Obfuscating Many-to-one Functional Re-encryption, and its Connection to Fully-Homomorphic Encryption

Stefano Tessaro
MIT CSAIL

David A. Wilson
MIT CSAIL

{tessaro,dwilson}@mit.edu

Abstract

Following up on its first construction by Gentry (STOC 2009), fully-homomorphic encryption (FHE) has generated a multitude of different works, seeking both for new theoretical approaches as well as for more efficient instantiations. All existing FHE schemes, however, are tightly connected to specific assumptions, and no generic constructions are known.

In this paper, we investigate *generic constructions* of FHE. Specifically, we introduce a new primitive, called Many-to-One Functional Re-encryption, which allows, given multiple messages encrypted under a public key for one encryption scheme, to produce an encryption of a function of these messages under another key— possibly for a different encryption scheme. We introduce a new notion of obfuscation for many-to-one functional re-encryption, and show that such obfuscation yields a generic transformation from a semantically-secure encryption scheme to leveled FHE. We further demonstrate that existing FHE schemes (both those that employ bootstrapping and relinearization) can be viewed as instantiations of this paradigm.

1 Introduction

Fully-Homomorphic Encryption. The discovery of fully-homomorphic encryption schemes (FHE) has been a key development in modern cryptography. FHE schemes allow arbitrary computation on encrypted data without decrypting. The notion was first proposed by Rivest, Adleman, and Dertouzos [RAD78], but it took more than three decades for the first schemes to be developed. Several FHE schemes have now been developed, first under somewhat nonstandard lattice assumptions [Gen09, SV10], then under hardness assumptions for approximate GCD [vDGHV10, CMNT11, CNT12], and finally under various forms of the Learning With Errors assumption [BV11b, BV11a, BGV11, Bra12, GHS12b, GHS12a, GSW13] or other lattice-based assumptions [GH11].

At the same time, no general construction is known from smaller primitives, even for the case of *leveled FHE schemes*. A d-leveled FHE scheme allows computation of depth-d circuits on encrypted data, allowing its public key size to be a polynomial function in d. In this paper, we address the question of finding a primitive which allows a generic construction of FHE on top of a suitable encryption scheme, and revisit existing works in terms of instantiations of this blueprint.

Obfuscating re-encryption. Our approach relies on the notion of *obfuscated re-encryption*, which has been developed in parallel to FHE. While obfuscation of general functions is impossible [BGI+01], there have been several positive results detailing function families that can be obfuscated (e.g. [Wee05, DS05, CRV10], among many others). In particular, there has been a line of research on obfuscation that is secure *on average* (that is, for a random function from a family), rather than for any function in the family ([GK05, AW07], and others); this definition is particularly relevant to cryptographic applications that use randomized functions. Hohenberger et al [HRSV07] show a method to obfuscate a re-encryption functionality–that is, a functionality which allows for decryption under one key and encryption under a second–such that the re-encryption procedure can be delegated to a third party who does not learn anything about the re-encrypted messages. Chandran et al [CCV12] extended this work even further, and consider functional re-encryption, in which the second encryption key is a function of the underlying message, in the context of obfuscation of the function (and hiding the message). However, such functionalities have generally only been defined for single-input functions.

Many-to-one functional re-encryption. Our first contribution is to introduce and define the notion of *many-to-one functional re-encryption* and its obfuscation. More specifically, for a function f, this functionality allows an evaluator to take multiple ciphertexts $c_1, \ldots, c_q$ encrypting messages $m_1, \ldots, m_q$ under the same key pk for some public-key cryptosystem PKE, and computes an encryption of $f(m_1, \ldots, m_q)$ under a different key for some possibly different cryptosystem PKE'.

Clearly, this functionality is by itself uninteresting, as it can be trivially realized by decrypting the input messages, computing the function, and encrypting the result. However, this functionality becomes interesting if it can be obfuscated and hence delegated to a user without revealing the corresponding secret key. For this reason, we also define a notion of obfuscation for this functionality, which is substantially different than the one proposed by previous works on re-encryption, despite its similar "average-case" perspective: At a high level, our first definition states that for a random circuit computing the re-encryption and for an observer who knows the public key of the source scheme, the obfuscation of that circuit *and* the public key of the target scheme are indistinguishable from the output of a simulator that only knows the public-key of the source scheme. We also consider a stronger notion, where the simulator does not simulate the public key of the target scheme, but obtains it externally. We show that the latter definition is in fact *implied* by

the definition from [HRSV07].

FHE FROM MANY-TO-ONE FUNCTIONAL RE-ENCRYPTION. As one application of many-to-one functional encryption, our second contribution is to show a generic construction of leveled FHE given a semantically-secure encryption scheme such that the corresponding multi-input functional re-encryption functionalities for a complete set of operations (e.g., for the NAND operation) can be obfuscated with respect to the new notions introduced in this paper.

As an application, we show that Regev-style encryption [Reg05] admits such obfuscated re-encryption for multiplication, which, combined with our main result and the existing additive homomorphism of the encryption yields a level FHE scheme. This scheme corresponds to the one recently proposed by Brakerski [Bra12], for which we provide a more modular abstraction. We also reinterpret the technique of "bootstrapping" ([Gen09] and followup work) as specific implementations of our generic construction.

2 Preliminaries

2.1 Public-Key Encryption and Semantic Security

We start by introducing our notation to describe public-key encryption schemes. Specifically, a *public-key* encryption scheme is a triple of algorithms $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where:

- the randomized algorithm Gen is the key generation algorithm, which takes as input the security parameter 1^k, and outputs a public-key / secret-key pair $(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^k)$.
- Enc is the randomized encryption algorithm, and Dec is the deterministic decryption algorithm.

We assume that PKE is *correct* if for all valid public-key / secret-key pairs $(\mathsf{pk}, \mathsf{sk})$, and all messages m, the probability $\mathsf{Pr}\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \neq m\right]$ is negligible, where the probability is taken over the random coins of the encryption algorithm Enc. Moreover, we say that PKE is semantically secure if for all PPT distinguishers $\mathcal{D}$ and all messages m, we have

$$\mathsf{Pr}\left[(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m))) = 1\right] - \mathsf{Pr}\left[(\mathsf{pk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0))) = 1\right] \leq negl(n) \ .$$

2.2 Fully-Homomorphic Encryption

A fully homomorphic encryption (FHE) scheme is an encryption scheme which allows for arbitrary computation on encrypted data. Namely, it consists of a tuple $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ such that Gen outputs a triple of keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk})$, where evk is the additional *evaluation key*. The correctness requirements for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ are as in traditional public-key encryption. Moreover, Eval is the evaluation algorithm and is such that for every circuit f with q inputs, and messages $m_1, \ldots, m_q$, we have

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{evk}, f, \mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_q))) = f(m_1, \ldots, m_q) \ ,$$

where $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}$. Informally, we say that FHE is *leveled* (with d levels), if it only evaluates circuits of depth d (in some well defined circuit model), and the parameters are allowed to depend

on d. Finally, we say that FHE is semantically secure, if for all PPT distinguishers $\mathcal{D}$ and all messages m, we have

$$\mathsf{Pr}\left[(\mathsf{pk},\mathsf{sk},\mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{pk},\mathsf{evk},\mathsf{Enc}(\mathsf{pk},m))) = 1\right] - \mathsf{Pr}\left[(\mathsf{pk},\mathsf{sk},\mathsf{evk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{pk},\mathsf{evk},\mathsf{Enc}(\mathsf{pk},0))) = 1\right] \leq negl(n) \ .$$

FHE constructions in the literature include [Gen09, SV10, vDGHV10, CMNT11, BV11b, BV11a, GH11, BGV11, Bra12, GHS12b, CNT12, GHS12a, GSW13].

3 Many-to-one Functional Re-encryption and its Obfuscation

In this section, we introduce the notion of many-to-one functional re-encryption, as well as a new notion of obfuscation for this functionality which, while tailored at our applications, exhibits natural connections to previous notions.

3.1 Many-to-one Functional Re-encryption

We start by defining circuits providing many-to-one functional re-encryption. In the most general case, we are given two public-key encryption schemes PKE and PKE' (where potentially, but not necessarily, $\mathsf{PKE} = \mathsf{PKE}'$). We are interested in families of circuits $R^f_{\mathsf{sk},\mathsf{pk}'}$ indexed by valid *secret* keys sk for PKE and valid *public* keys pk' for PKE' which, given encryptions of messages $m_1,\ldots,m_q$ under PKE, produce an encryption of $f(m_1,\ldots,m_q)$ for PKE'. Of course, a canonical implementation of such circuit simply decrypts $c_1,\ldots,c_q$, and then re-encrypts $f(m_1,\ldots,m_q)$ with *fresh randomness*. However, we will not make any further assumptions on these circuits, i.e., they may be randomized or not, and we require them to work in a more general sense, where *any* q ciphertexts $c_1,\ldots,c_q$ decrypting to $m_1,\ldots,m_q$ under sk will result in a ciphertext decrypting to $f(m_1,\ldots,m_q)$.

Definition. *Let* $\mathsf{PKE} = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ *and* $\mathsf{PKE}' = (\mathsf{Gen}',\mathsf{Enc}',\mathsf{Dec}')$ *be public-key encryption schemes. Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be the message spaces of* PKE *and* PKE'*, respectively, and let* $f : \mathcal{M}^q \to \mathcal{M}'$ *be a function. A* f-re-encryption functionality from PKE to PKE' *is a family of (possibly randomized) circuits* $\mathcal{R}^f = \left\{R^f_{\mathsf{sk},\mathsf{pk}'}\right\}_{(\mathsf{sk},\mathsf{pk}')}$ *indexed by secret keys* sk *of* PKE *and public keys* pk' *of* PKE' *such that for all valid ciphertexts* $c_1,\ldots,c_q$ *for* PKE*,*

$$\mathsf{Dec}'(\mathsf{sk}', R^f_{\mathsf{sk},\mathsf{pk}'}(c_1,\ldots,c_q)) = f(m_1,\ldots,m_q) \ ,$$

with overwhelming probability over the random choices of $(\mathsf{pk},\mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}$*,* $(\mathsf{pk}',\mathsf{sk}') \xleftarrow{\$} \mathsf{Gen}'$*, and* R^f*, where* $m_i = \mathsf{Dec}(\mathsf{sk},c_i)$ *for* $i = 1,\ldots,q$*.*

Without loss of generality, it will be convenient to assume that the description of the circuit $R^f_{\mathsf{sk},\mathsf{pk}'}$ allows one to recover the value of sk and pk' efficiently.

Note that in the case where $q = 1$ and f is the identity, this notion corresponds to the traditional setting of re-encryption introduced by Hohenberger et al [HRSV07]. In contrast, the more general setting of functional re-encryption introduced by Chandran et al [CCV12] is different, in that it considers multiple recipients with different keypairs, and a function applied to an attribute associated with the ciphertext determines the *recipient* of the encryption. In their setting, however, no transformation is applied to the plaintext itself.

3.2 Obfuscation for Many-to-one Functional Re-encryption

We now define our new notion of secure obfuscation as specifically applied to the many-to-one re-encryption regime, i.e., to a f-re-encryption functionality $\mathcal{R}^f$ from a source scheme PKE to a target scheme PKE'. Following earlier work on obfuscation [Wee05, DS05, AW07, HRSV07, CRV10], we want the obfuscated circuit to perform the same computation as the original circuit. However, at the same time, we want to argue that an adversary does not learn any useful information from the obfuscated circuit beyond what it would learn by evaluating its functionality in purely black-box manner. This latter requirement is defined using a simulation-based approach, in contrast to indistinguishability-based obfuscation as in e.g. [AW07].

We note that for the case of one-argument functions, our notion will differ from the one proposed by Chase et al [CCV12], while still following the same average-case viewpoint. Intuitively, our notion attempts 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.

For now, more concretely, let Obf be a PPT algorithm whose input and output are both circuits. Obf is a secure obfuscator for re-encryption circuit family $\mathcal{R}^f$ if the following definition is satisfied.

Definition (Re-encryption Obfuscation). *We say that* Obf securely obfuscates the f-re-encryption functionality $\mathcal{R}^f$ *from* PKE *to* PKE' *if the following two properties hold:*

- ***Correctness****: For any* $C = R^f_{\mathsf{sk},\mathsf{pk}'} \in \mathcal{R}^f$*, the statistical distance* $\Delta(\mathsf{Obf}(C)(x), C(x))$ *is negligible for all inputs* x*.*
- ***Simulatability****: There exists a PPT simulator* S *such that for all PPT distinguishers* $\mathcal{D}$ *and security parameter* n*,*

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

  where the probabilities are taken over the coins of Gen *and* S*.*

This notion is somewhat different than those found in the existing literature on obfuscation; let us discuss this notion a little bit further. Generally, one defines obfuscators as being secure whenever the resulting obfuscation does not help more in computing the function implemented by the underlying circuit than black-box access to the function itself. We note 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 pk of the source scheme. 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.

We stress that the simulator is required to simulate the public-key pk' *together* with the obfuscation $\mathsf{Obf}(R^f_{\mathsf{sk},\mathsf{pk}})$. We also discuss a stronger notion of obfuscation where the simulator is restricted to use an externally generate public key pk' for the target scheme.

Definition (Strong Re-encryption Obfuscation). *We say that* Obf strongly securely obfuscates the f-re-encryption functionality $\mathcal{R}^f$ *from* PKE *to* PKE' *if correctness as above holds, and additionally, the following stronger simulatability requirement holds:*

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

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

where the probabilities are taken over the coins of Gen *and* S*.*

Relation to earlier definitions. As mentioned above, previous works on re-encryption [HRSV07, CCV12] considered 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 pk,pk'. Formally, when translated to our setting of multi-input functional re-encryption, 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 n,

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

  where the probabilities are taken over the coins of Gen and S.

We will now prove that strong virtual black-boxness implies our strong 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

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

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 pk only. Then, we can show the following:

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

Proof. As our new simulator $\hat{S}$ for the strong simulatability property, we use the simulator S for virtual black-boxness, taking pk and pk' as inputs, and use S' guaranteed to exist by simulatability of the functionality $\mathcal{R}^f$ to answer S's queries, i.e., for short, $\hat{S}(\cdot,\cdot) = S^{S'}(\cdot,\cdot)$. Then, if there exists an attacker $\mathcal{D}$ violating strong obfuscability, distinguishing with non-negligible advantage ε, then $\mathcal{D}$ also violates the virtual black-boxness property (without making oracle queries) with distinguishing advantage $\varepsilon - negl(n)$. This is because by the simulatability of $\mathcal{R}^f$, the probabilities that $\mathcal{D}$ outputs one when interacting with either of $(\mathsf{pk},\mathsf{pk}',\hat{S}(\mathsf{pk},\mathsf{pk}')) = (\mathsf{pk},\mathsf{pk}',S^{S'}(\mathsf{pk},\mathsf{pk}'))$ and $(\mathsf{pk},\mathsf{pk}',S^{R^f_{\mathsf{sk},\mathsf{pk}}}(\mathsf{pk},\mathsf{pk}')))$ are negligibly close. □

4 Fully Homomorphic Encryption from Many-to-one Functional Re-encryption

In this section, we connect the notion of obfuscated many-to-one functional re-encryption with FHE, by presenting a generic construction from the former to the latter. In particular, we assume the possibility of obfuscating functional-re-encryption for specific families of functions, which we will discuss first.

4.1 Universal Operations and Circuits

We define the notion of an (unobfuscated) re-encryption circuit that applies a universal operation to its inputs. In particular, for a message space $\mathcal{M} = \{\mathcal{M}_n\}_{n\in\mathbb{N}}$ (e.g., $\mathcal{M} = \{0,1\}$), let $\mathcal{F} = \{\mathcal{F}_n\}$ be a universal class of functions, i.e., such that $\mathcal{F}_n$ is small enough (i.e., polynomial in n, though usually constant) and such that every function $\mathcal{M}_n^q \to \mathcal{M}_n$ can be computed by circuits having gates implementing functions from $\mathcal{F}_n$. For example, we could have $\mathcal{M}_n = \{0,1\}$ for all $n \in \mathbb{N}$, and $\mathcal{F}_n$ simply contains the NAND function. Similarly, if $\mathcal{M}_n = \mathbb{F}_q$ for some prime power q depending on n, then $\mathcal{F}$ could consists of addition and multiplication in $\mathbb{F}_q$.

As usual, the gates of the circuit with $\mathcal{F}$-gates can be divided into *layers*: any gate whose inputs consist only of input bits to the entire circuit is defined to be in layer 0, and any gate whose input consists only of outputs of layer-i gates is in layer $i+1$. Without loss of generality, we can consider circuits where each layer-i gate only outputs to layer $i+1$.

4.2 Main Construction

For $i \in \{0,1,\ldots,d\}$, let $\mathsf{PKE}_i = (\mathsf{Gen}_i, \mathsf{Enc}_i, \mathsf{Dec}_i)$ be public-key encryption schemes (later to be assumed semantically secure) with common message space $\mathcal{M}$, and let $\mathcal{F}$ be a universal family of functions for $\mathcal{M}$. Also, for all $f \in \mathcal{F}$ and $i \in \{0,1,\ldots,d-1\}$, let $\mathcal{R}_i^f = \{R^{f,i}_{\mathsf{sk}_i,\mathsf{pk}_{i+1}}\}$ be the a f-re-encryption functionality from PKE_i to PKE_{i+1}. Moreover, assume we have an obfuscator Obf_i^f for $\mathcal{R}_i^f$.

We construct a d-leveled FHE scheme $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ as follows:

- $\mathsf{Gen}(1^n)$: Run $\mathsf{Gen}^{(i)}$ to generate $(\mathsf{pk}_i, \mathsf{sk}_i) \xleftarrow{\$} \mathsf{Gen}^{(i)}$ for all $i = 0,1,\ldots,d$. Let the public key $\mathsf{pk} = (\mathsf{pk}_0, \ldots \mathsf{pk}_d)$, and let the evaluation key $\mathsf{evk} = (\{\mathsf{Obf}_0^f(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \ldots \mathsf{Obf}_{d-1}^f(R^{f,d-1}_{sk_{d-1},pk_d})\}_{f\in\mathcal{F}})$. The secret key is $\mathsf{sk} = (\mathsf{sk}_0, \ldots \mathsf{sk}_d)$.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Return $c = \mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m)$.
- $\mathsf{Dec}_{\mathsf{sk}}(c)$: Run $\mathsf{Dec}_{sk_d}(c)$. (For depths i less than d, other sk_i may be used.)
- $\mathsf{Eval}_{\mathsf{evk}}(B, c_1, \ldots, c_q)$, where B is a circuit consisting of $\mathcal{F}$ gates of depth at most d and with q inputs: Start with $c_1, \ldots, c_q$ as values on the q input wires, and for each r-ary gate f with inputs at layer $i = 0,1,\ldots,d-1$ with value $c'_1, \ldots, c'_r$ on the input layers, run $\mathsf{Obf}_i^f(R^{f,i}_{\mathsf{sk}_{i-1},pk_{i+1}})$ on inputs $c'_1, \ldots, c'_q$, and assign the resulting value c'' to the output wire.

Remark. In many situations, the encryption schemes PKE_i may present some partial homomorphism properties, i.e., it may allow for computing some function $f \in \mathcal{F}$ (e.g., addition in $\mathbb{F}_q$) without resorting to re-encryption. In these situations, the obvious efficiency improvements can be made for the scheme, avoiding the use of re-encryption to compute f gates. We dispense with a formal specification of the construction in this case.

4.3 Security

We will prove the following theorems, which are the main result of this section.

Theorem (Security of the Main Construction). *Assume that PKE_0 is semantically secure, and that for all $i \in \{0, \ldots, d-1\}$ and $f \in \mathcal{F}$, the obfuscators Obf_i^f strongly securely obfuscate the f re-encryption functionality $\mathcal{R}_i^f$. Then the Main Construction above is a semantically-secure d-leveled FHE scheme.*

The following result shows that if $\mathcal{F} = \{f\}$, i.e., only one function is contained, then we can instead use the weaker notion of (non-strong) obfuscation.[1]

Theorem (Security of the Main Construction – Single Function Case). *Assume that PKE_0 is semantically secure, and that for all $i \in \{0, \ldots, d-1\}$, the obfuscator Obf_i^f securely obfuscates the f re-encryption functionality $\mathcal{R}_i^f$. Then the Main Construction above is a semantically-secure d-leveled FHE scheme.*

For both theorems, note that correctness is obvious by the definition of the re-encryption functionality and the correctness properties of the obfuscators. We are going to focus on proving the second theorem, as the proof is in fact more complicated than in the first case.

Therefore, as the core of our proof, we wish to show that the above construction achieves semantic security. Specifically, we show that for all PPT $\mathcal{D}$,

$$\begin{aligned}|\mathsf{Pr}[(\mathsf{sk}, \mathsf{pk}, \mathsf{evk}) \leftarrow \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{Enc}_{\mathsf{pk}}(m), \mathsf{pk}, \mathsf{evk}) = 1 \\ -\mathsf{Pr}[(\mathsf{sk}, \mathsf{pk}, \mathsf{evk}) \leftarrow \mathsf{Gen}(1^n) : \mathcal{D}(\mathsf{Enc}_{\mathsf{pk}}(0), \mathsf{pk}, \mathsf{evk}) = 1]| < negl(n)\end{aligned}$$

where the probability is taken over the random coins of Gen and of the encryptions.

To this end, we first prove a useful lemma to show that we can securely chain together obfuscators to perform multiple operations on an underlying message.

Lemma. *For all $m \in \mathcal{M}$, there exists PPT simulator S^* such that*

$$\begin{aligned}|\mathsf{Pr}[(\mathsf{pk}, \mathsf{evk}, \mathsf{sk}) \xleftarrow{\$} \mathsf{Gen}(1^n) : D(\mathsf{Enc}_{\mathsf{pk}}(m), \mathsf{pk}, \mathsf{evk}) = 1] \\ - \mathsf{Pr}[(\mathsf{sk}_0, \mathsf{pk}_0) \xleftarrow{\$} \mathsf{Gen}^{(0)}(1^n) : \mathcal{D}(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0)) = 1]| < negl(n)\end{aligned}$$

where the probabilities are taken over the coins of Gen, $\mathsf{Gen}^{(0)}$, the encryptions, and the simulator S^*.

[1] There are multiple reasons why $\mathcal{F}$ may only contain one function: EIther f is the NAND function or the underlying scheme already provides some level of homomorphism (e.g. additions).

Proof. The real distribution $(\mathsf{Enc}_{\mathsf{pk}}(m), \mathsf{pk}, \mathsf{evk})$ can be rewritten explicitly as

$$(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, \mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, ..., \mathsf{Obf}^f_{d-1}(R^{fd-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d) \ .$$

We now use a hybrid argument to show that this distribution is computationally indistinguishable from the simulated distribution

$$(\mathsf{Enc}^{(0)}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0)) \ ,$$

for a simulator S^* which is given below.

To do this, we construct a series of distributions, and argue that a polynomial-time distinguisher cannot notice a difference at each step, except with negligible probability.

Distribution 0 : The distinguisher is given the "real-world view"

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, \mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, ...\mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d) \ .$$

Distribution 1 : Let S_{d-1} be the simulator guaranteed by the security of Obf^f_{d-1}. The distinguisher is given

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, \mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, ..., \mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}}), \mathsf{pk}_{d-1}, S_{d-1}(\mathsf{pk}_{d-1}))$$

That is, the only change from Distribution 0 is that $(\mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d)$ is replaced by $S_{d-1}(\mathsf{pk}_{d-1})$.

By definition, we know that $(\mathsf{pk}_{d-1}, \mathsf{Obf}^f_{d-1}(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d)$ is computationally indistinguishable from $(\mathsf{pk}_{d-1}, S_{d-1}(\mathsf{pk}_{d-1}))$. The only remaining element of these distributions that depends on the values $(\mathsf{sk}_{d-1}, \mathsf{pk}_{d-1})$ is $\mathsf{Obf}^{f,d-2}(R^{f,d-2}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}})$. Note that this value only depends on pk_{d-1} and not sk_{d-1}. Thus, since we are already giving pk_{d-1} in the clear, an adversary gains no additional information about sk_{d-1} by seeing $\mathsf{Obf}^f_{d-2}(R^{f,d-2}_{\mathsf{sk}_{d-2},\mathsf{pk}_{d-1}})$. The other elements of the distribution are independent of the keys at index $d-1$ and d, so we know that the Distribution 0 is computationally indistinguishable from Distribution 1.

Distribution 2 Again, let S_{d-1} be the simulator guaranteed by the security of Obf^f_{d-1}, and let S'_{d-2} be the simulator guaranteed by the security of Obf^f_{d-2}. Define S_{d-2} as a function that applies S'_{d-2} to its input to get a pair, then applies S_{d-1} to the second element of that pair to get another pair, and outputs the 4-tuple that consists of both pairs. The distinguisher is given

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, \mathsf{Obf}^f_0(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}^f_1(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, \ldots$$
$$\mathsf{pk}_{d-3}, \mathsf{Obf}^f_{d-3}(R^{f,d-3}_{\mathsf{sk}_{d-3},\mathsf{pk}_{d-2}}), \mathsf{pk}_{d-2}, S_{d-2}(\mathsf{pk}_{d-2}))$$

This step is different from the previous step since the "pk_{d-1}" used to generate the last two elements is now itself simulated instead of being output by $\mathsf{Gen}^{(d-1)}$ directly. However, if an adversary could distinguish Distribution 2 from Distribution 1, he could use S' to break break the security of the obfuscator itself (by generating the encryption and $\mathsf{pk}_0, \ldots, \mathsf{pk}_{d-3}$ himself, using the challenge as pk_{d-2}, x, y, and running $S_{d-1}(y)$ to generate the final two elements). Thus, Distribution 2 must be computationally indistinguishable from Distribution 1.

We continue replacing pairs with a simulator in this manner until we reach:

Distribution d In Distribution d, we have replaced d (obfuscated circuit, public key) pairs with simulated values, yielding

$$(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0))$$

as desired. By hybrid argument, since each adjacent pair of distributions are computationally indistinguishable, Distribution 0 and Distribution d are computationally indistinguishable. □

We therefore know that the security of the obfuscation algorithm implies that we can use many obfuscated re-encryption algorithms in succession without breaking security. From here on, proving the semantic security of the main construction is straightforward. Indeed, assume an adversary has both pk and evk. We know that

$$\begin{aligned}(\mathsf{Enc}_{\mathsf{pk}_0}(m), m_0, \mathsf{Obf}_0^f(R^{f,0}_{\mathsf{sk}_0,\mathsf{pk}_1}), \mathsf{pk}_1, \mathsf{Obf}_1^f(R^{f,1}_{\mathsf{sk}_1,\mathsf{pk}_2}), \mathsf{pk}_2, ..., \mathsf{Obf}_{d-1}^f(R^{f,d-1}_{\mathsf{sk}_{d-1},\mathsf{pk}_d}), \mathsf{pk}_d)\\ \approx (\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0))\end{aligned}$$

for some S^*. Furthermore, since S^* is efficient, we know that the output of $S^*(\mathsf{pk}_0)$ can give no more information about sk_0 to the adversary than pk_0 itself can (since the adversary could have simply run S^* on his own). Since the original encryption scheme is semantically secure, we thus know that $(\mathsf{Enc}_{\mathsf{pk}_0}(m), \mathsf{pk}_0, S^*(\mathsf{pk}_0)) \approx (\mathsf{Enc}_{\mathsf{pk}_0}(0), \mathsf{pk}_0, S^*(\mathsf{pk}_0))$ to any PPT adversary. Thus, such an adversary can only have negligible advantage at distinguishing encryptions of m and of 0, and the FHE is semantically secure.

5 Example Construction

In this section, we exercise our framework by taking the public-key system of Regev [Reg05], which is semantically secure under the Learning With Errors assumption, and give a secure obfuscation algorithm for the multiplication-re-encryption functionality from this scheme to itself. This scheme is naturally additively homomorphic; thus, by the main theorem, this implies a (leveled) fully-homomorphic encryption scheme. Note that the resulting construction is essentially that of [Bra12]; however, we believe that viewing the problem as one of obfuscated re-encryption provides a cleaner approach.

5.1 A Public-Key Encryption Scheme

The basic public-key encryption scheme is due to Regev [Reg05]. It is parameterized by n, m, q, χ from the LWE assumption used. We will refer to this scheme as $\mathsf{PKE}_{n,q,\chi}$.

- $\mathsf{Gen}(1^k)$: Choose vector $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$, and vector $\mathbf{e} \xleftarrow{\$} \chi^m$. Compute $\mathbf{b} = \mathbf{A} \cdot \mathbf{s}' + \mathbf{e}$. Output secret key $\mathbf{s} = (\mathbf{s}', -1)$ and public key $(\mathbf{A}, \mathbf{b})$.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Given $m \in \{0, 1\}$, choose $\mathbf{r} \xleftarrow{\$} \{0, 1\}^m$ and output $(\mathbf{A}^T\mathbf{r}, \langle \mathbf{b}, \mathbf{r} \rangle + \lfloor \frac{q}{2} \rfloor \cdot m)$.
- $\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c})$: Compute $(\langle \mathbf{s}, \mathbf{c} \rangle \pmod q)$. Output 0 if this value is closer to 0 and 1 if this value is closer to $\lfloor \frac{q}{2} \rfloor \pmod q$.

This encryption scheme is semantically secure under the $LWE_{q,\chi}$ assumption [Reg05]. Furthermore, it is clearly additively homomorphic over $GF[2]$ (for appropriate choice of χ), since $(\langle \mathbf{s}, \mathbf{c}_1 + \mathbf{c}_2 \rangle \pmod q) = \lfloor \frac{q}{2} \rfloor \cdot (m_1 + m_2) - \langle \mathbf{e}, \mathbf{r}_1 + \mathbf{r}_2 \rangle \pmod q$.

5.2 Re-encryption and Obfuscation

Re-encryption functionality. We consider the family of circuits $\mathcal{R}^{\times}$, the re-encryption-with-multiplication circuits from $\mathsf{PKE}_{n,q,\chi}$ to $\mathsf{PKE}_{n,q,\chi'}$. (The values n and q could change as well, if desired.) A circuit $R^{\times}_{\mathsf{sk},\mathsf{pk}'} \in \mathcal{R}^{\times}$ contains the secret key $\mathsf{sk} = \mathbf{s}$ of a scheme in $\mathsf{PKE}_{n,q,\chi}$ and the public key $\mathsf{pk}' = (\mathbf{A}', \mathbf{b}')$ of a scheme in $\mathsf{PKE}_{n,q,\chi'}$, hardwired inside. It takes as input two ciphertexts and applies $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ to each of them to obtain two bits. It multiplies these two bits (corresponding to a logical and), runs $\mathsf{Enc}_{\mathsf{pk}'}(\cdot)$ on the result, and outputs the resulting ciphertext.

Construction of Obf. To construct our obfuscator, we first define transformations $\mathsf{BitDecomp}$ and $\mathsf{PowersOf2}$ (used previously in [BV11a], [BGV11], [Bra12], [GSW13]). If $\mathbf{v} = (v_1, v_2, ...v_\ell) \in \mathbb{Z}_q^\ell$, then:

- $\mathsf{BitDecomp}_q(\mathbf{v}) = (v_{1,0}, v_{1,1}, ...v_{1,\lceil \lg q \rceil}, v_{2,0}, ...v_{\ell,\lceil \lg q \rceil})$, where $v_{i,j}$ is the j-th least significant bit of v_i (that is, $v_i = \sum_j 2^j v_{i,j}$).
- $\mathsf{PowersOf2}_q(\mathbf{v}) = (v_1, 2v_1, 4v_1, ...2^{\lceil \lg q \rceil} \cdot v_1, v_2, 2v_2 ...2^{\lceil \lg q \rceil} \cdot v_\ell)$.

In the following, we will generally omit the subscript q. Of note is that for any $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathsf{BitDecomp}(\mathbf{u}), \mathsf{PowersOf2}(\mathbf{v}) \rangle$.

We will describe the transformation we want Obf to perform first, and then define its circuit output. We first compute $\tilde{\mathbf{s}} = \frac{2}{q}(\mathsf{BitDecomp}(\mathbf{s}) \otimes \mathsf{BitDecomp}(\mathbf{s}))$, a rational vector of length $((n+1)\lceil \lg q \rceil)^2$. Here $\otimes$ denotes the tensor product.

We then use $\mathsf{pk}' = (\mathbf{A}', \mathbf{b}')$ to "encrypt"[2] each element of $\mathsf{PowersOf2}(\tilde{\mathbf{s}})$. That is, we choose $\mathbf{R} \xleftarrow{\$} \{0,1\}^{((n+1)^2 \lceil \lg q \rceil^3) \times m}$ and compute $\mathbf{D} = [\mathbf{A}'|\mathbf{b}']^T \cdot \mathbf{R} + \frac{q}{2}[\mathbf{0}|\mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T$, where $\mathbf{0}$ is an $m \times n$ matrix of zeroes. (Note that $\mathbf{D}$ is an integer matrix.)

Define $\tilde{\mathbf{c}} = \frac{2}{q}(\mathsf{PowersOf2}(\mathbf{c}_1) \otimes \mathsf{PowersOf2}(\mathbf{c}_2))$. Obf will extract $\mathbf{s}$ and $(\mathbf{A}', \mathbf{b}')$ from its input. Then it constructs a randomized circuit that chooses a random $\mathbf{R}$ as defined above and computes the corresponding $\mathbf{D}$. The circuit takes in two input ciphertexts $\mathbf{c}_1$ and $\mathbf{c}_2$, computes $\mathbf{D} \cdot \mathsf{BitDecomp}(\lfloor \tilde{\mathbf{c}} \rceil)$, and outputs this value. Obf outputs this circuit as the obfuscation of $R^{\times}_{\mathsf{sk},\mathsf{pk}'}$.

[2] As in [BV11a], this is not true encryption, since the encrypted values are not bits; thus, they cannot be decrypted properly. However, the operation is the same, and the intuition that these values are "encrypted" may be useful.

Correctness. The circuit $\mathsf{Obf}(R^{\times}_{\mathsf{sk},\mathsf{pk}'})$ calculates

$$
\begin{aligned}
&\mathbf{D}\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R}\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil)+\frac{q}{2}[\mathbf{0}|\mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T\cdot\mathsf{BitDecomp}(\lfloor\tilde{\mathbf{c}}\rceil)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'+\frac{q}{2}(0^n,\langle\tilde{\mathbf{s}},\lfloor\tilde{\mathbf{c}}\rceil\rangle)\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'+(0^n,\langle\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s}),\frac{2}{q}(\mathsf{PowersOf2}(\mathbf{c}_1)\otimes\mathsf{PowersOf2}(\mathbf{c}_2)\rangle))+\mathbf{e}'_1\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'+\frac{2}{q}(0^n,\langle\mathbf{s},\mathbf{c}_1\rangle\cdot\langle\mathbf{s},\mathbf{c}_2\rangle)+\mathbf{e}'_1\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'+\frac{2}{q}(0^n,(\langle\mathbf{e}_1,\mathbf{r}_1\rangle+\frac{q}{2}m_1)(\langle\mathbf{e}_2,\mathbf{r}_2\rangle+\frac{q}{2}m_2))+\mathbf{e}'_1\\
&\quad= [\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'+\frac{q}{2}(0^n,m_1m_2)+\mathbf{e}'_1+\mathbf{e}'_2
\end{aligned}
$$

We wish to show that this is statistically close to the output of $R^{\times}_{\mathsf{sk},\mathsf{pk}'}$ (which is a fresh encryption of m_1m_2). There are two differences: the fact that $\mathbf{r}'$ is not a binary vector, and the presence of an additional additive error term $(\mathbf{e}_1+\mathbf{e}_2)$.

For the first difference, note that $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'\in\mathbb{Z}_q^n$, and that both $\mathbf{A}'$ and $\mathbf{R}$ are chosen randomly. There are 2^m choices of $\mathbf{r}''\in\{0,1\}^m$. Thus, for a value $m=\Omega(n\lg q)$, with high probability there exists $\mathbf{r}''\in\{0,1\}^m$ such that $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}'=[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{r}''$.

For the second difference, we note that both $\mathbf{e}'_1$ and $\mathbf{e}'_2$ are "small". Specifically, $\mathbf{e}'_1$ comes from rounding error; each element is rounded by at most $1/2$, so its magnitude is bounded[3] by $||\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s})||_1\cdot\frac{1}{2}\leq((n+1)(\lceil\lg q\rceil+1))^2/2$. $\mathbf{e}'_2$ is due to the presence of $\mathbf{e}_1$ and bfe_2 in the original ciphertexts; however, the presence of the $\frac{2}{q}$ coefficient means that this term is bounded by $O(m\varepsilon)$, where ε is the original error bound of χ. Note that the magnitude of $(\mathbf{e}_1+\mathbf{e}_2)$ is *independent* of q aside from a logarithmic factor; thus, we can choose the LWE parameters (in particular, q and χ') such that the output distributions of the obfuscated and unobfuscated circuits are statistically close.

Simulatability. We show a simulator S that satisfies the strong simulatability condition for this construction, as defined in section 3.2. Recall that $\mathsf{Obf}(R^{\times}_{\mathsf{sk},\mathsf{pk}'})$ constructs a circuit that only depends on the values $(\mathsf{sk},\mathsf{pk}')$ through a matrix $\mathbf{D}$, defined as $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R}+\frac{q}{2}[\mathbf{0}|\mathsf{PowersOf2}(\tilde{\mathbf{s}})]^T$. The simulator S simply chooses $\mathbf{R}\xleftarrow{\$}\{0,1\}^{((n+1)^2\lceil\lg q\rceil^3)\times m}$ and returns a circuit that uses $[\mathbf{A}'|\mathbf{b}']^T\cdot\mathbf{R}$ in place of $\mathbf{D}$.

Note that this is simply a Regev encryption of 0 under the key pk'; indistinguishability holds by the semantic security of the original Regev scheme.

6 Bootstrapping

Many existing FHE schemes, starting with that of Gentry [Gen09], operate on the principle of "bootstrapping". That is, they first define a "somewhat homomorphic" scheme, which is capable

[3] Bounding this error is the reason to introduce $\mathsf{BitDecomp}$ and $\mathsf{PowersOf2}$–this allows the vector $\mathsf{BitDecomp}(\mathbf{s})\otimes\mathsf{BitDecomp}(\mathbf{s})$ to be binary.

of homomorphically evaluating its own decryption circuit plus a single operation under a single key. They then provide a chain of encrypted keys under this scheme, where the i-th decryption key is encrypted under the $(i+1)$st key. This construction allows for (leveled) fully-homomorphic evaluation: given a ciphertext encrypted under the i-th key, the evaluator encrypts the ciphertext under the $(i+1)$st key and then homomorphically evaluates the decryption circuit on the new ciphertext and the encrypted i-th key, followed by one operation. The net result is an encryption under key $i+1$ of the operation applied to the plaintext corresponding to the input.

The general bootstrapping paradigm can be seen under our framework as providing an obfuscated re-encryption-with-operation functionality. Specifically, given the keys pk_{i+1}, sk_i, one can construct a circuit that encrypts its input under pk_{i+1}, runs the decryption operation homomorphically using a hardcoded value $\mathsf{Enc}_{pk_{i+1}}(sk_i)$, and then homomorphically performs one operation. This circuit performs the same computation as decrypting, performing the operation, and encrypting (by the correctness of the FHE scheme), and does not leak any information about the encrypted data (by the semantic security of the FHE scheme). Thus, at a high level it is an obfuscated re-encryption-with-operation circuit under our definition. However, our definition is more general, since we do not require starting with a "somewhat homomorphic" encryption scheme, but any semantically-secure encryption scheme with a securely-obfuscatable f-re-encryption functionality.

7 Acknowledgements

The authors would like to thank Shafi Goldwasser for her help and guidance.

References

[AW07] Ben Adida and Douglas Wikström. How to shuffle in public. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 555–574. Springer, February 2007.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, August 2001.

[BGV11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive, Report 2011/277, 2011. http://eprint.iacr.org/.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, August 2012.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 97–106. IEEE Computer Society Press, October 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, August 2011.

[CCV12] Nishanth Chandran, Melissa Chase, and Vinod Vaikuntanathan. Functional re-encryption and collusion-resistant obfuscation. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 404–421. Springer, March 2012.

[CMNT11] Jean-Sébastien Coron, Avradip Mandal, David Naccache, and Mehdi Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 487–504. Springer, August 2011.

[CNT12] Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 446–464. Springer, April 2012.

[CRV10] Ran Canetti, Guy N. Rothblum, and Mayank Varia. Obfuscation of hyperplane membership. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 72–89. Springer, February 2010.

[DS05] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 654–663. ACM Press, May 2005.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, May / June 2009.

[GH11] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In Rafail Ostrovsky, editor, *52nd Annual Symposium on Foundations of Computer Science*, pages 107–109. IEEE Computer Society Press, October 2011.

[GHS12a] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, April 2012.

[GHS12b] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer, August 2012.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *46th Annual Symposium on Foundations of Computer Science*, pages 553–562. IEEE Computer Society Press, October 2005.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. Cryptology ePrint Archive, Report 2013/340, 2013. http://eprint.iacr.org/.

[HRSV07] Susan Hohenberger, Guy N. Rothblum, Abhi Shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 233–252. Springer, February 2007.

[RAD78] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. In Richard A. DeMillo, David P. Dobkin, Anita K. Jones, and Richard J. Lipton, editors, *Foundations of Secure Computation*, pages 165–179. Academic Press, 1978.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 84–93. ACM Press, May 2005.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, May 2010.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer, May 2010.

[Wee05] Hoeteck Wee. On obfuscating point functions. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 523–532. ACM Press, May 2005.