the entropy of the message distribution. In other words, conditioning $\mathcal{M}$ on the information that $f(\mathsf{m}) = \perp$ for all $f \in \mathcal{F}$ yields $\mathcal{M}$ itself.[5]

We now show that this definition is satisfied by a large class of all-or-nothing functionalities, corresponding to predicate encryption systems with *public index*, as introduced in [BSW11]. For such schemes, no claim about hiding idx is made. This means that their security is analyzed with respect to the special potential leakage relation

$$\mathsf{R}^*((\mathsf{x}_0, \mathsf{idx}_0), (\mathsf{x}_1, \mathsf{idx}_1)) := (|\mathsf{x}_0| = |\mathsf{x}_1| \ \wedge \ \mathsf{idx}_0 = \mathsf{idx}_1) .$$

The fact that message distributions are now restricted by R^* allow us to obtain the following result.

Theorem 4. *All predicate encryption systems are JNOT with respect to* R^* *as defined above.*

Proof. Message distributions are defined over pairs $(\mathsf{x}, \mathsf{idx})$. Further, for any message distribution $\mathcal{M}$, the restriction imposed by the potential leakage relation R guarantees that the same index idx^* is present in all

[5] Note that this is not the case for general all-or-nothing schemes, since an image list that does not reveal m may leak information that excludes part of the support of $\mathcal{M}$ as potential preimages.

pairs in its support. Now take any subset of functions $\mathcal{F}$ from the function space (i.e., a set of predicates in the appropriate class). Clearly, if for some element in the support of $\mathcal{M}$, all elements in $\mathcal{F}$ evaluate to $\perp$, this means that they are all rejecting idx^*. Consequently, this will also be the case for *all* other elements in $[\mathcal{M}]$. Conversely, if one of the predicates accepts idx^*, then the corresponding function will return $(\mathsf{x}, \mathsf{idx}^*)$ for all such elements in $[\mathcal{M}]$. □

The previous result includes primitives such as PKE, (non-anonymous) IBE, non-attribute-hiding ABE, and inner-product encryption that *reveals* the index in the ciphertext. We now prove the final result of this section.

Theorem 5 (JNOT ⇒ CPS). *Take an FE scheme supporting a JNOT functionality with respect to potential leakage relation* R*. Then this scheme is conditionally preimage samplable (in strict polynomial time) with respect to* R*.*

The intuition behind the proof of this theorem is exactly the same as in Theorem 3 (we also include it in Appendix G). The difference to all-or-nothing functionalities is that the JNOT property guarantees that, either the sampler gets the challenge message in the image list, or sampling a single message from the message distribution will yield a valid result. We obtain the following corollary.

Corollary 2. *All (public-index) predicate encryption systems are setup-secure with respect to potential leakage relation* R^**.*

7 Preimage Samplability

Despite the shortcomings of the indistinguishability model highlighted in [O'N10,BSW11], O'Neill shows that, for certain classes of functionalities, indistinguishability-based security is no less adequate than his proposed notion of semantic security. Indeed, it is shown in [O'N10] that if an FE scheme is *preimage samplable* (see Appendix E) then, in the non-adaptive token-extraction attack scenario, indistinguishability and semantic security are equivalent. Furthermore, functionalities such as those for identity-based encryption and inner-product encryption are shown to be preimage samplable.

In light of the new syntactical and definitional approach introduced above, we propose a modified definition of preimage samplability and show that a similar result holds. Our definition, however, permits extending the equivalence result to the *multi-message and restricted adaptive* token extraction model, and hence generalizes known results for, e.g., IBE schemes, in this area. This is an important extension, as (restricted) adaptive extraction of secret keys is the standard attack model for all predicate encryption systems.

Definition 9 ((Un)Restricted Preimage Samplability). *We call an FE (un)restricted preimage samplable for the potential leakage relation* R *if, for any algorithm* $\mathcal{A}$ *there exists a sampling algorithm* Samp *such that the advantage function*

$$\mathbf{Adv}^{\mathsf{mode\text{-}ps}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp}}(\lambda) := \Pr\left[\mathsf{PS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp},\mathsf{mode}}(\lambda) \Rightarrow \mathsf{F}\right]$$

is negligible, where $\mathsf{mode} \in \{\mathsf{res}, \mathsf{unres}\}$ *and game* $\mathsf{PS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp},\mathsf{mode}}$ *is defined in Figure 8.*

Comparison with O'Neill PS definition. Our definition differs from that in [O'N10] in several aspects. First, the adversary now has access to $(\mathsf{Msk}, \mathsf{Mpk})$ rather than 1^λ only. Access to Mpk is consistent with our syntax of FE schemes, which permits generation of function space together with the master public key. Access to Msk in needed when arguing that the actions of some $\mathsf{IND\text{-}CPA}$ adversary contradict preimage samplability. More precisely, the adversary may use information dependent on the Msk (i.e., decryption

```
Game $\mathsf{PS}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathsf{Samp},\mathsf{mode}}(\lambda)$:
$(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\$ \mathsf{Setup}(1^\lambda)$
$(\mathcal{M}, [f_j]_{j=1}^n) \leftarrow\$ \mathcal{A}(\mathsf{Msk}, \mathsf{Mpk})$
$\mathsf{m}_0 \leftarrow\$ \mathcal{M}$
$\mathsf{m}_1 \leftarrow\$ \mathsf{Samp}(\mathcal{M}, [(f_j, f_j(\mathsf{m}_0))]_{j=1}^n, \mathsf{Mpk})$
If $(\exists j\ :\ f_j(\mathsf{m}_0) \neq f_j(\mathsf{m}_1))$ Return $\mathsf{F}$
If $\neg\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ Return $\mathsf{F}$
If $(\mathsf{mode} = \mathsf{res} \wedge \mathsf{m}_1 \notin [\mathcal{M}])$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

Fig. 8. Game defining (un)restricted preimage samplability. $\mathcal{A}$ is legitimate if $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ holds for all $\mathsf{m}_0, \mathsf{m}_1 \in [\mathcal{M}]$.

tokens) to come up with a non-samplable message. This issue seems to have been overlooked in [O'N10]. Second, the definition is parameterized by a potential leakage relation R. This ensures that the sampler obtains as much information about the challenge message as a real-world semantic security adversary. Technically, this allows the equivalence proof to go through (for the non-adaptive case) for a larger class of functional encryption schemes than those covered by O'Neill. For example, our results cover those schemes that can be captured using our syntactic conventions, but not under those in [O'N10]. (A simple example of this is standard (non-anonymous) IBE.) Finally, the adversary now outputs a message distribution rather than a single message. The unrestricted sampler, similar to O'Neill's, is only bound to producing a message that collides with m_0 on all functions. The (stronger) restricted sampler is bound to return an m_1 that is in the support of $\mathcal{M}$. As we shall see, this is necessary to enable extending the equivalence result to the *restricted adaptive* token extraction setting. For the unrestricted case this condition is dropped, and we end up with a definition which is implied by (and hence *weaker* than) O'Neill PS definition (the $\mathcal{M}$ and $\mathcal{A}$ can be merged). Consequently, all positive feasibility results in [O'N10] carry over to our setting.

The following theorem, proven in Appendix F, establishes equivalence between our two notions of FE security for restricted preimage samplable schemes and restricted adaptive token extraction scenarios.

Theorem 6 (Equivalence under PS). *Fix potential leakage relation* R. *For every* IND-CPA *adversary* $\mathcal{A}$ *against scheme* FE, *there exist a (single-message)* SS-Real *adversary* $\mathcal{B}$ *and a distinguisher* $\mathcal{D}$ *such that for any simulator* $\mathcal{S}$

$$\mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda) \leq 2 \cdot \mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{B},\mathcal{S},\mathcal{D}}(\lambda)\,.$$

Furthermore, for every single-message SS-Real *adversary* $\mathcal{A}$, *there is a* PS *adversary* $\mathcal{C}$ *with sampler* Samp, *and a* SS-Ideal *simulator* $\mathcal{S}$ *such that for every distinguisher* $\mathcal{D}$ *there is an* IND-CPA *adversary* $\mathcal{B}$ *with*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) \leq \mathbf{Adv}^{\mathsf{ind\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{B}}(\lambda) + \mathbf{Adv}^{\mathsf{res\text{-}ps}}_{\mathsf{FE},\mathsf{R},\mathcal{C},\mathsf{Samp}}(\lambda)\,.$$

The running time of $\mathcal{S}$ *in the ideal world is that of running* $\mathcal{A}$ *in the real world plus the running time of* Samp.

We observe that the guarantee on the running time of the simulator allows us to obtain semantic security in the multi-message scenario from single-message indistinguishability via Theorem 1, provided that the running time of the sampler is independent of the running time of the adversary. We will see that this is indeed the case in our feasibility results below.

Remark. The above result can be extended to a setting where samplers, adversaries and simulators may execute in expected polynomial time. More precisely, for expected PPT preimage samplers, one can prove that IND-CPA security with respect to expected PPT adversaries is equivalent to semantic security when both the real-world adversary and the ideal-world simulator may run in expected polynomial time.

Feasibility. We conclude this section with a discussion of the feasibility results we obtain with the new definition of preimage samplability. On the negative side, it is easy to see that no FE scheme supporting a one-way function can be preimage samplable with respect to O'Neill's or our definition. Indeed, let f be a one-way function and consider an adversary $\mathcal{A}(\mathsf{Msk}, \mathsf{Mpk})$ that sets $\mathcal{M}$ to be the uniform distribution on the domain of f. Now any successful Samp for $\mathcal{A}$ can be used to break the one-wayness of f: run $\mathsf{Samp}(\mathcal{M}, [f, y], \mathsf{Mpk})$, where y is the image value to be inverted, and return the result.

On the positive side, and on top of all of O'Neill's feasibility results for the non-adaptive token extraction scenario, we show feasibility results for restricted preimage samplability for a large class of functionalities, which in turn immediately yield positive feasibility results for semantically secure functional encryption under restricted adaptive token extraction scenarios.

Theorem 7 (CPS $\Rightarrow$ PS). *Any conditionally preimage samplable functional encryption scheme is also restricted preimage samplable.*

Proof. Any PS adversary $\mathcal{A}$ can be easily converted in to a CPS adversary $(\mathcal{A}'_1, \mathcal{A}'_2)$. Let $\mathcal{A}'_1$ run $\mathcal{A}$ and output the message distribution $\mathcal{M}$ and $\mathsf{st} := ([f_j]_{j=1}^n)$. Algorithm $\mathcal{A}'_2$ simply queries all functions $[f_j]_{j=1}^n$ to **Eval**. Now observing that the PS sampler Samp takes exactly the same inputs as the CPS sampler Samp', which we know to exist, we will use $\mathsf{Samp} := \mathsf{Samp}'$. Since Samp' succeeds in the CPS game, we have that the indistinguishability of $(\mathsf{m}_b, \mathsf{trace})$ for $b \in \{0, 1\}$ implies that $[f_j]_{j=1}^n$ must coincide on $(\mathsf{m}_0, \mathsf{m}_1)$ with overwhelming probability. Furthermore $\mathsf{m}_1 \in [\mathcal{M}]$ by the legitimacy of the CPS sampler, which means that Samp is a successful *restricted* sampler. □

Combining this theorem with the equivalence result in Appendix F and Corollary 1 we get that any IND-CPA-secure FE scheme supporting an ANOT functionality is semantically secure under restricted adaptive token-extraction attacks and also enjoys setup security, both with respect to expected PPT simulators. (Note that if the PS sampler runs in expected PPT, then we must impose that the original scheme is IND-CPA-secure against *expected* PPT adversaries to obtain semantic security with respect to expected PPT simulators.) For the special cases where the potential leakage relation allows us to construct a strict PPT preimage sampler (e.g., PKE, IBE, and other predicate encryption schemes that explicitly leak the index) the result holds for strict PPT simulators. Furthermore, since the sampler executes $\mathcal{M}$ once, this allows us to extend the implication from single-message IND-CPA security to multi-message semantic security under restricted adaptive token-extraction attacks.

Remark. Recall that setup security formalized the intuition that tokens and the master secret key do not leak any "trapdoor" information about the functionality. Another way to look at this is to see it in terms of allowing a simulator to "cheat." For instance, the ability to control the function space generation leads to a cheating simulator in the BSW counterexample. Our approach in defining setup security was to rule out cheating information for free (i.e., non-black-box) simulators. However, we can also look at cheating for black-box simulation. In particular we can look at the simulator constructed using a PS sampler. One can argue that this simulator is "non-cheating," as it uses its state info in exactly the same way as the real-world adversary uses it, and assuming that the real-world adversary cannot cheat by definition. However, this still leaves open the possibility of leakage through tokens which, as we have discussed, is a *grey area* in terms of intuitive security/insecurity. We leave demonstrating a separation here as an interesting open problem, as this would imply a functionality that is PS but not CPS.

8 Inner-Product Encryption

Inner-product encryption (IPE) [KSW08] is a form of functional encryption where the index space corresponds to vectors $\boldsymbol{x}$ in $\mathbb{Z}_q^m$, and each secret key is also associated with a vector $\boldsymbol{y}$ in $\mathbb{Z}_q^m$. The associated predicate is given by

$$P_{\boldsymbol{y}}(\boldsymbol{x}) := \begin{cases} \mathsf{T} & \text{if } \langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0 \mod q\,; \\ \mathsf{F} & \text{otherwise.} \end{cases}$$

Without loss of generality, we will concentrate on the predicate-only version of inner-product encryption, where the payload is empty and the functionality is $f_{\boldsymbol{y}}(\mathsf{m}) = P_{\boldsymbol{y}}(\boldsymbol{x})$. Note that IPE is *not* an all-or-nothing functionality, since upon successful decryption one does not learn $\boldsymbol{x}$.

Our goal is to show that inner-product encryption is not restricted preimage samplable. To this end, we will rely on well-established intractable problems related to finding short solutions to linear equations. More precisely, we will be relying on the Small Integer Solution (SIS) problem and a decisional variant of it that we call DSIS [MR07,GPV08,Lyu12] (see Appendix H for the details).

We will show that, for certain parameters q, m in the inner-product functionality, no restricted preimage sampler can be successful against the PS adversary $\mathcal{A}$ shown in Figure 9. This adversary is parameterized by four values n, m, q, and d, all of which we assume to be polynomial in the security parameter. This guarantees that the algorithm executes in polynomial time.

```
Algorithm $\mathcal{A}_{q,n,m,d}(\mathsf{Msk}, \mathsf{Mpk})$:
For $i$ from 1 to $n$ do
    $\boldsymbol{y}_i \leftarrow\$ \mathbb{Z}_q^m$
Set $\mathcal{M}$ to uniform on $\mathrm{B}(d)^m$
Return $(\mathcal{M}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$
```

Fig. 9. PS adversary for the inner-product encryption.

The formal statement of our result is as follows.

Theorem 8 (IPE Is Not Restricted PS). *Take the PS adversary $\mathcal{A}$ as defined in Figure 9. Then for any PPT sampler* Samp*, there exist PPT adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$1 - \mathbf{Adv}^{\mathsf{res\text{-}ps}}_{\mathsf{FE},\mathsf{R},\mathcal{A}_{q,n,m,d},\mathsf{Samp}}(\lambda) \leq \mathbf{Adv}^{(q,m,n,d)\text{-}\mathsf{dsis}}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{(q',m',n',\beta)\text{-}\mathsf{sis}}_{\mathcal{B}_2}(\lambda) + \nu(\lambda)\,,$$

where $d = q^{n/m}$, $q' = q$, $m' = m$, $n' = n/q - \sqrt{n/q}\log(n/q)$, $\beta = d\sqrt{m}$, and $\nu(\lambda)$ is a negligible function depending on q and n.

We note that this is a stronger result than what we need as it establishes that the sampler will fail with *overwhelming* probability. We leave the details of the proof to Appendix H, where we also briefly discuss how to extend the theorem to large values of q, and give a high-level overview here.

The main idea behind the proof is that a successful sampler should match the zero values in the image list it receives, while being restricted to outputting solutions in the support of the message distribution, which consists of *small* vectors. In other words, the sampler is solving a system of linear equations with a small solution. This allows us to establish a connection with the SIS problem. Despite this, in order to solve a SIS problem instance using the sampler, we need to make sure that the sampler is forced to match sufficiently many zeros. (Note it cannot be the case that sampling a random message leads to only zero image values as otherwise we can preimage sample by repeated sampling as before.) Hence enough zero and nonzero values must be present in the image list. We achieve this by making sure the adversary $\mathcal{A}$ returns more vectors $\boldsymbol{y}_i$ than the SIS dimension n'. But now there is a problem as we do not know the image values for the newly generated vectors. This is where we appeal to the DSIS problem and simply assume these values are random: any change in the sampler's success probability would translate to a DSIS break. We are now in a position where we can reduce to the SIS problem. We assign the rows of the SIS matrix to (some of) the zeros in the randomly generated values, and assign the newly generated rows to the remaining ones. A successful sampler for this set of images would also solve the SIS problem instance.

Acknowledgments. The authors would like to thank Daniel Cabarcas, Angelo De Caro, Gottfried Herold, Vincenzo Iovino, Vadim Lyubashevsky, and Giuseppe Persiano for helpful discussions.

References

ABN10. Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 480–497. Springer, 2010. (Cited on page 6.)

ACG$^+$06. Nuttapong Attrapadung, Yang Cui, David Galindo, Goichiro Hanaoka, Ichiro Hasuo, Hideki Imai, Kanta Matsuura, Peng Yang, and Rui Zhang. Relations among notions of security for identity based encryption schemes. In José R. Correa, Alejandro Hevia, and Marcos A. Kiwi, editors, *LATIN*, volume 3887 of *Lecture Notes in Computer Science*, pages 130–141. Springer, 2006. (Cited on pages 4 and 14.)

BDPR98. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In Hugo Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998. (Cited on pages 14 and 15.)

BR06. Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 409–426. Springer, 2006. (Cited on pages 7, 30, and 33.)

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, 2011. (Cited on pages 1, 2, 3, 4, 6, 7, 8, 9, 10, 14, 18, 19, and 20.)

BW07. Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, 2007. (Cited on page 6.)

DDN00. Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000. (Cited on page 16.)

GM84. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. (Cited on page 2.)

Gol04. Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications.* Cambridge University Press, 2004. (Cited on pages 2, 4, 9, 14, 16, and 25.)

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *STOC*, pages 197–206. ACM, 2008. (Cited on page 23.)

GVW12. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Safavi-Naini and Canetti [SNC12], pages 162–179. (Cited on page 1.)

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, 2008. (Cited on pages 3 and 22.)

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012. (Cited on page 23.)

MR07. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. (Cited on pages 23 and 34.)

Nie02. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In Moti Yung, editor, *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126. Springer, 2002. (Cited on pages 4 and 14.)

O'N10. Adam O'Neill. Definitional issues in functional encryption. *IACR Cryptology ePrint Archive*, 2010:556, 2010. (Cited on pages 2, 3, 4, 5, 6, 7, 8, 11, 14, 20, 21, and 28.)

SNC12. Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012. (Cited on page 24.)

Wat12. Brent Waters. Functional encryption for regular languages. In Safavi-Naini and Canetti [SNC12], pages 218–235. (Cited on page 1.)

A Counterexample with a Fixed Function Space

Consider the following alternative version of the TDP-based counterexample presented in Section 4. Take a functional encryption scheme that is token non-adaptive BSW semantically secure, and suppose it supports the a family of trapdoor one-way permutations (for any choice of f), as well as the identity function. Modify this scheme by sampling a special circuit $f^\star$ at setup, and include it in the Mpk. Again, the corresponding trapdoor is "forgotten" by the setup algorithm. We use the token for the identity function for $f^\star$ and the original tokens for the other functions in the function space. The decryption procedure is modified accordingly similarly to the counterexample given in Figure 3.

This scheme should be intuitively classified as insecure, as an adversary that extracts the token for $f^\star$ learns more than it should about the message. However, the BSW semantic security of the original scheme allows proving that the modified scheme is also BSW semantically secure. Given any adversary $\mathcal{A}$ against the modified scheme, construct an adversary $\mathcal{A}'$ against the original scheme by letting $\mathcal{A}'_1$ run $\mathcal{A}_1$ on Mpk plus the description of an $f^\star$ which $\mathcal{A}'_1$ samples accordingly to the definition of the TDP. Token-extraction queries on this $f^\star$ are answered by extracting the identity function instead; for other functions, the token-extraction oracle is used directly. $\mathcal{A}'_2$ simply runs $\mathcal{A}_2$. The security of the original scheme implies a black-box simulator $\mathcal{S}'$. Observe that if $\mathcal{A}_1$ extracts the token for $f^\star$, then $\mathcal{A}'_1$ will be extracting the token for the identity function, which means $\mathcal{S}'_3$ has access to m.

We now use $\mathcal{S}'$ to construct a simulator $\mathcal{S}$ for the modified scheme. $\mathcal{S}_1$ uses $\mathcal{S}'_1$ to obtain (Mpk, τ), it generates a new $f^\star$, and keeps the trapdoor for it and τ in its state. $\mathcal{S}_2$ uses $\mathcal{S}'_2$ to answer the token-extraction queries, keeping a list of all such queries, and obtaining the token for the identity function if the adversary extracts $f^\star$. Finally, $\mathcal{S}_3$ runs $\mathcal{S}'_3$ forwarding inputs and outputs. The only caveat occurs when $\mathcal{S}'_3$ queries the **Eval** oracle on the identity function. Whenever this is the case, $\mathcal{S}_3$ checks its records to see if this query corresponds to an extraction query on $f^\star$ and, if so, obtains an image under $f^\star$ instead. Since $\mathcal{S}_3$ knows a trapdoor, it can invert the image and provide the unique correct value to $\mathcal{S}'_3$ to ensure a correct simulation. On the other hand, by ensuring that the correct $f^\star$ query is placed on **Eval**, algorithm $\mathcal{S}_3$ is able to satisfy the requirement of making the same queries as those of the real adversary.

B PKE Semantic Security

We recall Goldreich's definition of semantic security for public-key encryption scheme in Figure 10. This definition is taken from [Gol04, Definition 5.4.8].

Definition 10 (PKE Semantic Security). *A public-key encryption scheme,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, is said to be semantically secure under chosen-plaintext attacks if for every pair of PPT oracle machines,* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a pair of PPT algorithms,* $\mathcal{S} := (\mathcal{S}_1, \mathcal{S}_2)$*, such that*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{PKE},\mathcal{A},\mathcal{S}}(\lambda) := \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{PKE},\mathcal{A}}(\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{PKE},\mathcal{S}}(\lambda) \Rightarrow \mathsf{T}\right]$$

is negligible, where games $\mathsf{SS\text{-}Real}_{\mathsf{PKE},\mathcal{A}}$ *and* $\mathsf{SS\text{-}Ideal}_{\mathsf{PKE},\mathcal{S}}$ *are shown in Figure 10. Furthermore, we require the traces in the two games to be identically distributed.*

Game $\mathsf{SS\text{-}Real}_{\mathsf{PKE},\mathcal{A}}(\lambda)$:	**Game** $\mathsf{SS\text{-}Ideal}_{\mathsf{PKE},\mathcal{S}}(\lambda)$:
$(\mathsf{sk}, \mathsf{pk}) \leftarrow\$ \mathsf{PKGen}(1^\lambda)$	
$((\mathcal{M}, h, \mathsf{func}), \mathsf{st}) \leftarrow\$ \mathcal{A}_1(\mathsf{pk})$	$((\mathcal{M}, h, \mathsf{func}), \mathsf{st}) \leftarrow\$ \mathcal{S}_1(1^\lambda)$
$\mathsf{m} \leftarrow\$ \mathcal{M}$	$\mathsf{m} \leftarrow\$ \mathcal{M}$
$\mathsf{c} \leftarrow\$ \mathsf{PKEnc}(\mathsf{m}, \mathsf{Mpk})$	
$v \leftarrow\$ \mathcal{A}_2(\mathsf{c}, h(\mathsf{m}), \mathsf{st})$	$v \leftarrow\$ \mathcal{S}_2(1^{\lvert\mathsf{m}\rvert}, h(\mathsf{m}), \mathsf{st})$
$\mathsf{trace} \leftarrow (\mathcal{M}, h, \mathsf{func})$	$\mathsf{trace} \leftarrow (\mathcal{M}, h, \mathsf{func})$
Return $(v = \mathsf{func}(\mathsf{m}))$	Return $(v = \mathsf{func}(\mathsf{m}))$

Fig. 10. Games defining the semantic security of a PKE scheme.

C A Naïve Approach to Excluding Counterexamples

The most immediate approach to address all the counterexamples presented in the paper is to constrain the simulator so that the ideal-world experiment directly captures the intuition that the simulator can only get to see the images. This definition is presented in Figure 11 and it formalizes the following intuition:

> **(Strong Intuition)** *The information leaked by a ciphertext and a set of decryption tokens is no more than that leaked by an equivalent set of images.*

Note that the simulator is highly restricted in that it can no longer control the parameter generation procedure, and that it gets no more information about the master secret key than that leaked directly by the master public key (in particular, it does not have access to a token oracle). Clearly in this definition the simulator has access only to public information and a set of images, which matches the strong intuition of security.

```
Game SS-Real_{FE,R,A,D}(λ):
    $\mathsf{FuncList} \leftarrow [\,]$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M}, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $(\mathsf{m}, h, t) \leftarrow\!\!\$\ \mathcal{M}$
    $\mathsf{c} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{m}, \mathsf{Mpk})$
    $v \leftarrow\!\!\$\ \mathcal{A}_2^{\mathbf{Token}}(\mathsf{c}, h, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$

oracle Token(f):
    $\mathsf{TK} \leftarrow\!\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $\mathsf{TK}$
```

```
Game SS-Ideal_{FE,R,S,D}(λ):
    $\mathsf{FuncList} \leftarrow [\,];\ \mathsf{m} \leftarrow \perp$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\!\$\ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M}, \mathsf{st}) \leftarrow\!\!\$\ \mathcal{S}_1^{\mathbf{Eval}}(\mathsf{Mpk})$
    $(\mathsf{m}, h, t) \leftarrow\!\!\$\ \mathcal{M}$
    $\mathsf{ImgList} \leftarrow [f(\mathsf{m}) : f \in \mathsf{FuncList}]$
    $v \leftarrow\!\!\$\ \mathcal{S}_2^{\mathbf{Eval}}(\mathsf{ImgList}, h, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$

oracle Eval(f):
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $f(\mathsf{m})$
```

Fig. 11. Games defining the "strong" semantic security of an FE scheme.

It is straightforward to show that this definition is unrealizable for an FE scheme that supports at least one function. Indeed, consider a real-world adversary that requests a token for a function f and outputs $v := (\mathsf{TK}_f, f)$. Define a distinguisher which checks if TK_f is well formed by encrypting a random message m, decrypting the resulting ciphertext using TK_f, and checking equality with $f(\mathsf{m})$. It is clear that any simulator that could emulate this sequence of calls (which it would have to do so the traces are indistinguishable) can be used to extract a valid token, leading to a break of security.

D Proof of the Composition Theorem

We start by formally defining multi-message semantic security.

Definition 11 (Multi-Message Semantic Security). *Let games* $\mathsf{M\text{-}SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A}}$ *and* $\mathsf{M\text{-}SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S}}$ *be as shown in Figure 12.*[6] *The semantic security of an FE scheme relative to potential leakage relation* R *requires that for any PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, there exists a legitimate PPT simulator* $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ *such that for all PPT distinguishers* $\mathcal{D}$ *the following advantage function is negligible.*

$$\mathbf{Adv}^{\mathsf{m\text{-}ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) := \Pr\left[\mathsf{M\text{-}SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{M\text{-}SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right]$$

We are now ready to prove the composition theorem.

[6] When we say that a relation R holds for $\mathsf{R}(\mathbf{m}_0, \mathbf{m}_1)$ we mean that the relation holds point-wise on *all* elements.

```
Game $\mathsf{M\text{-}SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda)$:
    $\mathsf{FuncList} \leftarrow [\,]$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M}, \mathsf{st}) \leftarrow\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $(\mathbf{m}, h, t) \leftarrow\!\$\ \mathcal{M}$
    $\mathbf{c} \leftarrow\!\$\ \mathsf{Enc}(\mathbf{m}, \mathsf{Mpk})$
    $v \leftarrow\!\$\ \mathcal{A}_2^{\mathbf{Token}}(\mathbf{c}, h, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$

oracle Token($f$):
    $\mathsf{TK} \leftarrow\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $\mathsf{TK}$

Game $\mathsf{M\text{-}SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{S}}(\lambda)$:
    $\mathsf{FuncList} \leftarrow [\,]$; $\mathbf{m} \leftarrow \perp$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M}, \mathsf{st}) \leftarrow\!\$\ \mathcal{S}_1^{\mathbf{Eval}}(\mathsf{Mpk})$
    $(\mathbf{m}, h, t) \leftarrow\!\$\ \mathcal{M}$
    $\mathsf{ImgList} \leftarrow [f(\mathbf{m}) : f \in \mathsf{FuncList}]$
    $v \leftarrow\!\$\ \mathcal{S}_2^{\mathbf{Eval}}(\mathsf{ImgList}, h, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$

oracle Eval($f$):
    $\mathsf{TK} \leftarrow\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $(\mathsf{TK}, f(\mathbf{m}))$
```

Fig. 12. Games defining the multi-message semantic security of an FE scheme under restricted adaptive token-extraction attacks. An adversary is legitimate if: 1) $\mathsf{R}(\mathbf{m}_0, \mathbf{m}_1)$ holds for every pair of messages in $[\mathcal{M}]_1$; 2) for all second-stage **Token** queries f, we have that $f(\mathbf{m}_0) = f(\mathbf{m}_1)$ for all $\mathbf{m}_0, \mathbf{m}_1 \in [\mathcal{M}]_1$; and 3) in the token non-adaptive model, $\mathcal{A}_2$ and $\mathcal{S}_2$ do not call **Token** and **Eval** respectively.

Proof. The proof proceeds as a sequences of game where we show for any adversary $\mathcal{A}_i$ in Game^i there is an adversary $\mathcal{A}_{i+1}$ in Game^{i+1} which does as well as $\mathcal{A}_i$ in attacking the scheme. Moreover, Game^0 is defined such that it corresponds to be the multi-message real-world environment, while the final game Game^q will correspond to the multi-message ideal-world environment. This would prove the theorem. So, for $i = 0, \ldots, q$, where q is an upper bound on the number of messages that are output in the multi-message game, we define a series of games Game^i as shown below.

```
Game $\mathsf{Game}^i_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}(\lambda)$:
    $\mathsf{FuncList} \leftarrow [\,]$; $\mathbf{m} \leftarrow \perp$
    $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow\!\$\ \mathsf{Setup}(1^\lambda)$
    $(\mathcal{M}, \mathsf{st}) \leftarrow\!\$\ \mathcal{A}_1^{\mathbf{Token}}(\mathsf{Mpk})$
    $(\mathbf{m}, h, t) \leftarrow\!\$\ \mathcal{M}$
    $\mathbf{c} \leftarrow\!\$\ \mathsf{Enc}([\mathbf{m}]_1^{q-i}, \mathsf{Mpk})$
    $\mathsf{ImgList} \leftarrow [f([\mathbf{m}]_{q-i+1}^{q}) : f \in \mathsf{FuncList}]$
    $v \leftarrow\!\$\ \mathcal{A}_2^{\mathbf{Token}}(\mathbf{c}, \mathsf{ImgList}, h, \mathsf{st})$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$

oracle Token($f$):
    $\mathsf{TK} \leftarrow\!\$\ \mathsf{TKGen}(f, \mathsf{Msk})$
    $\mathsf{FuncList} \leftarrow f : \mathsf{FuncList}$
    Return $(\mathsf{TK}, f([\mathbf{m}]_{q-i+1}^{q}))$
```

For a given adversary $\mathcal{A}_i$ in Game^i, we construct a single-message adversary $\mathcal{B}_i$ as shown below.

```
Adversary $\mathcal{B}_{i,1}(\mathsf{Mpk})$:
    Run $\mathcal{A}_{i,1}(\mathsf{Mpk})$
    Token($f$) queries:
        Query own Token($f$) to get $\mathsf{TK}$
        Return $(\mathsf{TK}, \perp)$
    $\mathcal{A}_{i,1}$ terminates with $(\mathcal{M}, \mathsf{st})$
    Construct $\mathcal{M}'$ as shown
    Return $(\mathcal{M}', \mathsf{st})$

Adversary $\mathcal{B}_{i,2}(\mathsf{c}, h', \mathsf{st})$:
    $(\tilde{\mathbf{c}}, \mathsf{ImgList}, h) \leftarrow h'$
    $(\mathbf{m'}, \cdot, \cdot) \leftarrow\!\$\ \mathcal{M}$
    Run $\mathcal{A}_{i,2}([\tilde{\mathbf{c}}, \mathsf{c}], \mathsf{ImgList}, h, \mathsf{st})$
    Token($f$) queries:
        Query own Token($f$) to get $\mathsf{TK}$
        Return $(\mathsf{TK}, f([\mathbf{m'}]_{q-i+1}^{q}))$
    $\mathcal{A}_{i,2}$ terminates with $v$
    Return $v$

Algorithm $\mathcal{M}'$:
    $(\mathbf{m}, h, t) \leftarrow\!\$\ \mathcal{M}$
    $\tilde{\mathbf{c}} \leftarrow\!\$\ \mathsf{Enc}([\mathbf{m}]_1^{q-i-1}, \mathsf{Mpk})$
    $\mathsf{ImgList} \leftarrow [f([\mathbf{m}]_{q-i+1}^{q}) :$
                                $f \in \mathsf{FuncList}]$
    $h' \leftarrow (\tilde{\mathbf{c}}, \mathsf{ImgList}, h)$
    $t' \leftarrow (\mathcal{M}, t)$
    Return $([\mathbf{m}]_{q-i}, h', t')$
```

For any distinguisher $\mathcal{D}$ in Game^i we define a single-message distinguisher $\mathcal{D}'$ as shown below.

```
Distinguisher $\mathcal{D}'(\mathsf{trace}', v)$:
    $(\mathsf{Mpk}, \mathcal{M}', t', \mathsf{FuncList}) \leftarrow \mathsf{trace}'$
    $(\mathcal{M}, t) \leftarrow t'$
    $\mathsf{trace} \leftarrow (\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList})$
    Return $\mathcal{D}(\mathsf{trace}, v)$
```

By code expansion we get that

$$\Pr\left[\mathsf{Game}^i_{\mathsf{FE},\mathsf{R},\mathcal{A}_i,\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{B}_i,\mathcal{D}'}(\lambda) \Rightarrow \mathsf{T}\right] .$$

By the single-message semantic security of the scheme, we are guaranteed that a simulator $\mathcal{S}_i$ replicates $\mathcal{B}_i$'s attack, while getting a list of images $[f([\mathbf{m}]_{q-i}) \ : \ f \in \mathsf{FuncList}\,]$ rather than the ciphertext c. Therefore for any $\mathcal{D}'$ we have that

$$\Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{B}_i,\mathcal{D}'}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S}_i,\mathcal{D}'}(\lambda) \Rightarrow \mathsf{T}\right] + \mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathsf{R},\mathcal{B}_i,\mathcal{S}_i,\mathcal{D}'}(\lambda)\,.$$

Taking the simulator $\mathcal{S}_i$, we construct a hybrid adversary $\mathcal{A}_{i+1}$ in Game^{i+1} as shown below.

```
Adversary $\mathcal{A}_{i+1,1}(\mathsf{Mpk})$:
Run $\mathcal{S}_{i,1}(\mathsf{Mpk})$
Eval$(f)$ queries:
    Query own Token$(f)$ to get $(\mathsf{TK}, \bot)$
    Return $(\mathsf{TK}, \bot)$
$\mathcal{S}_{i,1}$ terminates with $(\mathcal{M}', \mathsf{st})$
Construct $\mathcal{M}$ as shown
Return $(\mathcal{M}, \mathsf{st})$
```

```
Adversary $\mathcal{A}_{i+1,2}(\mathbf{c}, \mathsf{ImgList}, h, \mathsf{st})$ :
Run $\mathcal{S}_{i,2}([\mathsf{ImgList}]_{q-i}, [\mathbf{c}, [\mathsf{ImgList}]^{q}_{q-i+1}, h], \mathsf{st})$
Eval$(f)$ queries:
    Query own Token$(f)$ to get $(\mathsf{TK}, z)$
    Return $(\mathsf{TK}, z)$
$\mathcal{S}_{i,2}$ terminates with $v$
Return $v$
```

```
Algorithm $\mathcal{M}$:
$([\mathbf{m}]_{q-i}, h', t') \leftarrow\!\!\$\ \mathcal{M}'$
$(\mathcal{M}, t) \leftarrow t'$
$(\mathbf{m}, h, t) \leftarrow\!\!\$\ \mathcal{M}$
Return $(\mathbf{m}, h, t)$
```

Now, observing that the message distribution passed to $\mathcal{D}$ inside $\mathcal{D}'$ is identical to the message distribution output by $\mathcal{A}_{i+1,1}$, we get by code expansion that for any $\mathcal{D}'$ which is built from a $\mathcal{D}$

$$\Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S}_i,\mathcal{D}'}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{Game}^{i+1}_{\mathsf{FE},\mathsf{R}\mathcal{A}_{i+1},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right]\,.$$

Therefore, putting the above together we obtain that for any adversary $\mathcal{A}_i$ and distinguisher $\mathcal{D}$ in Game_i, there is an adversary $\mathcal{A}_{i+1}$ against Game_{i+1} such that

$$\Pr\left[\mathsf{Game}^{i}_{\mathsf{FE},\mathsf{R},\mathcal{A}_i,\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{Game}^{i+1}_{\mathsf{FE},\mathsf{R},\mathcal{A}_{i+1},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] + \nu_i(\lambda)\,,$$

where ν_i are negligible. It is easy to check that our requirements on the execution time of a single-message simulator guarantees that the final adversary $\mathcal{A}_q$ also runs in polynomial time. Putting this sequence of inequalities together for $0 \le i \le q$, the theorem follows. □

E O'Neill's Preimage Samplability

We recall the definition of preimage samplability due to O'Neill [O'N10].

Definition 12 (O'Neill's Preimage Samplability). *We say a function space* $\mathsf{FunSp}(1^\lambda)$ *is O'Neill preimage samplable if there exists a sampling algorithm* Samp *such that for any algorithm* $\mathcal{A}$ *the following advantage is negligible, where game* $\mathsf{ON\text{-}PS}_{\mathsf{FunSp},\mathsf{Samp},\mathcal{A}}$ *is given in Figure 13.*

$$\mathbf{Adv}^{\mathsf{on\text{-}ps}}_{\mathsf{FunSp},\mathsf{Samp},\mathcal{A}}(\lambda) := \Pr\left[\mathsf{ON\text{-}PS}_{\mathsf{FunSp},\mathsf{Samp},\mathcal{A}}(\lambda) \Rightarrow \mathsf{F}\right]\,.$$

```
Game $\mathsf{ON\text{-}PS}_{\mathsf{FunSp},\mathsf{Samp},\mathcal{A}}(\lambda)$:
$(\mathsf{m}_0, [f_j]^n_{j=1}) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{FunSp}(1^\lambda))$
$\mathsf{m}_1 \leftarrow\!\!\$\ \mathsf{Samp}(1^\lambda, |\mathsf{m}_0|, [(f_j, f_j(\mathsf{m}_0))]^n_{j=1})$
If $|\mathsf{m}_1| \neq |\mathsf{m}_0|$ Return $\mathsf{F}$
If $(\exists j \ : \ f_j(\mathsf{m}_0) \neq f_j(\mathsf{m}_1))$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

Fig. 13. Game defining O'Neill's preimage samplability.

O'Neill's preimage samplability is defined for a function space. Whenever FE is a functional encryption scheme with a fixed function space, we say it is O'Neill preimage samplable if its function space is O'Neill preimage samplable.

F Proof of Theorem 6

Proof. For the first part of the theorem, suppose the FE scheme is not $\mathsf{IND\text{-}CPA}$-secure. We show that it cannot be $\mathsf{SS\text{-}CPA}$-secure. Take any $\mathsf{IND\text{-}CPA}$ adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, and construct an $\mathsf{SS\text{-}Real}$ adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows. Algorithm $\mathcal{B}_1$ runs $\mathcal{A}_1$. Whenever $\mathcal{A}_1$ queries its **Token** oracle, $\mathcal{B}_1$ makes the same query, providing the answer to $\mathcal{A}_1$. When $\mathcal{A}_1$ outputs two messages $(\mathsf{m}_0, \mathsf{m}_1)$ and a state $:$, algorithm $\mathcal{B}_1$ constructs the message distribution $\mathcal{M}$ that assigns probability $1/2$ to each of these messages and returns $h = \perp$ and $t = \mathsf{m}$, where m is the sampled message. It then outputs $\mathcal{M}$ and its own state, which includes the states of $\mathcal{A}_1$ and $\mathcal{M}$. Algorithm $\mathcal{B}_2$ resumes $\mathcal{A}_2$ (using the stored state), also passing it the challenge ciphertext that it receives. When $\mathcal{A}_2$ performs an adaptive **Token** oracle query, $\mathcal{B}_2$ makes the same query, providing the answer to $\mathcal{A}$. When $\mathcal{A}_2$ outputs a bit b', algorithm $\mathcal{B}_2$ outputs $\mathsf{m}_{b'}$. Note that the restrictions on $\mathcal{A}$'s queries in the $\mathsf{IND\text{-}CPA}$ game ensure that adversary $\mathcal{B}$ is legitimate with respect to R. We now define a distinguisher $\mathcal{D}$ as follows

$$\mathcal{D}((\mathsf{Mpk}, \mathcal{M}, t, \mathsf{FuncList}), v) := (v = t \ \wedge \ \forall f \in \mathsf{FuncList},\ f(\mathsf{m}_0) = f(\mathsf{m}_1)) \ ,$$

where m_0 and m_1 are recovered from the support of the message distribution. Given the legitimacy of adversary $\mathcal{A}$, we therefore have that

$$\Pr\left[\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{A}}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{B},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] \ .$$

Now, since we assume that the FE scheme is semantically secure, we know there exists a simulator $\mathcal{S}$ that produces a pair (trace, v) that is indistinguishable from that occurring in the real world to $\mathcal{D}$. We also know that $\mathcal{D}$ will only return true in the ideal-world experiment if the message distribution is correctly formed and $f(\mathsf{m}_0) = f(\mathsf{m}_1)$ for all functions. However, it is clear that in this case the simulator receives no information about b. We must have that, for any $\mathcal{S}$,

$$\Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] \leq 1/2 \, .$$

The first part of the theorem follows from the combination of this equation with the one we obtained for the real world and from the definition of $\mathsf{IND\text{-}CPA}$ advantage.

For the second part, assume that the scheme is $\mathsf{IND\text{-}CPA}$-secure and preimage samplable. We show that it is also $\mathsf{SS\text{-}CPA}$-secure. Take any adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ against $\mathsf{SS\text{-}Real}$. We construct an $\mathsf{SS\text{-}Ideal}$ simulator $\mathcal{S}$ with a similar advantage as follows. $\mathcal{S}_1$ runs $\mathcal{A}_1$ keeping track of all its inputs, outputs and interactions with the **Token** oracle in its state (if $\mathcal{A}_1$ calls the **Token** oracle, then $\mathcal{S}_1$ uses its own oracle to provide the answers to the adversary). The message distribution output of $\mathcal{S}_1$ is identical to that output by $\mathcal{A}_1$.

Before defining $\mathcal{S}_2$, let us define the following PS adversary $\mathcal{C}[\mathcal{A}_1]$ associated to $\mathcal{A}_1$:[7]

- $\mathcal{C}$ runs $\mathcal{A}_1$, answering $\mathcal{A}_1$'s token-extraction queries using the master secret key provided to it, and constructing a list of queried functions $[f_j]_{j=1}^n$.
- When $\mathcal{A}_1$ outputs $\mathcal{M}'$, algorithm $\mathcal{C}$ defines $\mathcal{M}$ as the message distribution that runs $\mathcal{M}'$ and outputs the first component m.
- $\mathcal{C}$ then outputs $\mathcal{M}$ and the list of functions $[f_j]_{j=1}^n$ queried by $\mathcal{A}_1$.

Note that $\mathcal{C}[\mathcal{A}_1]$ is legitimate as $\mathcal{A}_1$ is. Now since FE is preimage samplable, we know there exists a sampler Samp that succeeds with overwhelming probability in the PS game against $\mathcal{C}[\mathcal{A}_1]$.

[7] By structuring the argument in this way, we are establishing a stronger result by letting the sampler to depend on the adversary.

We define algorithm $\mathcal{S}_2$ using this sampler Samp. On input $(\mathsf{ImgList}, h, \mathsf{st})$, algorithm $\mathcal{S}_2$ samples a message m' using $\mathsf{Samp}(\mathcal{M}, [(f_j, f_j(\mathsf{m}))]_{j=1}^{n}, \mathsf{Mpk})$. It encrypts m' to obtain c, and runs $\mathcal{A}_2$ on $(\mathsf{c}, h, \ : [\mathcal{A}_1])$. If $\mathcal{A}_2$ calls the **Token** oracle, then $\mathcal{S}_2$ uses its own oracle to provide the answers to the adversary. When $\mathcal{A}_2$ outputs v, algorithm $\mathcal{S}_2$ also outputs this and terminates.

We show that, for any distinguisher $\mathcal{D}$, the simulator constructed above does as well as the adversary. To this end, we define an $\mathsf{IND\text{-}CPA}$ security adversary $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ as follows. Algorithm $\mathcal{B}_1$ runs $\mathcal{A}_1$, answering its **Token** queries using its own oracle. It also keeps a list of queried functions $[f_j]_{j=1}^{n}$. When it obtains a message distribution $\mathcal{M}$, it samples a tuple (m_0, h_0, t_0) from $\mathcal{M}$ and constructs a list of images $[f_j(\mathsf{m}_0)]_{j=1}^{n}$. Next it samples a message m_1 using $\mathsf{Samp}(\mathcal{M}, [f_j, f_j(\mathsf{m}_0)]_{j=1}^{n}, \mathsf{Mpk})$. Algorithm $\mathcal{B}_1$ checks if the functions collides on $(\mathsf{m}_0, \mathsf{m}_1)$ and $\mathsf{R}(\mathsf{m}_0, \mathsf{m}_1)$ holds. It stops by returning a random guess if these are not the case. Otherwise it outputs $(\mathsf{m}_0, \mathsf{m}_1)$ to **LR** and keeps the information it collected as its state. In the second stage, $\mathcal{B}_2$ runs $\mathcal{A}_2$ on $(\mathsf{c}, h_0, \ : [\mathcal{A}_1])$ as follows. If $\mathcal{A}_2$ calls the **Token** oracle, then $\mathcal{B}_2$ first checks if the function collides on $(\mathsf{m}_0, \mathsf{m}_1)$, and again stops early by returning a random guess if this is not the case. Otherwise, it uses its own **Token** oracle to provide a token for the adversary. When $\mathcal{A}_2$ outputs v, adversary $\mathcal{B}_2$ outputs $\mathcal{D}((\mathsf{Mpk}, \mathcal{M}, t_0, \mathsf{FuncList}), v)$.

Conditioned on the event E that the adversary $\mathcal{B}$ does not stop early (whose probability we bound below), it is easy to see that $\mathcal{B}$ is legitimate and furthermore if the bit chosen in the $\mathsf{IND\text{-}CPA}$ game is 0, we obtain an output v as in the real game. In case it is 1, we get an output v as in the ideal game with a simulator as defined above. Therefore

$$\Pr\left[\mathsf{SS\text{-}Real}_{\mathsf{FE},\mathsf{R},\mathcal{A},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{B}}(\lambda) \Rightarrow \mathsf{T} \mid b = 0 \ \wedge \ \mathsf{E}\right]$$
$$\Pr\left[\mathsf{SS\text{-}Ideal}_{\mathsf{FE},\mathsf{R},\mathcal{S},\mathcal{D}}(\lambda) \Rightarrow \mathsf{T}\right] = \Pr\left[\mathsf{IND\text{-}CPA}_{\mathsf{FE},\mathsf{R},\mathcal{B}}(\lambda) \Rightarrow \mathsf{T} \mid b = 1 \ \wedge \ \mathsf{E}\right] .$$

We now bound the probability that adversary $\mathcal{B}$ stops early. Observe that when this happens, the PS adversary $\mathcal{C}[\mathcal{A}_1]$ breaks Samp. Indeed, there can be three reasons for a premature termination: 1) first-stage functions do not collide on the messages; 2) R is not satisfied by the messages; or 3) second-stage functions do not collide on the messages. All of these would immediately translate to $\mathcal{C}[\mathcal{A}_1]$ causing Samp to return a message that does not satisfy one of the checks in the PS game (note that we need the legitimacy of $\mathcal{A}_2$ to argue for this for the third case). We therefore have

$$\Pr\left[\neg\mathsf{E}\right] \leq \mathbf{Adv}^{\mathsf{res\text{-}ps}}_{\mathsf{FE},\mathsf{R},\mathcal{C},\mathsf{Samp}}(\lambda) .$$

The theorem follows by an application of the fundamental lemma of game-playing [BR06]. □

G Proof of Theorem 3

Proof. We define a CPS sampler as shown in Figure 14.

```
Algorithm Samp(trace):
(Mpk, $\mathcal{M}$, FuncList, ImgList) ← trace
If m ∈ ImgList s.t. m ≠⊥ Return m
Do
    m' ←$ $\mathcal{M}$
    ImgList' ← [$f$(m') : $f$ ∈ FuncList]
    If (ImgList = ImgList') Return m'
```

Fig. 14. CPS sampler for ANOT functionalities.

Note that Samp is always successful on termination. We show that Samp terminates in expected polynomial time. Let p_{trace} be the probability that the image values contain the message in the clear, when one conditions on the occurrence of a specific trace. If $p_{\mathsf{trace}} = 1$ then the number of loop iterations is zero. When $p_{\mathsf{trace}} < 1$, we have

$$\mathbb{E}\left[\text{number of iterations} \mid \mathsf{trace}\right] = 0 \cdot p_{\mathsf{trace}} + (1 - p_{\mathsf{trace}}) \cdot \sum_{n=1}^{\infty} n \cdot p_{\mathsf{trace}}^{n-1}(1 - p_{\mathsf{trace}}) = 1\,.$$

Therefore

$$\mathbb{E}\left[\text{number of iterations}\right] = \sum_{p_{\mathsf{trace}}<1} 1 \cdot p_{\mathsf{trace}} \leq 1\,.$$

Furthermore, the sampler always returns a message that is identically distributed to the actual message used in calculating the image list as it sampled from $\mathcal{M}$ conditioned on colliding images. □

The proof for Theorem 5 uses exactly the same sampler as define above. However note that if the functionality is JNOT with respect to R, then the legitimacy of $\mathcal{A}$ essentially implies that Samp will either get m upfront in the image list, or it is guaranteed to succeed in sampling a valid message in a single iteration of the look. This is because the JNOT property says that, whatever rejection pattern you obtain for a message in the support of $\mathcal{M}$, this will be the same for *all* messages in the support of $\mathcal{M}$.

On strict polynomial-time samplers. Let us define a strict PPT sampler that samples a total of at most $q(\lambda)$ messages, for an a priori fixed q, as above, and fails if no correct preimage is found. We first show here that such sampler cannot be "universal," i.e., it cannot succeed with an overwhelming probability against *all* PPT adversaries. To this end, consider the adversary which returns a distribution and a function as follows.

$$\mathcal{M}_{q,\lambda} := \begin{cases} \mathsf{m}_0 & \text{with prob. } 1 - \frac{1}{q(\lambda)}\,; \\ \mathsf{m}_1 & \text{with prob. } \frac{1}{q(\lambda)}\,. \end{cases} \qquad f_{\mathsf{m}_0}(\mathsf{m}) := \begin{cases} \mathsf{m} & \text{if } \mathsf{m} = \mathsf{m}_0\,; \\ \bot & \text{otherwise.} \end{cases}$$

Here m_0 and m_1 are two fixed messages in the message space. The probability of the failure of the sampler against this adversary is

$$\Pr\left[\text{Fail}\right] = (1 - \frac{1}{q(\lambda)})^{q(\lambda)} \cdot \frac{1}{q(\lambda)} = \Theta\left(\frac{1}{q(\lambda)}\right)\,,$$

which is noticeable. Now recall that our definition of preimage samplability allows the sampler to *depend* on the adversary. This dependency might be sufficient to allow for a (non-universal) PPT sampler. (Indeed, for PS adversaries as given above such a sampler exists.) Despite this, we also rule out the existence of samplers which follow the above rejection sampling strategy. Consider the following adversary which "diagonalizes" over all distributions $\mathcal{M}_{q,\lambda}$. On input the security parameter λ, the adversary samples $i \leftarrow_{\$} \{1, \ldots, \lambda\}$ and returns $(\mathcal{M}_{i,\lambda}, f_{\mathsf{m}_0})$, where

$$\mathcal{M}_{c,\lambda} := \begin{cases} \mathsf{m}_0 & \text{with prob. } 1 - \frac{1}{\lambda^c}\,; \\ \mathsf{m}_1 & \text{with prob. } 1 - \frac{1}{\lambda^c}\,. \end{cases} \qquad \text{and} \quad f_{\mathsf{m}_0}(\mathsf{m}) := \begin{cases} \mathsf{m} & \text{if } \mathsf{m} = \mathsf{m}_0\,; \\ \bot & \text{otherwise.} \end{cases}$$

Note that $|\lambda^\lambda| \in \mathcal{O}(\lambda^2)$, and this adversary runs in polynomial time. Now for *any* constant d, the failure probability of the sampler that performs at most λ^d trials is

$$\Pr\left[\text{Fail}\right] = \sum_{i=1}^{\lambda}(1 - \frac{1}{\lambda^i})^{\lambda^d} \cdot \frac{1}{\lambda^i} \cdot \frac{1}{\lambda} = \Omega\left(\frac{1}{\lambda^{d+1}}\right)\,,$$

which is noticeable.

We note that this result does not exclude the existence of samplers that take advantage of extra (non-black-box) properties of the message distributions. For instance the condition that the potential leakage relation is true for any pair of messages in the support of the returned distributions, which is the case for, say, non-anonymous IBE schemes, leads to a strict PPT sampler.

H Proof of Theorem 8

In what follows below n, m, q, d, and β are polynomials in a security parameter λ. However, in order to ease notation we drop the explicit dependency on λ and write, for example, m for $m(\lambda)$. The polynomials n, m, and d take values in $\mathbb{N}$, q takes values in the set of primes, and $\beta \in \mathbb{R}^+$. We also consider $\mathbb{Z}_q$, for an odd q, to be encoded in the range $\mathrm{B}(q/2) := [-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$.

Definition 13 (The Small Integer Solution (SIS) Problem). *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix sampled uniformly at random from $\mathbb{Z}_q^{n\times m}$. The (q,m,n,β)-SIS problem consists of, given $\mathbf{A}$, finding a nonzero vector $\boldsymbol{x} \in \mathbb{Z}_q^m$ of ℓ_2-norm at most β satisfying $\mathbf{A}\boldsymbol{x} = \mathbf{0} \mod q$. Formally, we define the advantage of an adversary $\mathcal{A}$ against this problem as*

$$\mathbf{Adv}_{\mathcal{A}}^{(q,m,n,\beta)\text{-}\mathsf{sis}}(\lambda) := \Pr\left[\mathbf{A}\boldsymbol{x} = \mathbf{0} \mod q \ \wedge\ \|\boldsymbol{x}\|_2 \le \beta \ :\ \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\ \boldsymbol{x} \leftarrow_{\$} \mathcal{A}(\mathbf{A})\right] .$$

Typical parameters for the SIS problem to be hard are: $q = n^{\mathcal{O}(1)}$, $m = \Theta(n \log q)$, and $\beta = \sqrt{m} \cdot q^{n/m}$.

Definition 14 (The Decisional Small Integer Solution (DSIS) Problem). *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix sampled uniformly at random from $\mathbb{Z}_q^{n\times m}$. Let $\boldsymbol{t}$ be uniform in $\mathbb{Z}_q^n$ and $\boldsymbol{x}$ be a uniform vector in $\mathrm{B}(d)^m$. The (q,m,n,d)-DSIS problem consists of distinguishing the distributions $(\mathbf{A}, \mathbf{A}\boldsymbol{x})$ and $(\mathbf{A}, \boldsymbol{t})$. Formally, we define the advantage of an adversary $\mathcal{A}$ against this problem as*

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{(q,m,n,d)\text{-}\mathsf{dsis}}(\lambda) := \Pr&\left[b = 0 \ :\ \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\ \boldsymbol{x} \leftarrow_{\$} \mathrm{B}(d)^m;\ b \leftarrow_{\$} \mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{x})\right] \\ &- \Pr\left[b = 0 \ :\ \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\ \boldsymbol{t} \leftarrow_{\$} \mathbb{Z}_q^n;\ b \leftarrow_{\$} \mathcal{A}(\mathbf{A}, \boldsymbol{t})\right] .\end{aligned}$$

Typical parameters for the DSIS problem to be hard are $q = n^{\mathcal{O}(1)}$, $m = \Theta(n \log q)$, and $d = q^{n/m}$.

We now prove Theorem 8.

Proof. We can rewrite the restricted PS experiment with respect to the adversary $\mathcal{A}$ in Figure 9 in the following simplified form, which we will call Game_0.

```
$\mathsf{Game}_0(\lambda)$:
$\mathbf{Y} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$
$\boldsymbol{x} \leftarrow_{\$} \mathrm{B}(d)^m$
$\boldsymbol{t} \leftarrow \mathbf{Y}\boldsymbol{x}$
$\mathsf{Zeros} \leftarrow \{\ i\ :\ \boldsymbol{t}_i = 0 \wedge 1 \le i \le n\ \}$
$\boldsymbol{x}' \leftarrow_{\$} \mathsf{Samp}(\mathbf{Y}, \mathsf{Zeros})$
$\boldsymbol{t}' \leftarrow \mathbf{Y}\boldsymbol{x}'$
$\mathsf{Zeros}' \leftarrow \{\ i\ :\ \boldsymbol{t}'_i = 0 \wedge 1 \le i \le n\ \}$
If $\mathsf{Zeros} \ne \mathsf{Zeros}' \vee \boldsymbol{x}' \notin \mathrm{B}(d)^m$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

We now modify Game_0 by sampling $\boldsymbol{t}$ uniformly at random to obtain Game_1.

```
$\mathsf{Game}_1(\lambda)$:
$\mathbf{Y} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$
$\boxed{\boldsymbol{t} \leftarrow_{\$} \mathbb{Z}_q^n}$
$\mathsf{Zeros} \leftarrow \{\ i\ :\ \boldsymbol{t}_i = 0 \wedge 1 \le i \le n\ \}$
$\boldsymbol{x}' \leftarrow_{\$} \mathsf{Samp}(\mathbf{Y}, \mathsf{Zeros})$
$\boldsymbol{t}' \leftarrow \mathbf{Y}\boldsymbol{x}'$
$\mathsf{Zeros}' \leftarrow \{\ i\ :\ \boldsymbol{t}'_i = 0 \wedge 1 \le i \le n\ \}$
If $\mathsf{Zeros} \ne \mathsf{Zeros}' \vee \boldsymbol{x}' \notin \mathrm{B}(d)^m$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

We show that any change in the sampler's behavior can be translated into an attack on the (q, m, n, d)-DSIS problem by constructing adversary $\mathcal{B}_1$ as follows.

```
$\underline{\mathcal{B}_1(\mathbf{A}, \boldsymbol{t})}$:
$\mathbf{Y} \leftarrow \mathbf{A}$
$\mathsf{Zeros} \leftarrow \{\ i\ :\ \boldsymbol{t}_i = 0 \wedge 1 \le i \le n\ \}$
$\boldsymbol{x}' \leftarrow\$\ \mathsf{Samp}(\mathbf{Y}, \mathsf{Zeros})$
$\boldsymbol{t}' \leftarrow \mathbf{Y}\boldsymbol{x}'$
$\mathsf{Zeros}' \leftarrow \{\ i\ :\ \boldsymbol{t}'_i = 0 \wedge 1 \le i \le n\ \}$
If $\mathsf{Zeros} \ne \mathsf{Zeros}' \vee \boldsymbol{x}' \notin \mathrm{B}(d)^m$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

It is clear that by construction, we have

$$\Pr[\mathsf{Game}_0(\lambda) \Rightarrow \mathsf{T}] - \Pr[\mathsf{Game}_1(\lambda) \Rightarrow \mathsf{T}] = \mathbf{Adv}^{(q,m,n,d)\text{-}\mathsf{dsis}}_{\mathcal{B}_1}(\lambda)\,.$$

We now modify Game_1 so that the sampler immediately loses the game if there are at most n' zero positions in $\boldsymbol{t}$. This gives Game_2.

```
$\underline{\mathsf{Game}_2(\lambda)}$:
$\mathbf{Y} \leftarrow\$\ \mathbb{Z}_q^{n \times m}$
$\boldsymbol{t} \leftarrow\$\ \mathbb{Z}_q^{n}$
$\mathsf{Zeros} \leftarrow \{\ i\ :\ \boldsymbol{t}_i = 0 \wedge 1 \le i \le n\ \}$
$\boxed{\text{If } |\mathsf{Zeros}| \le n' \text{ Return } \mathsf{F}}$
$\boldsymbol{x}' \leftarrow\$\ \mathsf{Samp}(\mathbf{Y}, \mathsf{Zeros})$
$\boldsymbol{t}' \leftarrow \mathbf{Y}\boldsymbol{x}'$
$\mathsf{Zeros}' \leftarrow \{\ i\ :\ \boldsymbol{t}'_i = 0 \wedge 1 \le i \le n\ \}$
If $\mathsf{Zeros} \ne \mathsf{Zeros}' \vee \boldsymbol{x}' \notin \mathrm{B}(d)^m$ Return $\mathsf{F}$
Return $\mathsf{T}$
```

By the fundamental lemma of game-playing [BR06], we have that

$$\Pr[\mathsf{Game}_1(\lambda) \Rightarrow \mathsf{T}] - \Pr[\mathsf{Game}_2(\lambda) \Rightarrow \mathsf{T}] \le \Pr\left[|\mathsf{Zeros}| \le n'\right]\,.$$

Since $\boldsymbol{t}$ is sampled uniformly at random from $\mathbb{Z}_q^n$, we have that the probability of getting a zero in each position is exactly $1/q$. By standard Chernoff–Hoeffding bounds on the tails of the binomial distribution for $n' \le n/q$ we have

$$\Pr\left[|\mathsf{Zeros}| \le n'\right] \le \exp\left(-\frac{q}{2} \cdot \frac{(n/q - n')^2}{n}\right)\,.$$

Setting $n' = n/q - \sqrt{n/q}\log(n/q)$, we obtain a negligible bound. This justifies our choice of n' in the theorem statement.

To conclude the proof, we construct an algorithm $\mathcal{B}_2$ that converts any sampler succeeding with non-negligible probability in game Game_2 into a successful attacker on the (q, m, n', β)-SIS problem.

```
$\underline{\mathcal{B}_2(\mathbf{A})}$:
$\boldsymbol{t} \leftarrow\$\ \mathbb{Z}_q^{n}$
$\mathsf{Zeros} \leftarrow \{\ i\ :\ \boldsymbol{t}_i = 0 \wedge 1 \le i \le n\ \}$
If $|\mathsf{Zeros}| \le n'$ Return $\perp$
$\mathbf{B} \leftarrow\$\ \mathbb{Z}_q^{(n-n') \times m}$
$\mathbf{Y} \leftarrow [\mathbf{A}^t | \mathbf{B}^t]^t$
$\mathsf{PermuteRows}(\mathbf{Y}, n', \mathsf{Zeros})$
$\boldsymbol{x}' \leftarrow\$\ \mathsf{Samp}(\mathbf{Y}, \mathsf{Zeros})$
$\boldsymbol{t}' \leftarrow \mathbf{Y}\boldsymbol{x}'$
$\mathsf{Zeros}' \leftarrow \{\ i\ :\ \boldsymbol{t}'_i = 0 \wedge 1 \le i \le n\ \}$
If $\mathsf{Zeros} \ne \mathsf{Zeros}' \vee \boldsymbol{x}' \notin \mathrm{B}(d)^m$ Return $\perp$
Return $\boldsymbol{x}'$
```

Here, $\mathsf{PermuteRows}$ is a deterministic algorithm that permutes the rows of $\mathbf{Y}$ so that the first n' rows of the matrix are moved into the subset of row indices indicated by Zeros. Observe that, if $\mathcal{B}_2$ reaches this point of execution, there are always enough rows so that $\mathsf{PermuteRows}$ can always succeed.

It is clear that the inputs provided to Samp in the simulation are correctly distributed under the rules of Game_2. Furthermore, if the sampler succeeds, then it will have output a vector $\boldsymbol{x}'$ of norm at most $d\sqrt{m}$ that is guaranteed to satisfy $\mathbf{A}\boldsymbol{x}' = \mathbf{0}$, and therefore is a valid solution to the given $(q, m, n', d\sqrt{m})$-SIS problem instance. □

For example, based on the parameters for the hardness of the SIS and DSIS problems, for any constant $\delta \in (0, 1)$ we may set

$$n = \lambda, \quad q = \lambda^{\delta}, \quad m = \lambda^2, \quad d = \lambda^{\delta/\lambda} \approx 1, \quad n' = \lambda^{1-\delta} - \sqrt{\lambda^{1-\delta}} \log \lambda, \quad \beta \approx \lambda\,.$$

Note that our slightly larger choice for m ensures that $q^{n/m}$ and $q^{n'/m}$ are approximately 1 so that the number of columns in the SIS and DSIS problems can be the same. The smallness conditions in the two problems match, i.e., $q^{n/m}\sqrt{m} = q^{n'/m'}\sqrt{m'}$.

Large modulus. The SIS and DSIS problems are also hard when the modulus q is large [MR07]. However, in this case the argument given above for small values of q becomes vacuous as the $n' = \mathcal{O}(n/q)$ condition no longer applies (the probability of hitting zero becomes negligible). We deal with this problem by restricting the matrices to be uniform over $\mathbb{Z}_{q'}^{n\times m}$ where $q' \ll q$. This leads to the following analogues of the SIS and DSIS problems over the *integers* (where we have made the simplifying assumption that $q = \infty$).

Definition 15 (The Subset-Sum (SS) Problem). *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix sampled uniformly at random from $\mathbb{Z}_q^{n\times m}$. The (q, m, n, β)-SS problem for a given $\mathbf{A}$ consists of finding a nonzero vector $\boldsymbol{x} \in \mathbb{Z}^m$ of ℓ_2-norm at most β satisfying $\mathbf{A}\boldsymbol{x} = \mathbf{0}$ over $\mathbb{Z}$. Formally, we define the advantage of an adversary $\mathcal{A}$ as*

$$\mathbf{Adv}_{\mathcal{A}}^{(q,m,n,\beta)\text{-}\mathsf{ss}}(\lambda) := \Pr\left[\mathbf{A}\boldsymbol{x} = \mathbf{0} \textit{ in } \mathbb{Z} \;\wedge\; \|\boldsymbol{x}\|_2 \le \beta \;:\; \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\; \boldsymbol{x} \leftarrow_{\$} \mathcal{A}(\mathbf{A})\right]\,.$$

Note that the (q, m, n, β)-SS problem is at least as hard as the (q, m, n, β)-SIS problem.

Definition 16 (The Decisional Subset-Sum (DSS) Problem). *Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix sampled uniformly at random from $\mathbb{Z}_q^{n\times m}$. Define the distribution*

$$\mathcal{Z}(q, m, d) := \sum_{i=1}^{m} X_i Y_i$$

over the integers, where Y_i are uniform over $\mathbb{Z}_q$ and X_i are uniform over $\mathrm{B}(d)$. Let $\boldsymbol{t}$ and $\boldsymbol{x}$ be sampled from $Z(q, m, d)^n$ and uniformly at random from $\mathrm{B}(d)^m$ respectively. The (q, m, n, d)-DSS problem consists of distinguishing the distributions $(\mathbf{A}, \mathbf{A}\boldsymbol{x})$ and $(\mathbf{A}, \boldsymbol{t})$ when matrix multiplication is performed over the integers. Formally, we define the advantage of an adversary $\mathcal{A}$ as

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{(q,m,n,d)\text{-}\mathsf{dss}}(\lambda) := \Pr&\left[b = 0 \;:\; \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\; \boldsymbol{x} \leftarrow_{\$} \mathbb{Z}_q^{m};\; b \leftarrow_{\$} \mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{x} \textit{ in } \mathbb{Z})\right]\\ &- \Pr\left[b = 0 \;:\; \mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m};\; \boldsymbol{t} \leftarrow_{\$} \mathcal{Z}(q, m, d)^n;\; b \leftarrow_{\$} \mathcal{A}(\mathbf{A}, \boldsymbol{t})\right]\,.\end{aligned}$$

Note that if inner-product values and matrix multiplication are calculated modulo q, we recover the DSIS problem: $\mathcal{Z}(q, m, d)$ will be uniform over $\mathbb{Z}_q$. Under the hardness of the above problems, which we conjecture to be true for a set of parameters that can be embedded in the inner-product encryption preimage samplability scenario, the proof of non-samplability for large moduli closely follows that for small q. Note that the probability of hitting zero when sampling from $\mathcal{Z}$, by the central-limit theorem, is approximately $1/\sqrt{2\pi m\sigma^2}$, where $\sigma^2 = \frac{1}{9}q(q+1)d(d+1)$ is the variance of $X_i Y_i$. We defer the details to a final treatment.