# On the Relationship between Functional Encryption, Obfuscation, and Fully Homomorphic Encryption

Joël Alwen[1], Manuel Barbosa[2], Pooya Farshim[3], Rosario Gennaro[4],
S. Dov Gordon[5], Stefano Tessaro[6,7], and David A. Wilson[7]

[1] ETH Zurich
[2] HASLab – INESC TEC and Universidade do Minho
[3] Fachbereich Informatik, Technische Universität Darmstadt
[4] City University of New York
[5] Applied Communication Sciences
[6] University of California, Santa Barbara
[7] MIT

**Abstract.** We investigate the relationship between Functional Encryption (FE) and Fully Homomorphic Encryption (FHE), demonstrating that, under certain assumptions, a Functional Encryption scheme supporting evaluation on two ciphertexts implies Fully Homomorphic Encryption. We first introduce the notion of Randomized Functional Encryption (RFE), a generalization of Functional Encryption dealing with randomized functionalities of interest in its own right, and show how to construct an RFE from a (standard) semantically secure FE. For this we define the notion of entropically secure FE and use it as an intermediary step in the construction. Finally we show that RFEs constructed in this way can be used to construct FHE schemes thereby establishing a relation between the FHE and FE primitives. We conclude the paper by recasting the construction of RFE schemes in the context of obfuscation.

**Keywords.** Functional encryption, Randomized functionality, Obfuscation, Homomorphic encryption.

## 1 Introduction

The goal of this work is to draw connections between FE and FHE. The motivation is twofold. On the one hand, from a purely theoretical point of view it is interesting to understand the relationship between these new concepts, both of which promise to guide much future research in the area. On the other hand, from a constructive point of view, we hope that by exploring this relationship we can also discover connections that will help us progress towards new constructions of FHE schemes, taking advantage of techniques used in constructing FE schemes. This motivation is even stronger in light of very recent developments in the area, which indicate that constructing efficient FE schemes supporting complex functionalities may be within our reach in the near future. For example, an FE for arbitrary functionalities has been recently proposed in [13], as well as functional encryption for regular languages in [19]. Goldwasser et al. recently demonstrated that FHE together with attribute-based encryption implies

FE [12]; here we study the opposite problem, asking what additional assumptions can be used to build FHE from FE.

To explain our high-level approach to accomplishing this goal, let us consider the main differences between FE and FHE. For any function $f(\cdot)$, an FHE scheme allows us to compute *an encryption* of $f(x)$ from an encryption of $x$; whereas an FE scheme allows us to compute, *in the clear*, $f(x)$ from an encryption of $x$. Our intuition is that an FE scheme supporting functions of the type $\mathsf{Enc}_f$, where $\mathsf{Enc}_f(x) = \mathsf{Enc}(f(x))$ is a re-encryption of $f(x)$, would be very close to constructing an FHE scheme. However, an immediate problem inherent to this approach is how to provide the tokens needed for the computation of $\mathsf{Enc}_f$, when $f$ can be arbitrary. We will demonstrate that this can be done if one considers FE schemes that support functions on two inputs $f(x_1, x_2)$. In this case, it is sufficient to publish the token for $\mathsf{Enc}_{\mathsf{NAND}}(x_1, x_2) = \mathsf{Enc}(\neg(x_1 \wedge x_2))$, as any Boolean function can be computed using a circuit of NAND gates.

When we set out to formally prove this intuition we stumbled across two major definitional issues: First, the "re-encryption" functionality described above is a "randomized" functionality, a case which is not treated in the original definitions of FE [7,16]. Second, this approach to constructing FHEs from FEs relies on a functionality that re-encrypts under the original encryption key. However, this implies extending the syntax of functional encryption schemes to allow for functionalities that take the domain parameters as an additional input (i.e., where the functionality can depend on the master public key).

RANDOMIZED FUNCTIONAL ENCRYPTION. In Randomized Functional Encryption an encryptor is able to hide an input within a ciphertext in such a way that authorized decryptors can only recover the result of applying a randomized function to it. Intuitively, the encryption and/or decryption operations can take *additional randomness* that is fed into the randomized function upon decryption, ensuring that the recovered image $f(x)$ is distributed as if the randomized function was calculated on fresh randomness $r$ as $f(x; r)$. This is a correctness criterion. As for functional encryption, the security goal in RFE is to ensure that the decryption operation leaks no more about $x$ than that which could be inferred from the output of the randomized functionality with private randomness.We introduce a natural indistinguishability-based notion of security for randomized functional encryption to capture this intuition, and show that it suffices to establish a relation to fully homomorphic encryption. Indeed, we show that an $\mathsf{IND\text{-}CPA}$-secure RFE supporting the *NAND re-encryption* functionality $\mathsf{Enc}_{\mathsf{NAND}}(x_1, x_2)$ described above can be used as a secure fully homomorphic encryption scheme.

Randomized Functional Encryption is an interesting cryptographic primitive in its own right. In the full version of this paper we show that various existing public-key encryption schemes can actually be seen as particular instances of this primitive, e.g., those supporting re-randomization of ciphertexts and re-encryption. That said, our primary motivation in introducing the notion of RFE is to enable the construction of FHE from FE, so we do not attempt to fully explore the subtleties of RFE. In particular, we provide and use a very strong indistinguishability notion for RFE which is likely *too* strong to admit many interesting functionalities, but suffices for building FHE. We leave further exploration of RFE to future work.

FROM FUNCTIONAL ENCRYPTION TO RFE AND FHE. We show how FE schmes can be used to build RFE schemes, which, in turn, can be used to construct a fully homomorphic encryption scheme, as described above. To this end, we begin by extending the syntax and security definitions for functional encryption schemes as follows: 1) we modify the syntax to allow functionalities to take multiple inputs,as well as the domain parameters as an extra input; 2) we introduce a new indistinguishability notion of security for FE schemes that we call entropic security in the spirit of the eponymous definition of Russel and Wang [18]; and 3) we extend the semantic security definition of Gorbunov, Vaikuntanathan, and Wee [13] to accommodate our new syntactic extensions. We also discuss the relation between the two security notions we propose, and show that, under certain conditions, semantic security implies entropic security.

We then present a generic construction of an RFE scheme supporting a given randomized functionality $\mathsf{RFn}$ of arity $n$, from an entropically secure FE supporting a specific functionality $\mathsf{Fn}$ of the same arity. Interestingly, for binary functionalities we require a PRF that resists a particular class of related-key attacks [5,4] in order to prove the security of our construction. When combined with the semantically secure FE construction of Gorbunov, Vaikuntanathan, and Wee [13], our results for the unary case yield positive feasibility results for RFE supporting complex functionalities.

Finally, we present two constructions of FHE from functional encryption. The direct construction from an RFE supporting NAND re-encryption mentioned above gives us that entropically secure functional encryption supporting a binary functionality (and taking also the domain parameters) implies FHE. However, we cannot show that the required RFE scheme can be constructed by assuming semantic security alone. Our second construction uses two RFEs and a PKE, and it enables leveled homomorphic computation using an adaptation of the bootstrapping technique of Gentry [11] to the functional setting. We can relate the security of this construction to semantically secure functional encryption supporting functionalities of arity 2 (there is no need for the functionality to take the domain parameters). As is typical of bootstrapping techniques, an extra assumption akin to key-dependent message (KDM) security [11] allows us to use this construction as an FHE scheme.

CONNECTION WITH OBFUSCATION. As our final contribution, we explore the problem of obfuscating specific re-encryption functionalities, introducing new notions extending those proposed in earlier works on re-encryption [14,9]. We then explain how to use such obfuscated circuits to obtain RFE schemes suitable for FHE constructions.

STRUCTURE OF THE PAPER. In Section 2 we recall the syntax and security of FE schemes, and introduce a set of extensions that we require for our results. In Section 3 we introduce randomized functional encryption. In Section 4 we introduce the notion of entropic securityand, in Section 5, we present generic constructions of fully homomorphic encryption schemes from RFE schemes.Finally, the connection with obfuscated re-encryption is explored in Section 6.

## 2 Functional Encryption

NOTATION. We start by settling the notation. We write $\mathsf{x} \leftarrow \mathsf{y}$ for assigning value $\mathsf{y}$ to variable $\mathsf{x}$. We write $\mathsf{x} \leftarrow_{\$} \mathsf{X}$ for sampling $\mathsf{x}$ from set $\mathsf{X}$ uniformly at random. If $\mathsf{A}$ is a probabilistic algorithm we write $\mathsf{y} \leftarrow_{\$} \mathsf{A}(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ for the action of running $\mathsf{A}$ on inputs $\mathsf{x}_1, \ldots, \mathsf{x}_n$ with random coin chosen uniformly at random, and assigning the result to $\mathsf{y}$. We use ":" for appending to a list, and denote the empty string by $\epsilon$. For random variable $\mathsf{X}$ we denote by $[\mathsf{X}]$ the support of $\mathsf{X}$, i.e., the set of all values that $\mathsf{X}$ takes with nonzero probability. All algorithms in this paper are probabilistic polynomial-time. We say $\eta(\lambda)$ is negligible if $|\eta(\lambda)| \in \lambda^{-\omega(1)}$. Finally in various security games we use the notation $\mathcal{A}^{\mathbf{O}}$ and adversary $\mathcal{A}$ haveing oracle access to all oracles specified within the same figure as the game.

We now formalize the syntax and security of an FE scheme, extending it with respect to several features that were proposed in [7].

DETERMINISTIC FUNCTIONALITY. A deterministic functionality is an algorithm $\mathsf{Fn}$ implementing an $n$-ary function $f$ over a parameter space $\mathsf{Fn.Prms}$, a key space $\mathsf{Fn.KeySp}$, and a plaintext space $\mathsf{Fn.MsgSp}^n$:

$$f : \mathsf{Fn.Prms} \times \mathsf{Fn.KeySp} \times \mathsf{Fn.MsgSp}^n \longrightarrow \mathsf{Fn.Rng}\,.$$

We require that the key space contains a special key called the *empty key* denoted $\mathsf{k}_\epsilon$. The empty key abstracts the information that publicly leaks from a ciphertext. This function in most natural cases would only depend on a single argument, although one may envisage the more general case where it has arity

$n$. Throughout the paper we assume the messages in the message space are of equal length, and whenever the empty key is not explicitly defined, it is assumed that it returns the length of its first input.

SYNTAX. A *functional encryption (FE) scheme* $\mathsf{FE}$ for the functionality $\mathsf{Fn}$ (of arity $n$) is specified by four algorithms as follows.

1. $\mathsf{FE.Setup}(1^\lambda)$: This is the setup algorithm. On input the security parameter, this algorithm outputs a master secret key $\mathsf{Msk}$ and master public key $\mathsf{Mpk}$, a key space, and a message space. We require that the set of all possible master public keys output by the setup algorithm, the key space, and the message space are identical to $\mathsf{Fn.Prms}$, $\mathsf{Fn.KeySp}$, and $\mathsf{Fn.MsgSp}$ respectively.
2. $\mathsf{FE.TKGen}(\mathsf{k}, \mathsf{Msk})$: This is the token-generation algorithm. On input a key $\mathsf{k} \in \mathsf{Fn.KeySp}$, and a master secret key $\mathsf{Msk}$, it outputs a token $\mathsf{TK}$. We assume, without loss of generality, that the token for the empty functionality is the empty string.
3. $\mathsf{FE.Enc}(\mathsf{m}, \mathsf{Mpk})$: This is the encryption algorithm. On input a message $\mathsf{m} \in \mathsf{Fn.MsgSp}$ and a master public key $\mathsf{Mpk}$, it outputs a ciphertext $\mathsf{c}$.
4. $\mathsf{FE.Dec}(\mathsf{c}_1, \ldots, \mathsf{c}_n, \mathsf{TK}, \mathsf{Mpk})$: This is the deterministic decryption (or evaluation) algorithm. On input $n$ ciphertexts, a token $\mathsf{TK}$, and a master public key $\mathsf{Mpk}$, it outputs a message $\mathsf{m}$ or a special failure symbol $\perp$.

CORRECTNESS. We call scheme $\mathsf{FE}$ correct if, for $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow_\$ \mathsf{FE.Setup}(1^\lambda)$, any $\mathsf{m}_1, \ldots, \mathsf{m}_n \in \mathsf{Fn.MsgSp}$, randomly chosen $\mathsf{c}_i \leftarrow_\$ \mathsf{FE.Enc}(\mathsf{m}_i, \mathsf{Mpk})$ for $1 \leq i \leq n$, any $\mathsf{k} \in \mathsf{Fn.KeySp}$, and randomly chosen $\mathsf{TK} \leftarrow_\$ \mathsf{FE.TKGen}(\mathsf{k}, \mathsf{Msk})$, $\Pr[\mathsf{FE.Dec}(\mathsf{c}_1, \ldots, \mathsf{c}_n, \mathsf{TK}, \mathsf{Mpk}) \neq \mathsf{Fn}(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}_1, \ldots, \mathsf{m}_n)]$ is negligible as a function of $\lambda$.

SECURITY. We now discuss security notions for FE that match our syntactic extensions to the primitive. Given the limitations of indistinguishability-based notions already identified in [7,16] and further discussed in the full version of this paper, we adopt a semantic (or simulation-based) security notion akin to those in [7,16,13,3,6,2]. More precisely, we adopt of [13] extended to multiple encryption queries. Our choice is due to the conceptual simplicity of the model and to the general feasibility result we automatically obtain for unary functionalities. We restrict our attention to $\mathsf{TNA}$ queries to avoid impossibility results such as those presented in [7,6,2]. The definition formalizes the intuition that in a secure FE scheme, no information beyond that which is leaked through extracted tokens is available to an adversary. The fine details of the semantic security model are not important to our results, as long as the model is strong enough to imply an indistinguishability-based notion (Definition 8) that we introduce later in the paper.

**Definition 1 (Semantic security).** *Let games* $\mathsf{SS\text{-}R}_{\mathsf{FE},\mathcal{A},\mathcal{D}}$ *and* $\mathsf{SS\text{-}I}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}$ *be as in Figure 1. Let* $\mathsf{FE}$ *be an (*$n$*-ary) functional encryption scheme. We say* $\mathsf{FE}$ *is semantically secure if, for any adversary* $\mathcal{A}$*, there exists a simulator* $\mathcal{S}$ *such that for all distinguishers* $\mathcal{D}$ *the following definition of advantage is negligible.*

$$\mathbf{Adv}^{\mathsf{ss\text{-}cpa}}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) := \Pr\left[\mathsf{SS\text{-}R}_{\mathsf{FE},\mathcal{A},\mathcal{D}}(1^\lambda) \Rightarrow \mathsf{T}\right] - \Pr\left[\mathsf{SS\text{-}I}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(1^\lambda) \Rightarrow \mathsf{T}\right]$$

```
Game $\mathsf{SS\text{-}R}_{\mathsf{FE},\mathcal{A},\mathcal{D}}(1^\lambda)$:
  $\mathsf{KList} \leftarrow [\,\mathsf{k}_\epsilon\,]$; $\mathsf{m} \leftarrow \perp$
  $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow_\$ \mathsf{FE.Setup}(1^\lambda)$
  $\alpha \leftarrow_\$ \mathcal{A}^{\mathbf{Token},\mathbf{Enc}_0}(\mathsf{Mpk})$
  Return $\mathcal{D}(\mathsf{MList}, \alpha)$

Game $\mathsf{SS\text{-}I}_{\mathsf{FE},\mathcal{A},\mathcal{S},\mathcal{D}}(1^\lambda)$:
  $\mathsf{KList} \leftarrow [\,\mathsf{k}_\epsilon\,]$; $\mathsf{m} \leftarrow \perp$
  $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow_\$ \mathsf{FE.Setup}(1^\lambda)$
  $\alpha \leftarrow_\$ \mathcal{A}^{\mathbf{Token},\mathbf{Enc}_1}(\mathsf{Mpk})$
  Return $\mathcal{D}(\mathsf{MList}, \alpha)$

oracle $\mathbf{Token}(\mathsf{k})$:
  $\mathsf{TK} \leftarrow_\$ \mathsf{FE.TKGen}(\mathsf{k}, \mathsf{Msk})$
  $\mathsf{KList} \leftarrow \mathsf{k} : \mathsf{KList}$
  Return $\mathsf{TK}$

oracle $\mathbf{Enc}_0(\mathsf{m})$:
  $\mathsf{c} \leftarrow_\$ \mathsf{FE.Enc}(\mathsf{m}, \mathsf{Mpk})$
  $\mathsf{MList} \leftarrow \mathsf{m} : \mathsf{MList}$
  Return $\mathsf{c}$

oracle $\mathbf{Func}(\mathcal{I}, \mathsf{k}, \mathsf{m}_1, \ldots, \mathsf{m}_n)$:
  For $(i, j) \in \mathcal{I}$ do
    $\mathsf{m}_i \leftarrow \mathsf{MList}[j]$
  Return $\mathsf{Fn}(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}_1, \ldots, \mathsf{m}_n)$

oracle $\mathbf{Enc}_1(\mathsf{m})$:
  $\mathsf{c} \leftarrow_\$ \mathcal{S}^{\mathbf{Func}}(\mathsf{Mpk}, \mathsf{KList})$
  $\mathsf{MList} \leftarrow \mathsf{m} : \mathsf{MList}$
  Return $\mathsf{c}$
```

Fig. 1: Games defining the semantic security of an FE scheme. $\mathcal{A}$ is legitimate if it does not call **Token** after calling $\mathbf{Enc}_b$. In the single-message model $\mathcal{A}$ can call $\mathbf{Enc}_b$ exactly once. Simulator $\mathcal{S}$ is legitimate if it queries **Func** with $\mathsf{k} \in \mathsf{KList}$ only.

THE **Func** ORACLE. Note that providing a ciphertext and a set of tokens to the adversary in the real world allows the adversary to compute many images of the functionality simply by encrypting fresh ciphertexts and decrypting (evaluating) them along with the provided ciphertext. Our definition provides an oracle to the simulator in the ideal world that gives it essentially the same power. Here $\mathcal{I}$ denotes a set of index pairs $(i, j)$ which indicates hidden message $j$ should be used in position $i$ of the functionality. Note that in the single-message setting and for unary functionalities, this oracle is equivalent to directly providing a list of images to the simulator, as in the definition proposed in [13]

FEASIBILITY. The feasibility results presented in [13,16] directly carry over to our definition for unary functionalities in the single-message model, where the adversary may only request a single challenge ciphertext. This can be extended to the multi-message model (for the $\mathsf{TNA}$ case) through a composition theorem, but we do not discuss the details here due to space constraints. For the multi-ary case, however, the feasibility problems resurface even in this more restricted scenario. More precisely, a problem similar to that identified for noninteractive noncommitting encryption [15] may arise. However, we do not know of any impossibility result that excludes the construction of an FE scheme for the *specific* functionalities that we require for building FHE schemes and leave a detailed treatment to future research.

## 3 Randomized Functional Encryption

In this section we propose a new cryptographic primitive that generalizes FE to randomized functionalities. We start by formally introducing the syntax and security of randomized functional encryption schemes. In the full version we describe several examples of cryptographic primitives that can be seen as particular cases of RFE.

RANDOMIZED FUNCTIONALITY. A randomized functionality is a probabilistic algorithm $\mathsf{RFn}$ implementing an $n$-ary function $f$ that also takes a set of random coins from a randomness space $\mathsf{RFn.RndSp}$:

$$f : \mathsf{RFn.Prms} \times \mathsf{RFn.KeySp} \times \mathsf{RFn.MsgSp}^n \times \mathsf{RFn.RndSp} \longrightarrow \mathsf{RFn.Rng}\,.$$

We assume the random coins are uniformly distributed over $\mathsf{RFn.RndSp}$.

SYNTAX. A *randomized functional encryption (RFE) scheme* for the randomized functionality $\mathsf{RFn}$ (of arity $n$) is specified by four algorithms as follows.

1. $\mathsf{RFE.Setup}(1^\lambda)$: This is the setup algorithm. On input the security parameter, it outputs a master secret key $\mathsf{Msk}$ and a master public key $\mathsf{Mpk}$, a key space, and a message space. We require that the set of all possible master public keys output by the setup algorithm, the key space, and the message space are identical to $\mathsf{RFn.Prms}$, $\mathsf{RFn.KeySp}$, and $\mathsf{RFn.MsgSp}$ respectively.
2. $\mathsf{RFE.TKGen}(\mathsf{k}, \mathsf{Msk})$: This is the token-generation algorithm. On input a key $\mathsf{k} \in \mathsf{RFn.KeySp}$ and a master secret key $\mathsf{Msk}$, it outputs a token $\mathsf{TK}$.
3. $\mathsf{RFE.Enc}(\mathsf{m}, \mathsf{Mpk})$: This is the encryption algorithm. On input a message $\mathsf{m} \in \mathsf{RFn.MsgSp}$ and a master public key $\mathsf{Mpk}$, it outputs a ciphertext $\mathsf{c}$.
4. $\mathsf{RFE.Dec}(\mathsf{c}_1, \ldots, \mathsf{c}_n, \mathsf{TK}, \mathsf{Mpk})$: This is the (possibly probabilistic) decryption (or evaluation) algorithm. On input $n$ ciphertexts, a token $\mathsf{TK}$, and a master public key $\mathsf{Mpk}$, it outputs a message $\mathsf{m}$ or a special symbol $\perp$.

CORRECTNESS. Intuitively, the correctness requirement imposes that the distribution of a decrypted value (over the random coins of the encryption and decryption algorithms) is computationally indistinguishable from that obtained by sampling the randomized functionality directly on the encrypted message. We formalize this property next.

**Definition 2 (Correctness).** *Let game $\mathsf{COR}_{\mathsf{RFE},\mathcal{A}}$ be as in Figure 2. Let $\mathsf{RFE}$ be a randomized functional encryption scheme. We say $\mathsf{RFE}$ is correct if, for any adversary $\mathcal{A}$, the following definition advantage is negligible.*

$$\mathbf{Adv}^{\mathsf{cor}}_{\mathsf{RFE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{COR}_{\mathsf{RFE},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}\right] - 1$$

```
Game $\mathsf{COR}_{\mathsf{RFE},\mathcal{A}}(1^\lambda)$:
    $b \leftarrow_\$ \{0,1\}$
    $(\mathsf{Mpk},\mathsf{Msk}) \leftarrow_\$ \mathsf{RFE.Setup}(1^\lambda)$
    $b' \leftarrow_\$ \mathcal{A}^{\mathbf{O}}(\mathsf{Mpk},\mathsf{Msk})$
    Return $(b = b')$

oracle Token$(\mathsf{k})$:
    $\mathsf{TK} \leftarrow_\$ \mathsf{RFE.TKGen}(\mathsf{k},\mathsf{Msk})$
    $\mathsf{KList} \leftarrow (\mathsf{TK},\mathsf{k}) : \mathsf{KList}$
    Return $\mathsf{TK}$

oracle Func$(i_1,\ldots,i_n,j)$:
    For $\ell = 1$ to $n$ do $(\mathsf{m}_\ell,\mathsf{c}_\ell) \leftarrow \mathsf{MList}[i_\ell]$
    $(\mathsf{TK},\mathsf{k}) \leftarrow \mathsf{KList}[j]$
    $\mathsf{y}_0 \leftarrow_\$ \mathsf{RFE.Dec}(\mathsf{TK},\mathsf{c}_1,\ldots,\mathsf{c}_n)$
    $\mathsf{y}_1 \leftarrow_\$ \mathsf{RFn}(\mathsf{Mpk},\mathsf{k},\mathsf{m}_1,\ldots,\mathsf{m}_n)$
    Return $\mathsf{y}_b$

oracle Enc$(\mathsf{m})$:
    $\mathsf{c} \leftarrow_\$ \mathsf{RFE.Enc}(\mathsf{m},\mathsf{Mpk})$
    $\mathsf{MList} \leftarrow (\mathsf{m},\mathsf{c}) : \mathsf{MList}$
```

Fig. 2: Game defining the correctness of an RFE scheme. An adversary is legitimate if all **Func** queries are distinct.

We emphasize that the adversary in the correctness game has access to Msk (this is needed to model correctness for Msk-dependent messages), and that it may force some of the ciphertexts to be repeatedly used in the input to the decryption algorithm.

SECURITY. Our proposed notions of security for RFE are indistinguishability based. We will formalize the definitions and then discuss why the models do not necessarily suffer from the same limitations as IND-CPA security for FE.

**Definition 3 (Indistinguishability).** *Let game* $\mathsf{IND\text{-}CPA}_{\mathsf{RFE},\mathcal{A}}$ *be as shown in Figure 3. We say* RFE *is* IND-CPA *secure if, for any adversary* $\mathcal{A}$*, the following definition of advantage is negligible.*

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

```
Game $\mathsf{IND\text{-}CPA}_{\mathsf{RFE},\mathcal{A}}(1^\lambda)$:
    $b \leftarrow_\$ \{0,1\}$
    $(\mathsf{Msk},\mathsf{Mpk}) \leftarrow_\$ \mathsf{RFE.Setup}(1^\lambda)$
    $b' \leftarrow_\$ \mathcal{A}^{\mathbf{O}}(\mathsf{Mpk})$
    Return $(b = b')$

oracle LR$(\mathsf{m}_0,\mathsf{m}_1)$:
    $\mathsf{c} \leftarrow_\$ \mathsf{RFE.Enc}(\mathsf{m}_b,\mathsf{Msk})$
    Return $\mathsf{c}$

oracle Token$(\mathsf{k})$:
    $\mathsf{TK} \leftarrow_\$ \mathsf{RFE.TKGen}(\mathsf{k},\mathsf{Msk})$
    Return $\mathsf{TK}$
```

Fig. 3: Game defining the IND-CPA security of an RFE. In the TNA model $\mathcal{A}$ may not query **Token** after **LR**.

We observe that, unless the functionality has special characteristics, the images recovered by the adversary may allow it to trivially win the game; for instance, the above definition is unrealizable for nontrivial *deterministic* functionalities. To deal with this, we will consider restricted classes of *legitimate* adversaries that cannot "trivially" win the game. We start with a definition which characterizes when image values can be used to win the IND-CPA game.

**Definition 4 (Message-hiding RFE).** *Let game* $\mathsf{MH}_{\mathsf{RFE},\mathcal{A}}$ *be as in Figure 4. We say* RFE *is message hiding if, for any adversary* $\mathcal{A}$ *, the following definition of advantage is negligible.*

$$\mathbf{Adv}^{\mathsf{mh}}_{\mathsf{RFE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{MH}_{\mathsf{RFE},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}\right] - 1$$

```
Game $\mathsf{MH}_{\mathsf{RFE},\mathcal{A}}(1^\lambda)$:
  $b \leftarrow_\$ \{0,1\}$
  $(\mathsf{Msk}, \mathsf{Mpk}) \leftarrow_\$ \mathsf{RFE.Setup}(1^\lambda)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathbf{O}}(\mathsf{Mpk})$
  Return $(b = b')$

oracle $\mathbf{LR}(\mathsf{m}_0, \mathsf{m}_1)$:
  $\mathsf{MList} \leftarrow (\mathsf{m}_0, \mathsf{m}_1) : \mathsf{MList}$

oracle $\mathbf{Func}(\mathcal{I}, \mathsf{m}_1, \ldots, \mathsf{m}_n, \mathsf{k})$:
  For $(i, j) \in \mathcal{I}$ do
      $(\mathsf{m}^0, \mathsf{m}^1) \leftarrow \mathsf{MList}[j]$
      $\mathsf{m}_i \leftarrow \mathsf{m}^b$
  $\mathsf{y} \leftarrow_\$ \mathsf{RFn}(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}_1, \ldots, \mathsf{m}_n)$
  Return $\mathsf{y}$

oracle $\mathbf{Token}(\mathsf{k})$:
  $\mathsf{TK} \leftarrow_\$ \mathsf{RFE.TKGen}(\mathsf{k}, \mathsf{Msk})$
  $\mathsf{KList} \leftarrow \mathsf{k} : \mathsf{KList}$
  Return $\mathsf{TK}$
```

Fig. 4: Game defining the message-hiding property. An adversary $\mathcal{A}$ is legitimate if it queries **Func** with $\mathsf{k} \in \mathsf{KList}$ only. In the TNA model $\mathcal{A}$ may not query **Token** after querying **LR**.

The message-hiding property detects whether an adversary can distinguish messages queried to the **LR** oracle by looking at images of the functionality. Note that this may depend intrinsically on the way in which the domain parameters are sampled by the setup algorithm of the RFE scheme, and hence we have honest parameter generation. We can now introduce variants of the IND-CPA definition where we exclude trivial attacks by adversaries that win the game by exploiting information leaked via the images. To this end, we formalize the notion of an *associated adversary* $\mathcal{B}$ that mimics $\mathcal{A}$'s capabilities in the MH game.

**Definition 5 (Associated adversary).** *Let* RFE *be an RFE scheme, and let* $\mathcal{A}$ *be an adversary in the* $\mathsf{IND\text{-}CPA}_{\mathsf{RFE},\mathcal{A}}$ *game. Let also* $\mathcal{B}$ *be an adversary in the* $\mathsf{MH}_{\mathsf{RFE},\mathcal{B}}$ *game. Define the traces of* $\mathcal{A}$ *and* $\mathcal{B}$ *to be* (Mpk, MList, KList, TKList) *in their respective games, where* TKList *is the list of tokens returned by the* **Token** *oracle. We say* $\mathcal{B}$ *is an adversary associated to* $\mathcal{A}$ *if the traces of* $\mathcal{A}$ *and* $\mathcal{B}$ *are computationally indistinguishable. Weak associated adversary is defined analogously, where* TKList *is omitted from the traces.*

We now set legitimacy criteria for IND-CPA adversaries, excluding MH attacks.

**Definition 6 (Legitimacy).** *Let* RFE *be a randomized functional encryption scheme, and let* $\mathcal{A}$ *be an* $\mathsf{IND\text{-}CPA}_{\mathsf{RFE},\mathcal{A}}$ *adversary. We say* $\mathcal{A}$ *is legitimate if the advantage of any adversary* $\mathcal{B}$ *associated to* $\mathcal{A}$ *in the* $\mathsf{MH}_{\mathsf{RFE},\mathcal{B}}$ *game is negligible.*

Let us now introduce our weaker indistinguishability notions before discussing the legitimacy conditions.

**Definition 7 (Restricted indistinguishability).** *Let* $\mathsf{RFE}$ *be a randomized functional encryption scheme. The* $\mathsf{R\text{-}IND\text{-}CPA}$ *security of* $\mathsf{RFE}$ *requires the advantage of any legitimate adversary* $\mathcal{A}$ *in the* $\mathsf{IND\text{-}CPA}$ *game to be negligible.*

The following implications are easy to verify: $\mathsf{IND\text{-}CPA} \implies \mathsf{R\text{-}IND\text{-}CPA}$; $\mathsf{R\text{-}IND\text{-}CPA} + \mathsf{MH} \implies \mathsf{IND\text{-}CPA}$. Note also that, if scheme $\mathsf{RFE}$ is $\mathsf{IND\text{-}CPA}$ secure, then the supported functionality must be $\mathsf{MH}$ secure. (If the functionality is not message hiding, then the adversary could trivially break the security of the RFE scheme by simply distinguishing the images it recovers using legitimately retrieved tokens). As we shall see later in the paper, our constructions of a homomorphic scheme from an RFE impose that the adversary is *un*restricted in its challenge query. One way to achieve this is to concentrate on RFEs that are *provably* message hiding. The potential limitations of this indistinguishability-based definition are discussed in the full version of this paper.

## 4 Entropic security

We now turn our attention to a particular class of (standard) functional encryption schemes that we will use as a stepping-stone between the standard notions of functional encryption and randomized functional encryption. We call such functional encryption schemes *entropically secure*.

Intuitively, entropic security imposes that an adversary has a negligible advantage in distinguishing encryptions *provided that a part of the encrypted message is sampled uniformly at random*. To this end, we must restrict our attention to functionalities $\mathsf{Fn}$ for which the plaintext space $\mathsf{Fn.MsgSp}$ is partitioned as $\mathsf{Fn.MsgSp_m} \times \mathsf{Fn.MsgSp_r}$. For FE schemes supporting such functionalities, we can define a natural adaptation of the $\mathsf{IND\text{-}CPA}$ model as follows.

**Definition 8 (Entropic indistinguishability).** *Let game* $\mathsf{IND\text{-}ECPA}_{\mathsf{FE},\mathcal{A}}$ *be as in Figure 5. Let* $\mathsf{FE}$ *be a functional encryption scheme with a portioned message space. We say* $\mathsf{FE}$ *is* $\mathsf{IND\text{-}ECPA}$ *secure if for any adversary* $\mathcal{A}$*, the following definition of advantage is negligible.*

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

Unlike the standard definition of $\mathsf{IND\text{-}CPA}$, the above definition is meaningful for functionalities of arbitrary arity. Note also that unless the functionality has special characteristics the images recovered by the adversary may allow it to trivially win the game. We therefore adopt a similar strategy to that presented

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

oracle $\mathbf{LR}(\mathsf{m}_0, \mathsf{m}_1)$:
    $\mathsf{r} \leftarrow_\$ \mathsf{FE.MsgSp_r}$
    $\mathsf{c} \leftarrow_\$ \mathsf{FE.Enc}((\mathsf{m}_b, \mathsf{r}), \mathsf{Msk})$
    Return $\mathsf{c}$

oracle $\mathbf{Token}(\mathsf{k})$:
    $\mathsf{TK} \leftarrow_\$ \mathsf{FE.TKGen}(\mathsf{k}, \mathsf{Msk})$
    Return $\mathsf{TK}$
```

Fig. 5: Game defining the IND-ECPA security of an FE scheme. In the TNA model $\mathcal{A}$ may not query **Token** after querying **LR**.

for RFE security, define an *entropic message-hiding* property, and impose a legitimacy condition on the adversary to obtain a weaker flavor of the definition. We present these definitions in the full version of the paper. In entropic message hiding, the two differences to the model presented for RFEs are the following. Firstly, the **LR** oracle now samples part of the hidden message, as in the IND-ECPA definition. Secondly, the **Func** oracle allows the adversary to evaluate the functionality choosing all the r inputs except those that correspond to **LR** queries. This means that the EMH property (contrarily to message hiding for RFEs, where fresh images are sampled on each query to **Func**) entails a form of security under *randomness reuse* across the different functions k that are queried by the adversary. Note also that for multi-ary functionalities the adversary can *maliciously* choose part of the randomness components in the input to Fn. Later in the paper we will see how to build entropically message-hiding FEs. The definition of legitimacy is analogous to that presented for RFEs. We call the resulting definition *restricted* entropic indistinguishability. We also note that the legitimacy of an IND-ECPA adversary is a natural generalization of the restriction imposed in the IND-CPA model for FEs that the functions queried to the **Token** oracle must all collide on the challenge messages.

**Constructing RFEs from FEs** We now show how to construct RFEs for arbitrary functionalities starting from FE schemes. Technically, we will show that specific entropically secure FE schemes suffice, and then establish a connection to semantically secure FE.

The intuition behind our construction is the following. Suppose our goal is to construct an RFE scheme for a functionality RFn of arity $n$. We begin by defining a derandomized version of RFn, denoted Fn, of arity $n$, which we call the deterministic functionality *associated* to RFn. We then show that the simple generic construction in Figure 6 converts a functional encryption scheme FE into a correct and secure RFE scheme for RFn, provided that the underlying scheme FE is correct and entropically secure for Fn. Observe that the only modification that is made to the original functional encryption scheme is that the encryption algorithm samples extra randomness that is encrypted along with the message. This naturally relates to entropic security, where we considered functionalities where the message space was partitioned as $\mathsf{MsgSp} = \mathsf{MsgSp}_\mathsf{m} \times \mathsf{MsgSp}_\mathsf{r}$. We now formalize this intuition.

```
algo. RFE.Setup($1^\lambda$):
    Return FE.Setup($1^\lambda$)

algo. RFE.TKGen(k, Msk):
    Return FE.TKGen(k, Msk)

algo. RFE.Enc(m, Mpk):
    $\mathsf{r} \leftarrow_\$ \mathsf{Fn.MsgSp_r}$
    $\mathsf{c} \leftarrow_\$ \mathsf{FE.Enc((m, r), Mpk)}$
    Return c

algo. RFE.Dec($\mathsf{c}_1, \ldots, \mathsf{c}_n$, TK, Mpk):
    Return FE.Dec($\mathsf{c}_1, \ldots, \mathsf{c}_n$, TK, Mpk)
```

Fig. 6: Generic construction of an RFE scheme from an FE scheme.

CORRECTNESS. We begin by defining what it means to *correctly derandomize* a randomized functionality. We require that Fn and RFn are computationally indistinguishable to an adversary with oracle access that can choose all message inputs to Fn/RFn (except of course the components coming from $\mathsf{Fn.MsgSp_r}$).

**Definition 9 (Derandomized functionality).** *Let game* $\mathsf{DRND}_{\mathsf{RFn},\mathsf{Fn},\mathcal{A}}$ *be as shown in Figure 7. Let* RFn *be an $n$-ary randomized functionality. Let* Fn *be a deterministic functionality with the same arity, parameter space, and key space as those of* RFn. *Suppose further that the message space of* Fn *is partitioned as* $\mathsf{Fn.MsgSp} = \mathsf{RFn.MsgSp} \times \mathsf{Fn.MsgSp_r}$. *We say* Fn *correctly derandomizes* RFn *if, for any adversary $\mathcal{A}$, the following definition of advantage is negligible.*

$$\mathbf{Adv}^{\mathsf{drnd}}_{\mathsf{RFn},\mathsf{Fn},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{DRND}_{\mathsf{RFn},\mathsf{Fn},\mathcal{A}}(1^\lambda) \Rightarrow \mathsf{T}\right] - 1$$

```
Game $\mathsf{DRND}_{\mathsf{RFn},\mathsf{Fn},\mathcal{A}}(1^\lambda)$:
    $b \leftarrow_\$ \{0,1\}$
    $b' \leftarrow_\$ \mathcal{A}^{\mathbf{O}}(1^\lambda)$
    Return $(b = b')$

oracle Rand(m):
    $\mathsf{r} \leftarrow_\$ \mathsf{Fn.MsgSp_r}$
    $\mathsf{List} \leftarrow (\mathsf{m}, \mathsf{r}) : \mathsf{List}$

oracle Func(Mpk, $i_1, \ldots, i_n$, k):
    For $\ell = 1$ to $n$ do $(\mathsf{m}_\ell, \mathsf{r}_\ell) \leftarrow \mathsf{List}[i_\ell]$
    $\mathsf{y}_0 \leftarrow \mathsf{Fn}(\mathsf{Mpk}, \mathsf{k}, (\mathsf{m}_1, \mathsf{r}_1), \ldots, (\mathsf{m}_n, \mathsf{r}_n))$
    $\mathsf{y}_1 \leftarrow_\$ \mathsf{RFn}(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}_1, \ldots, \mathsf{m}_n)$
    Return $\mathsf{y}_b$
```

Fig. 7: Game defining correct derandomization of RFn by Fn. An adversary is legitimate if all **Func** queries are distinct.

Observe that the derandomized functionality must simulate independent samplings of RFn, whilst reusing the same input randomness. We now state the correctness result. The proof is a direct reduction and is given in the full version.

**Proposition 1.** *Let* RFn *and* Fn *be a randomized and a deterministic functionality, respectively. Suppose that* Fn *correctly derandomizes* RFn. *Then, any correct* FE *supporting* Fn *yields, by the construction in Figure 6, a correct* RFE *scheme supporting* RFn.

BUILDING DERANDOMIZED FUNCTIONALITIES. We now consider how unary and binary functionalities can be derandomized. (The techniques used to deal with the binary case naturally extend to arity greater than 2.) For each randomized functionality that we want to support in the generic RFE construction of Figure 6, we define an associated deterministic functionality as follows.

**Definition 10 (Associated deterministic functionality).** *Let* $\mathsf{RFn}$ *be a randomized functionality.*

**Unary** $\mathsf{RFn}$**:** *Let* $\mathsf{PRF} : \mathsf{PRF.KeySp} \times \mathsf{RFn.KeySp} \longrightarrow \mathsf{RFn.RndSp}$ *be a pseudorandom function. The associated unary deterministic functionality* $\mathsf{Fn}$ *is*

$$\mathsf{Fn}(\mathsf{Mpk}, \mathsf{k}, (\mathsf{m}, \mathsf{r})) := \mathsf{RFn}\left(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}; \mathsf{PRF}_{\mathsf{r}}(\mathsf{k})\right) .$$

*The key space and the parameter space of* $\mathsf{Fn}$ *match those of* $\mathsf{RFn}$ *and the message space of* $\mathsf{Fn}$ *is* $\mathsf{RFn.MsgSp} \times \mathsf{Fn.MsgSp}_{\mathsf{r}}$ *where* $\mathsf{Fn.MsgSp}_{\mathsf{r}} := \mathsf{PRF.KeySp}$.

**Binary** $\mathsf{RFn}$**:** *Let* $\mathsf{PRF} : \mathsf{PRF.KeySp} \times (\{0,1\}^{2\lambda} \times \mathsf{RFn.KeySp}) \longrightarrow \mathsf{RFn.RndSp}$ *be a pseudorandom function with* $\mathsf{PRF.KeySp}$ *forming an abelian group with operation* $\circ$*. The associated binary deterministic functionality* $\mathsf{Fn}$ *is defined as*

$$\mathsf{Fn}(\mathsf{Mpk}, \mathsf{k}, (\mathsf{m}_1, \mathsf{r}_1, \mathsf{s}_1), (\mathsf{m}_2, \mathsf{r}_2, \mathsf{s}_2)) := \mathsf{RFn}\left(\mathsf{Mpk}, \mathsf{k}, \mathsf{m}_1, \mathsf{m}_2; \mathsf{PRF}_{\mathsf{r}_1 \circ \mathsf{r}_2}(\mathsf{s}_1 || \mathsf{s}_2, \mathsf{k})\right) .$$

*The key space and the parameter space of* $\mathsf{Fn}$ *match those of* $\mathsf{RFn}$ *and the message space of* $\mathsf{Fn}$ *is* $\mathsf{RFn.MsgSp}_{\mathsf{m}} \times \mathsf{Fn.MsgSp}_{\mathsf{r}}$ *where* $\mathsf{Fn.MsgSp}_{\mathsf{r}} := \mathsf{PRF.KeySp} \times \{0,1\}^{\lambda}$.

We first show that an associated deterministic functionality, as defined as above, satisfies the correctness derandomization criterion. The proof of the following theorem is a direct reduction and can be found in the full version.

**Theorem 1 (Derandomization via PRFs).** *Let* $\mathsf{RFn}$ *be a unary or a binary randomized functionality. Let* $\mathsf{Fn}$ *be the associated deterministic functionality to* $\mathsf{RFn}$ *that uses the pseudorandom function* $\mathsf{PRF}$*. Suppose that* $\mathsf{PRF}$ *satisfies the standard notion of* $\mathsf{PRF}$ *security. Then* $\mathsf{Fn}$ *correctly derandomizes* $\mathsf{RFn}$.

SECURITY. We can now discuss the security of the construction. The next theorem shows that an FE scheme supporting the associated deterministic functionality to $\mathsf{RFn}$ suffices to obtain a secure RFE with an analogous security level.

**Theorem 2 (Security of the RFE construction).** *Let* $\mathsf{RFn}$ *be a randomized functionality and let* $\mathsf{Fn}$ *be its associated deterministic functionality. Let* $\mathsf{FE}$ *be a functional encryption scheme supporting* $\mathsf{Fn}$*. Then if* $\mathsf{FE}$ *is* $\mathsf{IND\text{-}ECPA}$ *secure, the RFE scheme* $\mathsf{RFE}$ *resulting from the generic construction in Figure 6 is*

$\mathsf{IND\text{-}CPA}$ *secure. A similar result holds for the restricted indistinguishability game if the PRF is* $\Phi$-$\mathsf{RKA}$ *secure, where*

$$\begin{aligned}\Phi := \{\phi_i^{\mathsf{L}} : (\mathsf{K}_1, \cdots, \mathsf{K}_n) \mapsto \mathsf{K}_i \circ \mathsf{L} \ \mid \ \mathsf{L} \in \mathsf{PRF.KeySp}\} \cup \\ \{\phi^{\star}_{i,j} : (\mathsf{K}_1, \cdots, \mathsf{K}_n) \mapsto \mathsf{K}_i \circ \mathsf{K}_j \mid i, j \in \mathbb{N} \wedge i \leq j\} \ .\end{aligned}$$

The proof is a direct reduction. However, for the restricted case, one needs to prove that the resulting $\mathsf{IND\text{-}ECPA}$ adversary is legitimate. The most challenging part of this argument consists of establishing that any successful $\mathsf{EMH}$ adversary against FE gives rise to a successful $\mathsf{MH}$ adversary against RFE. The intuition is as follows. The RKA security of the PRF ensures that its outputs look random even if one of the inputs to the functionality is maliciously chosen. This allows us to make a transition from the entropically message-hiding game to another game where the random coins of $\mathsf{RFn}$ are generated via a truly random function. The reduction to the legitimacy then consists of proving that any adversary succeeding in this game would either be breaking the $\mathsf{MH}$ property, or triggering the unlikely event of guessing input $\mathsf{s}$, when this is sampled uniformly at random. Details of the proof may be found in the full version.

**Constructing entropically secure FE** In this section we discuss how to construct entropically secure FEs from semantically secure functional encryption schemes. In our constructions we will be relying on FE schemes supporting functionalities that do *not* depend on the domain parameters, as this is the standard FE notion from [7,13] (modulo the arity extension). Therefore we first discuss how to construct a functionality that may depend on the parameters from one with a larger key space that no longer does. This then leads us to a natural transformation of FE schemes converting an FE scheme supporting the constructed parameter-independent functionality to one which supports the original parameter-dependent functionality. We show that this transformation preserves semantic security, and conclude the section by showing that any semantically secure FE scheme is also entropically secure.

REMOVING PARAMETER DEPENDENCY. Let $\mathsf{Fn}$ be a parameter-dependent functionality. We define a parameter-independent functionality as

$$\overline{\mathsf{Fn}}(\,\epsilon\,, (\mathsf{Mpk}, \mathsf{key}), \mathsf{m}_1, \cdots, \mathsf{m}_n) := \mathsf{Fn}(\mathsf{Mpk}, \mathsf{key}, \mathsf{m}_1, \cdots, \mathsf{m}_n) \,.$$

Therefore, the key space of $\overline{\mathsf{Fn}}$ is $\mathsf{Fn.Prms} \times \mathsf{Fn.KeySp}$. Let us look at a concrete example useful for our purposes. Consider an FE scheme in which we would like to support the following binary deterministic parameter-dependent functionality:

$$\begin{aligned}&\mathsf{Fn}_{\mathsf{NAND}}\,(\mathsf{pk}, \mathsf{k}_{\mathsf{NAND}}, (\mathsf{b}_1, (\mathsf{r}_1, \mathsf{s}_1)), (\mathsf{b}_2, (\mathsf{r}_2, \mathsf{s}_2))) := \\ &\qquad \mathsf{PKE.Enc}(\neg(\mathsf{b}_1 \wedge \mathsf{b}_2), \mathsf{pk}; \mathsf{PRF}_{\mathsf{r}_1 \circ \mathsf{r}_2}(\mathsf{s}_1 || \mathsf{s}_2, \mathsf{k}_{\mathsf{NAND}}))\,,\end{aligned}$$

where $\mathsf{PKE}$ is an encryption scheme and $\mathsf{k}_{\mathsf{NAND}}$ is the only key supported by the functionality. The converted functionality has key space identical to the public key space of the $\mathsf{PKE}$, and is given by

$$\overline{\mathsf{Fn}}_{\mathsf{NAND}}\,(\,\epsilon\,,(\mathsf{pk},\mathsf{k}_{\mathsf{NAND}}),(\mathsf{b}_1,(\mathsf{r}_1,\mathsf{s}_1)),(\mathsf{b}_2,(\mathsf{r}_2,\mathsf{s}_2))) := \mathsf{PKE.Enc}(\neg(\mathsf{b}_1 \wedge \mathsf{b}_2),\mathsf{pk};\mathsf{PRF}_{\mathsf{r}_1\mathsf{or}_2}(\mathsf{s}_1\|\mathsf{s}_2,\mathsf{k}_{\mathsf{NAND}}))\,.$$

We present our transformation in Figure 8, where we build scheme $\mathsf{FE}$ for $\mathsf{Fn}_{\mathsf{NAND}}$ from scheme $\overline{\mathsf{FE}}$ for $\overline{\mathsf{Fn}}_{\mathsf{NAND}}$. Note that although $\overline{\mathsf{FE}}_{\mathsf{NAND}}$ permits extracting tokens for encryption under *all* public keys in the underlying PKE scheme, $\mathsf{FE}_{\mathsf{NAND}}$ samples a *single* public key at set-up, which it publishes along with the master public key. The fact that the public key is sampled honestly means that not only we can rely on the security properties of the PKE, but also that we can include $(\mathsf{pk},\mathsf{sk})$ in the master key for $\mathsf{FE}_{\mathsf{NAND}}$. This means that the holder of the master public key is capable of recovering encrypted messages from $\mathsf{Fn}_{\mathsf{NAND}}$ images, a feature that we will use later on.

```
algo. $\mathsf{FE}_{\mathsf{NAND}}.\mathsf{Gen}(1^\lambda)$:
$(\mathsf{pk},\mathsf{sk}) \leftarrow_\$ \mathsf{PKE.Gen}(1^\lambda)$
$(\mathsf{Msk}',\mathsf{Mpk}') \leftarrow_\$ \overline{\mathsf{FE}}_{\mathsf{NAND}}.\mathsf{Setup}(1^\lambda)$
$\mathsf{Mpk} \leftarrow (\mathsf{Mpk}',\mathsf{pk})$
$\mathsf{Msk} \leftarrow (\mathsf{Msk}',\mathsf{sk},\mathsf{pk})$
Return $(\mathsf{Msk},\mathsf{Mpk})$

algo. $\mathsf{FE}_{\mathsf{NAND}}.\mathsf{TKGen}(\mathsf{Msk},\mathsf{k}_{\mathsf{NAND}})$:
$(\mathsf{Msk}',\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{Msk}$
$\mathsf{TK} \leftarrow_\$ \overline{\mathsf{FE}}_{\mathsf{NAND}}.\mathsf{TKGen}(\mathsf{k}_{\mathsf{pk}},\mathsf{Msk}')$
Return $\mathsf{TK}$

algo. $\mathsf{FE}_{\mathsf{NAND}}.\mathsf{Enc}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{Mpk})$:
$(\mathsf{Mpk}',\mathsf{pk}) \leftarrow \mathsf{Mpk}$
$\mathsf{c} \leftarrow_\$ \overline{\mathsf{FE}}_{\mathsf{NAND}}.\mathsf{Enc}(\mathsf{m}_1,\mathsf{m}_2,\mathsf{Mpk}')$
Return $\mathsf{c}$

algo. $\mathsf{FE}_{\mathsf{NAND}}.\mathsf{Dec}(\mathsf{c},\mathsf{TK},\mathsf{Mpk})$:
$(\mathsf{Mpk}',\mathsf{pk}) \leftarrow \mathsf{Mpk}$
Return $\overline{\mathsf{FE}}_{\mathsf{NAND}}.\mathsf{Dec}(\mathsf{c},\mathsf{TK},\mathsf{Mpk}')$
```

Fig. 8: Scheme $\mathsf{FE}_{\mathsf{NAND}}$ for $\mathsf{Fn}_{\mathsf{NAND}}$ based on scheme $\overline{\mathsf{FE}}_{\mathsf{NAND}}$ for $\overline{\mathsf{Fn}}_{\mathsf{NAND}}$.

The following result establishes that the above transformation yields a correct and secure FE scheme. The proof can be found in the full version. The intuition is that $\mathsf{FE}_{\mathsf{NAND}}$ uses only a subset of the functionality of $\overline{\mathsf{FE}}_{\mathsf{NAND}}$, and so the simulator implied by the hypothesis can be used to establish the semantic security of the construction.

**Proposition 2.** *If scheme $\overline{\mathsf{FE}}_{\mathsf{NAND}}$ is correct (for $\overline{\mathsf{Fn}}_{\mathsf{NAND}}$) and semantically secure, then scheme $\mathsf{FE}_{\mathsf{NAND}}$ is correct for $\mathsf{Fn}_{\mathsf{NAND}}$ and semantically secure.*

An important aspect of this result is that it goes through for the case in which the semantic security adversary places a single extraction query. This is important for feasibility, as the construction in [13] of FE schemes for general functionalities imposes the restriction that the semantic security adversary places a bounded number of such queries. Looking ahead, the FHE constructions we will present rely on FEs supporting functionalities with a single key.

FROM SEMANTIC TO ENTROPIC SECURITY. We now show that semantic security is strong enough to imply entropic security. Given that our definition of semantic security is restricted to the TNA scenario, we obtain entropic security in a similar setting. Luckily, entropic security under TNA attacks suffices for the results in the next section.

**Theorem 3.** *Let* FE *be a (deterministic) functional encryption scheme. Then if* FE *is semantically secure, it is also entropically secure in the* TNA *model.*

The intuition of the proof (given in the full version) is as follows. Any IND-ECPA attacker can be recast as a semantic security attacker that wins the real-world game with the same probability. The key observation is that the ideal world environment matches to entropic message hiding game, when this is played by an associated adversary to the original IND-ECPA attacker. However, the legitimacy condition on the attacker implies that this associated adversary cannot be successful, and that the real-world advantage (and hence the IND-ECPA advantage) must also be negligible.

Putting the above results together we obtain a path to constructing an RFE scheme for a randomized functionality RFn as follows. 1) Construct Fn, the associated deterministic functionality to RFn; 2) Construct scheme FE that is correct for Fn; 3) Prove that FE is semantically secure, and hence, by the above theorem, it is R-IND-ECPA secure; 4) Let RFE be the randomized scheme associated to FE. By Theorem 2 it is R-IND-CPA secure; 5) Finally, to achieve security against unrestricted adversaries (IND-CPA security), establish that RFE is message hiding. We will be using this strategy in the following section.

## 5 Relating Homomorphic and Functional Encryption Schemes

We saw earlier in the paper that homomorphic public-key encryption can be seen as a particular instance of a randomized functional encryption scheme. We now combine this observation with our results from the previous section to formalize a relation between FHE and FE.

FHE FROM ENTROPICALLY SECURE FE. We restrict our attention to homomorphic public-key encryption schemes that support encrypting bits (rather than strings) and which allow the homomorphic computation of an arbitrary number of NAND gates. We note that since NAND is complete, we can support the evaluation of arbitrary functions by first representing the function as a circuit of NAND gates and using bit-wise encryption on the inputs.

We start from an RFE scheme supporting the following binary randomized functionality $\mathsf{RFn}$, which we call *NAND re-encryption*, and is given by

$$\mathsf{RFn}(\mathsf{Mpk}, \mathsf{k}_{\mathsf{NAND}}, \mathsf{b}_1, \mathsf{b}_2) := \mathsf{RFE.Enc}(\neg(\mathsf{b}_1 \wedge \mathsf{b}_2), \mathsf{Mpk}) \,.$$

This functionality has message space $\{0, 1\}$ and it supports a single key $\mathsf{k}_{\mathsf{NAND}}$ (in addition to the empty key). We also assume, for the sake of correctness, that $\mathsf{RFE}$ supports a special decryption operation $\mathsf{Dec}(\mathsf{Msk}, \mathsf{c})$ akin to that of PKEs.

Our first construction of a fully homomorphic encryption scheme is as follows. The key generation algorithm generates $\mathsf{RFE}$ domain parameters and further extracts the token for $\mathsf{k}_{\mathsf{NAND}}$, which is added to the public key. Encryption is simply RFE encryption, and decryption is carried out using the special algorithm $\mathsf{Dec}(\mathsf{Msk}, \mathsf{c})$. Evaluation of a single NAND gate on two ciphertexts is carried out by running $\mathsf{RFE.Dec}(\mathsf{c}_1, \mathsf{c}_2, \mathsf{TK}_{\mathsf{k}_{\mathsf{NAND}}})$. Furthermore, the correctness of the underlying $\mathsf{RFE}$ scheme ensures that one can keep computing over the encrypted results, as the evaluated ciphertext will be with overwhelming probability in the co-domain of the RFE encryption circuit. In this way one can evaluate circuits of arbitrary size. Finally, it is easy to see that the $\mathsf{IND\text{-}CPA}$ security of the resulting FHE directly reduces to the $\mathsf{IND\text{-}CPA}$ security of the RFE scheme for single-message, $\mathsf{TNA}$ attacks, where by definition the adversary is *unrestricted*. We also observe that, by construction, this FHE is compact and function hiding. This is because the result of any computation is indistinguishable from a fresh encryption of the result of the computation, even to the holder of the decryption key. This construction and Theorems 1 and 2 immediately yield the following result.

**Theorem 4.** *Entropically secure FE with respect to unrestricted adversaries in the single message,* $\mathsf{TNA}$ *model and supporting the deterministic functionality associated with NAND re-encryption implies fully homomorphic encryption.*

We note that the underlying FE must be entropically secure against *unrestricted* adversaries. However, as we are dealing with bit-wise encryption, this is really the minimal assumption that one could have: the scheme should be secure when the adversary is allowed to choose challenge messages $\mathsf{m}_0 \neq \mathsf{m}_1$.

$\mathsf{FHE}$ FROM SEMANTICALLY SECURE $\mathsf{FE}$. The previous construction reveals an interesting relation between entropically secure FE, RFE, and FHE. However, it does not give a relation between semantically secure FE and FHE. It would be tempting to try to build an RFE scheme such as the one described above from semantically secure FE, using Theorem 3 to obtain security against restricted adversaries, and then proving that the resulting RFE is message hiding to obtain unrestricted security. However, this approach fails: the fact that the randomized functionality is defined using the same RFE scheme that supports it introduces a circular dependency that we cannot overcome. Intuitively, assuming that self-re-encryption is message hiding amounts to assuming that RFE construction is secure to begin with. (Note that in Theorem 4 this circular argument is broken by explicitly assuming security against unrestricted adversaries.)

We present an alternative, slightly more involved construction to overcome the above difficulty. We require two RFE schemes supporting the following binary functionalities, each having a single key (in addition to the empty key).

$\mathsf{RFE}_{\mathsf{NAND}}$ supports NAND re-encryption, with the caveat that re-encryption targets a standard public-key encryption scheme. More precisely,

$$\mathsf{RFn}_{\mathsf{NAND}}((\mathsf{Mpk},\mathsf{pk}),\mathsf{k}_{\mathsf{NAND}},\mathsf{b}_1,\mathsf{b}_2) := \mathsf{PKE.Enc}(\neg(\mathsf{b}_1 \wedge \mathsf{b}_2),\mathsf{pk})\,.$$

We also impose that the master secret key for $\mathsf{RFE}_{\mathsf{NAND}}$ includes the secret key $\mathsf{sk}$ corresponding to $\mathsf{pk}$.

$\mathsf{RFE}_{\mathsf{boot}}$ enables a functional analogue of the bootstrapping technique of Gentry [11]. It permits functionally decrypting a ciphertext under $\mathsf{PKE}$ and re-encrypting it under $\mathsf{RFE}_{\mathsf{NAND}}$:

$$\begin{aligned}\mathsf{RFn}_{\mathsf{boot}}\,((\mathsf{Mpk},\mathsf{Mpk}_{\mathsf{NAND}}),\mathsf{k}_{\mathsf{boot}},\mathsf{c},\mathsf{sk}) :=&\\ \mathsf{RFE}_{\mathsf{NAND}}.\mathsf{Enc}(\mathsf{PKE.Dec}(\mathsf{c},\mathsf{sk}),\mathsf{Mpk}_{\mathsf{NAND}})\,.&\end{aligned}$$

We also impose that the master secret key for $\mathsf{RFE}_{\mathsf{boot}}$ includes the master secret key $\mathsf{Msk}_{\mathsf{NAND}}$ corresponding to $\mathsf{Mpk}_{\mathsf{NAND}}$.

We observe that these RFE schemes can be constructed from parameter-independent FE schemes as discussed in Section 4. Indeed, $\mathsf{RFE}_{\mathsf{NAND}}$ can be constructed directly from the $\mathsf{FE}_{\mathsf{NAND}}$ construction in Figure 6. Also, from Theorems 2 and 3, these RFEs will be $\mathsf{R\text{-}IND\text{-}CPA}$ secure in the $\mathsf{TNA}$ model if the underlying FE schemes are semantically secure. To establish $\mathsf{IND\text{-}CPA}$ security, it suffices to prove the message-hiding property. We address this for $\mathsf{RFE}_{\mathsf{NAND}}$, as a similar argument follows for $\mathsf{RFE}_{\mathsf{boot}}$. The following result follows from the observation that the message-hiding game reduces to the $\mathsf{IND\text{-}CPA}$ security of the underlying PKE scheme.

**Theorem 5.** *If the underlying* $\mathsf{PKE}$ *is* $\mathsf{IND\text{-}CPA}$ *secure, then the RFE scheme supporting* $\mathsf{RFn}_{\mathsf{NAND}}$ *that results from plugging the FE scheme in Figure 8 in the generic construction in Figure 6 is message hiding.*

Using these RFE schemes, we construct an FHE scheme as follows. To encrypt or decrypt, one simply perform the equivalent operation using a $\mathsf{pk}$ for the underlying PKE. Evaluation of a single NAND gate proceeds as follows. The two ciphertexts are independently re-encrypted under $\mathsf{RFE}_{\mathsf{boot}}$, and then they are independently functionally decrypted and re-encrypted under $\mathsf{RFE}_{\mathsf{NAND}}$ using $\mathsf{TK}_{\mathsf{boot}}$. To enable this operation, $\mathsf{sk}$ is encrypted under $\mathsf{Mpk}_{\mathsf{boot}}$ and published in the public key, as is customary in bootstrapped constructions. Given any two PKE ciphertexts, one can therefore convert them into encryptions under $\mathsf{RFE}_{\mathsf{NAND}}$ from which one can evaluate a NAND gate and re-encrypt again under the PKE. We obtain a ciphertext that is indistinguishable from a fresh encryption, which means that the construction is compact and function hiding.

The construction is correct if the underlying RFE schemes are correct. For security, note that one can easily reduce the $\mathsf{IND\text{-}CPA}$ security of the resulting FHE scheme to that of $\mathsf{RFE}_{\mathsf{boot}}$ *if the secret key encrypted under* $\mathsf{c}_{\mathsf{boot}}$ *does not correspond to the public key* $\mathsf{pk}$ *used inside* $\mathsf{RFE}_{\mathsf{NAND}}$. This is typical in bootstrapping techniques [11], and the additional assumption that the construction securely encrypts key-dependent messages (i.e., that it is KDM secure) is necessary to use the scheme with the same $\mathsf{pk}$. On the other hand, without relying on this assumption, this result implies a leveled homomorphic encryption scheme [11], allowing a bounded number of cascaded NAND computations through multiple independent instances of $\mathsf{RFE}_{\mathsf{boot}}$. The last theorem in the paper follows from this discussion.

**Theorem 6.** *Semantically secure FE schemes supporting the deterministic functionalities associated to* $\mathsf{RFn}_{\mathsf{NAND}}$ *and* $\mathsf{RFn}_{\mathsf{boot}}$ *imply fully homomorphic encryption, under a key-dependent message security assumption.*

## 6 Relation to Obfuscated Re-Encryption

FUNCTIONAL RE-ENCRYPTION IN THE CIRCUIT MODEL. For concreteness, we consider the functional re-encryption notions of Section 5 in the randomized circuit model. Specifically, we consider a family of re-encryption circuits $\mathcal{R}^f = \{R^f_{\mathsf{sk},\mathsf{pk}'}\}$, called the *f-re-encryption functionality* from $\mathsf{PKE}$ to $\mathsf{PKE}'$, which decrypts each input (which is a ciphertext encrypted with $\mathsf{PKE}$) using key $\mathsf{sk}$, applies the function $f$ to the resulting values, and outputs the encryption of the result under key $\mathsf{pk}'$ using $\mathsf{PKE}'$. (In the full version, we consider the general discussion of functionalities that do not necessarily re-encrypt with fresh randomness, but here we restrict ourselves to such "canonical" functionalities.) Intuitively, a "secure" obfuscation of this circuit should serve as an evaluation key for a functional re-encryption scheme for function $f$.

OBFUSCATION FOR $f$-RE-ENCRYPTION. We define new notions of secure obfuscation as specifically applied to a $f$-re-encryption functionality $\mathcal{R}^f$. Following earlier work on obfuscation [20,10,1,14,8], we want the obfuscated circuit to perform the same computation as the original circuit without revealing any information beyond its input-output behavior.

Our notion differs from the one proposed by Chandran et al [9], while still following the same average-case viewpoint. We attempt to capture at the same time the fact that the obfuscated re-encryption functionality does not reveal *any* information beyond black-box access to the functionality *and* the fact that black-box access to the functionality does not reveal any information about the messages being encrypted. Still, our notion is connected to (and in many cases implied by) the notion defined in these earlier work, as we explain below.

**Definition 11 (Re-encryption Obfuscation).** *Let* $\mathsf{Obf}$ *be a PPT algorithm whose input and output are both circuits. We say* $\mathsf{Obf}$ securely obfuscates the $f$-re-encryption functionality $\mathcal{R}^f$ *from* $\mathsf{PKE}$ *to* $\mathsf{PKE}'$ *if the following properties hold:*

***Correctness****: For any circuit* $C = R^f_{\mathsf{sk},\mathsf{pk}'} \in \mathcal{R}^f$ *and for all inputs* $x$*, the statistical distance* $\Delta(\mathsf{Obf}(C)(x), C(x))$ *is negligible .*

***Simulatability****: There exists a PPT simulator* $S$ *such that for all PPT distinguishers* $\mathcal{D}$ *and security parameter* $\lambda$*,*

$$|\Pr[\mathcal{D}(\mathsf{pk},\mathsf{pk}',\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})) = 1] - \Pr[\mathcal{D}(\mathsf{pk}, S(\mathsf{pk})) = 1]| < negl(\lambda)$$

*where* $(\mathsf{sk},\mathsf{pk}) \leftarrow_{\$} \mathsf{Gen}(1^\lambda), (\mathsf{pk}',\mathsf{sk}') \leftarrow_{\$} \mathsf{Gen}'(1^\lambda)$*, and the probabilities are taken over the coins of* $\mathsf{Gen}$ *and* $S$*.*

We stress that the definition provides a very strong guarantee, in that it says that an attacker, given $\mathsf{pk}, \mathsf{pk}'$ and the obfuscation $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})$ does not learn *anything* beyond the public key $\mathsf{pk}$ of the source scheme. In particular, the simulator simulates the public key $\mathsf{pk}'$. Note that the obfuscation may be a randomized circuit itself, and that the correctness requirements assumes *honest* evaluation of the circuit, i.e., using honestly generated random coins. In the full version, we also consider a stronger simulatability requirement, where $\mathsf{pk}'$ is generated honestly and the simulator does not learn $\mathsf{sk}'$.

RELATION TO EARLIER DEFINITIONS OF OBFUSCATION. As mentioned above, previous works on re-encryption [14,9] consider a different notion of average-case obfuscation which appears at first incomparable to ours, in which the simulator must simulate $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})$, given *black-box* access to $R^f_{\mathsf{sk},\mathsf{pk}'}$ and knowing the public keys $\mathsf{pk}, \mathsf{pk}'$. Formally, when translated to our setting, the requirement of these earlier works is as follows:

**Virtual Black-boxness**: There exists a PPT simulator $S$ such that for all PPT distinguishers $\mathcal{D}$ and security parameter $\lambda$,

$$|\Pr[\mathcal{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}',\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'}))=1]- \\ \Pr[\mathcal{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}',S^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}')=1]| < negl(\lambda)$$

where $(\mathsf{sk},\mathsf{pk}) \leftarrow_{\$} \mathsf{Gen}(1^\lambda), (\mathsf{pk}',\mathsf{sk}') \leftarrow_{\$} \mathsf{Gen}'(1^\lambda)$, and the probabilities are taken over the coins of $\mathsf{Gen}$ and $S$.

This definition of strong virtual black-boxness implies our obfuscation notion above for natural re-encryption functionalities, hence making it a somewhat stronger notion. More concretely, we say that the $f$-re-encryption functionality $\mathcal{R}^f = \{R^f_{\mathsf{sk},\mathsf{pk}'}\}$ is *simulatable* if there exists a simulator $S'$ such that for all PPT distinguishers $\mathcal{D}$, we have

$$|\Pr[(\mathcal{D}^{R^f_{\mathsf{sk},\mathsf{pk}'}}(\mathsf{pk},\mathsf{pk}')=1]-\Pr[\mathcal{D}^{S'(\mathsf{pk},\mathsf{pk}')}(\mathsf{pk},\mathsf{pk}')=1]| < negl(\lambda)\ .$$

For example, the canonical re-encryption functionality is simulatable by semantic security, provided we can efficiently test if a ciphertext input to the functionality is decryptable given $\mathsf{pk}$ only. In the full version, we prove the following:

**Lemma 1.** *Assume that the obfuscator satisfies the virtual black-boxness property* and *the $f$-re-encryption functionality $\mathcal{R}^f$ is private. Then, the obfuscator satisfies the simulatability property.*

RELATION TO FUNCTIONAL ENCRYPTION. The main result of this section connects the notions of obfuscated re-encryption with functional encryption.

**Lemma 2.** *Any securely obfuscatable functional re-encryption scheme for function $f$ where the underlying public-key scheme(s) are semantically secure implies an* IND-CPA*-secure randomized functional encryption scheme with a single non-empty key for $f$-re-encryption.*

*Proof (Sketch).* We consider RFE defined as follows:

1. $\mathsf{RFE.Setup}(1^\lambda)$: Run $\mathsf{PKE.Gen}(1^\lambda)$ and $\mathsf{PKE}'.\mathsf{Gen}(1^\lambda)$ to obtain $(\mathsf{pk},\mathsf{sk})$ and $(\mathsf{pk}',\mathsf{sk}')$. Let $\mathsf{Mpk} = (\mathsf{pk},\mathsf{pk}')$ and $\mathsf{Msk} = \mathsf{sk}$. Let the message space of RFE be the message space of PKE, and the key space contain $\mathsf{k}_f$ (and $\mathsf{k}_\epsilon$).
2. $\mathsf{RFE.TKGen}(\mathsf{k}_f,\mathsf{Msk})$: Return $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})$.
3. $\mathsf{RFE.Enc}(\mathsf{m},\mathsf{Mpk})$: Return $\mathsf{PKE.Enc}(\mathsf{m},\mathsf{pk})$.
4. $\mathsf{RFE.Dec}(\mathsf{c}_1,\ldots,\mathsf{c}_n,\mathsf{TK},\mathsf{Mpk})$: Return $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}'})(\mathsf{c}_1,\ldots,\mathsf{c}_n)$; that is, the output of the obfuscated circuit applied to the ciphertexts.

Details about the security proof are deferred to the full version. □

Combined with the results of Section 5, we therefore conclude that secure obfuscators for circuits computing $\mathsf{RFn}_{\mathsf{NAND}}$ and $\mathsf{RFn}_{\mathsf{boot}}$ imply a fully-homomorphic encryption scheme. Such obfuscations can be constructed based on the LWE assumption, starting from the encryption scheme of Regev [17]. We defer the details of this construction to the full version.

### Acknowledgments

Tessaro and Wilson wish to thank Shafi Goldwasser for insightful feedback. Their work was partially supported by NSF Contract CCF-1018064 and is based on research sponsored by DARPA under agreement numbers FA8750-11-C-0096 and FA8750-11-2-0225. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

## References

1. Ben Adida and Douglas Wikström. How to Shuffle in Public. In Salil P. Vadhan, editor, *TCC 2007*, vol. 4392 of *LNCS*, pp. 555–574. Springer, February 2007.
2. Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: new perspectives and lower bounds. Cryptology ePrint Archive, Report 2012/468, 2012.
3. Manuel Barbosa and Pooya Farshim. On the semantic security of functional encryption schemes. In Kaoru Kurosawa, editor, *PKC 2013*, vol. 7778 of *LNCS*, pp. 143–161. Springer, 2013.
4. Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2010*, vol. 6223 of *LNCS*, pp. 666–684. Springer, 2010.
5. Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *EUROCRYPT 2003*, vol. 2656 of *LNCS*, pp. 491–506. Springer, 2004
6. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: possibility results, impossibility results and the quest for a general definition. Cryptology ePrint Archive, Report 2012/515, 2012.
7. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: definitions and challenges. In Yuval Ishai, editor, *TCC 2011*, vol. 6597 of *LNCS*, pp. 253–273. Springer, 2011.
8. Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of Hyperplane Membership. In Daniele Micciancio, editor, *TCC 2010*, vol. 5978 of *LNCS*, pp. 72-89. Springer, February 2010.
9. Nishanth Chandran, Melissa Chase, and Vinod Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In Ronald Cramer, editor, *TCC 2012*, vol. 7194 of *LNCS*, pp. 404–421. Springer, 2012.

10. Yevgeniy Dodis and Adam Smith. Correcting Errors Without Leaking Partial Information. In Harold N. Gabow and Ronald Fagin, editors, *STOC 2005*, pp. 654–663. ACM Press, May 2005.
11. Craig Gentry. *A fully homomorphic encryption scheme*. Ph.D. thesis, Stanford University, 2009.
12. Shafi Goldwasser, Yael Tauman Kalai, Raluca A. Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Succinct functional encryption and applications: reusable garbled circuits and beyond. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC '13*, pp. 555–564. ACM, 2013.
13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, vol. 7417 of *LNCS*, pp. 162–179. Springer, 2012.
14. Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely Obfuscating Re-encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pp. 233–252. Springer, February 2007.
15. Jesper Nielsen. Separating random oracle proofs from complexity theoretic proofs: the non-committing encryption case. In Moti Yung, editor, *CRYPTO 2002*, vol. 2442 of *LNCS*, pp. 111–126. Springer, 2002.
16. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.
17. Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In Harold N. Gabow and Ronald Fagin, editors, *STOC 2005*, pp. 84–93. ACM Press, May 2005.
18. Alexander Russell and Hong Wang. How to Fool an Unbounded Adversary with a Short Key. In Lars R. Knudsen, editors *EUROCRYPT 2002*, vol. 2332 of *LNCS*, pp. 133–148. Springer, 2002.
19. Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, vol. 7417 of *LNCS*, pp. 218–235. Springer, 2012.
20. Hoeteck Wee. On Obfuscating Point Functions. In Harold N. Gabow and Ronald Fagin, editors, *STOC 2005*, pp. 523–532. ACM Press, May 2005.