Fully Composable Homomorphic Encryption
DRAFT

Daniele Micciancio

October 2, 2023

1 Introduction

A homomorphic encryption scheme is a cryptosystem that supports the evaluation of arbitrary functions or programs on encrypted data, without using the secret decryption key. More specifically, an encryption scheme is $\mathcal{F}$-*homomorphic* (for some class of functions $\mathcal{F}$) if for any $f\colon \mathcal{M}^w \to \mathcal{M}$ in $\mathcal{F}$ and message vector $\mathbf{m} \in \mathcal{M}^w$, given $\mathbf{c} = \mathsf{Enc}(\mathbf{m})$, one can publicly compute a ciphertext $c = \mathsf{Eval}(f, \mathbf{c})$ that decrypts to $\mathsf{Dec}(c) = f(\mathbf{m})$. (See Figure 1 and Definition 5.) An encryption scheme is called *fully* homomorphic if it supports the computation of arbitrary programs, i.e., if $\mathcal{F}$ is the set of all possibly (efficiently computable) functions. This is the standard notion of fully homomorphic encryption (FHE), as used by Gentry's first FHE candidate construction [Gen09b, Gen09a], as well as much subsequent work. (E.g., see recent surveys [Hal17, Bra19].) While this definition closely models the intended use of homomorphic encryption schemes in typical applications, it has some shortcomings.

Stronger notions of security have also been considered (e.g., providing function privacy [Gen09b, BPMW16, OPP14, DD22, DS16]) but they are typically understood as "optional" features, and can often be obtained by extending a scheme satisfying the basic definition. In this paper we focus on a different issue: composability, i.e., the ability to concatenate (or, more generally, combine) different homomorphic computations together. The importance of composability, and the fact that it is not guaranteed by the standard definition of homomorphic correctness, was first pointed out by Gentry, Halevi and Vaikuntanathan [GHV10]. In fact, in a regular homomorphic encryption scheme, the evaluation function Eval_f may in general use different types of input and output ciphertexts, so that the output of Eval_f cannot even be used as input to another homomorphic evaluation

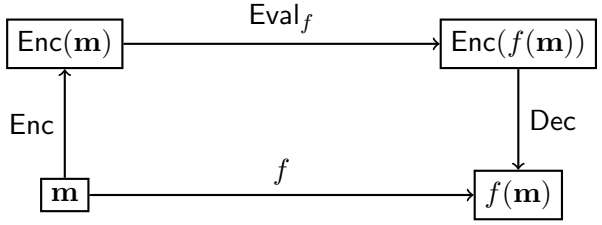

Figure 1: In a homomorphic encryption scheme, evaluating a function f homomorphically on the encryption of a message $\mathbf{m}$ produces a ciphertext that decrypts to $f(\mathbf{m})$.

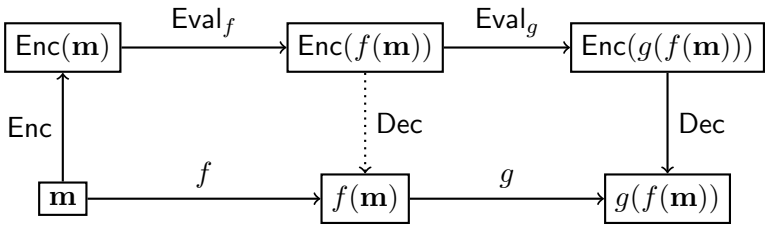

Figure 2: A 2-hop homomorphic encryption scheme supports the consecutive homomorphic evaluation of 2 functions, f, g. More generally, a i-hop encryption scheme allows to chain i homomorphic computations $\mathsf{Eval}_{f_1}, \ldots, \mathsf{Eval}_{f_i}$, mapping an encryption of m, to a ciphertext that decrypts to $f_i(f_{i-1}(\ldots f_1(m)))$.

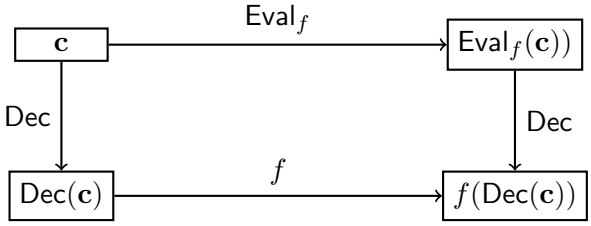

Figure 3: In a fully composable homomorphic encryption scheme, the homomorphic evaluation function commutes with decryption.

Eval_g. In order to address the composability problem, [GHV10] proposed a stronger notion of correctness, called *i-hop homomorphic encryption*. In a i-hop encryption scheme [GHV10], one can consecutively evaluate up to i functions homomorphically on a ciphertext $\mathbf{c} = \mathsf{Enc}(\mathbf{m})$, and the final result $c' = \mathsf{Eval}_{f_i}(\mathsf{Eval}_{i-1}(\cdots \mathsf{Eval}_{f_1}(\mathbf{m})))$ will be a ciphertext that decrypts to $\mathsf{Dec}(c') = f_i(f_{i-1}(\cdots f_1(\mathbf{m})))$. (See Figure 2 for an illustration for the case of $i = 2$.) The standard correctness definition corresponds to the special case when $i = 1$, and it is also called *single-hop* homomorphic encryption. If a scheme supports is i-hop homomorphic for any integer i, then it is called *multi-hop*.

The multi-hop property, while desirable, is rather involved, as it requires considering the homomorphic evaluation of an arbitrary number of functions $f_1, f_2, \ldots$. In this paper we investigate a different approach to achieve composability, which we call *fully composable* homomorphic encryption. Technically, a scheme is fully composable if the homomorphic evaluation function Eval_f commutes with the decryption function Dec: evaluating a function f homomorphically on a set ciphertexts $\mathbf{c}$ and then decrypting $\mathsf{Eval}_f(\mathbf{c})$ should produce the same result as first decrypting $\mathbf{m} = \mathsf{Dec}(\mathbf{c})$, and then computing $f(\mathbf{m})$ in the clear. (See Figure 3.) So, syntactically, the definition is as simple as the standard notion of homomorphic encryption, involving the evaluation of a single function. Still, it achieves very strong composition properties.

In this paper we investigate this notion of full composability and its relation to the other definitions of (fully) homomorphic encryption. In particular, we show that

1. Fully composable encryption satisfies the standard notion of homomorphic correctness (Theorem 1), and it is also composable, in the sense that it supports arbitrary computations, as

described by circuits with gates in the basic set of functions $\mathcal{F}$ (Theorem 2)

2. single-hop, 2-hops, 3-hops, ..., multi-hop and fully composability form a sequence of strictly stronger requirements, in the sense that each one is implied by the next, but (under minimal assumptions) there are schemes satisfying one notion but not the next one. (Theorem 3, Theorem 4.)

3. For a general class of homomorphic encryption schemes (satisfying a certain surjectivity property, see Definition 9) multi-hop correctness is equivalent to full composability. (Theorem 5.)

4. Finally, Gentry's celebrated bootstrapping technique [Gen09b] can be formulated as a method to transform a (single-hop, circularly secure) homomorphic encryption scheme into a fully composable one. (Theorem 6.)

As a contribution of possibly independent interest, we show how all definitions, implications and separation results, can be generalized, and extended to functions describing "partial" computations. More specifically, we consider functions $f\colon \mathcal{M}_{\perp}^{w} \to \mathcal{M}_{\perp}$ that may take as input (or produce as output) a special "undefined" symbol $\perp$. The symbols is special, in the sense that it does not represent an actual message, but the fact that the message is unknown or unavailable. These generalized functions can be used to model several types of computations of practical interest like the following:

- Computations with restricted domain $f\colon \mathcal{X} \to \mathcal{M}$ where $\mathcal{X} \subset \mathcal{M}^{w}$. Consider for example bounded integer arithmetic on $\mathcal{M} = \{-B, \ldots, B\} \subset \mathbb{Z}$, where the result of an arithmetic operation (e.g., $x + y$ or $x \cdot y$ for $x, y \in \mathcal{M}$) is set to $\perp$ if it falls outside the range $\mathcal{M}$. This is different from modular arithmetics, because in case of overflow, the function evaluates to $\perp$, and the output of the homomorphic computation $\mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})$ can be arbitrary.

- Computations that may recover from partially invalid inputs. For example, consider an "error correcting" function such that $f(m_1, m_2, m_3) = m$ if at least two of the three inputs equal m, even if the third input may be $m_i = \perp$ or a different value $m_i \neq m$.

2 Definitions

In this section we recall the standard notion of (homomorphic) encryption scheme and (circular) security against chosen plaintext attacks.

Definition 1 (Encryption scheme) *A public key encryption scheme with message space $\mathcal{M}$ is a triple of (probabilistic polynomial time) algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *where*

- *The* Key Generation *algorithm* Gen*, on input a security parameter κ, outputs a pair of (secret and public) keys* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$.

- *The* Encryption *algorithm* Enc *on input key* pk *and message $m \in \mathcal{M}$, outputs a ciphertext* $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$.

- *The* Decryption *algorithm* $\mathsf{Dec}(\mathsf{sk}, c)$*, on input a secret key* sk *and ciphertext c, outputs either a message $m \in \mathcal{M}$ or a special "failure" symbol $\perp$.*

We say that the scheme is valid *if it satisfies the* correctness *property*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m \tag{1}$$

for all messages $m \in \mathcal{M}$ *and keys* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$.

The symbol $\perp$ (output by the decryption algorithm) is special, in the sense that it does not represent a regular message, but denotes some kind of failure condition, e.g., when trying to decrypt an invalid ciphertext. Formally, this is modeled by endowing the extended message space $\mathcal{M}_\perp = \mathcal{M} \cup \{\perp\}$ with the flat partial ordering relation $\sqsubseteq$ where $x \sqsubseteq y$ if and only if $x = \perp$ or $x = y$. Then, the correctness condition (1) is equivalent to requiring[1]

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \sqsupseteq m \tag{2}$$

for all $m \in \mathcal{M}_\perp$, where $\mathsf{Enc}(\mathsf{pk}, \perp)$ may output any ciphertext. Notice that when $m \in \mathcal{M}$, condition (2) becomes an equality, and (1) is satisfied. On the other hand, when $m = \perp$, (2) imposes no restrictions on the output of Dec. In other words, the decryption algorithm is not required to detect invalid ciphertexts, and (as far as Definition 1 is concerned) Dec may output any fixed message $m \in \mathcal{M}$ instead of $\perp$. The extension to $\mathcal{M}_\perp$ will become more useful when treating homomorphic encryption schemes.

We assume the scheme satisfies perfect correctness, i.e., we require (1) to hold with probability 1, over the choice of the keys sk, pk and encryption randomness. The correctness condition could be relaxed to hold with overwhelming probability. But assuming perfect correctness is easier.

We focus on encryption schemes with a finite, fixed-length message space,[2] as these can be extended to variable length messages $\mathcal{M}^* = \bigcup_{\ell \geq 0} \mathcal{M}^\ell$ by letting the encryption and decryption functions operate on message sequences componentwise:

$$\begin{aligned} \mathsf{Enc}^*(\mathsf{pk}, m_1, \ldots, m_w) &= (\mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_w)) \\ \mathsf{Dec}^*(\mathsf{sk}, c_1, \ldots, c_w) &= (\mathsf{Dec}(\mathsf{sk}, c_1), \ldots, \mathsf{Dec}(\mathsf{sk}, c_w)). \end{aligned}$$

It is immediate to show that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a valid encryption scheme over fixed-length message space $\mathcal{M}$, then $(\mathsf{Gen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ is a valid encryption scheme over variable-length messages $\mathcal{M}^*$.

The standard notion of security against passive adversaries for encryption schemes is that of *indistinguishability under chosen plaintext attack* (IND-CPA) or semantic security [GM84].

Definition 2 (IND-CPA security) *An encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *satisfies* indistinguishability under chosen plaintext attack *(IND-CPA security for short) if any efficient (probabilistic polynomial time, stateful) adversary* $\mathcal{A}$ *has negligible advantage in the game defined by the following steps:*

1. $b \leftarrow \{0,1\}$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$
2. *The adversary* $(m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk})$ *selects a pair of messages* $m_0, m_1 \in \mathcal{M}$ *of equal-length.*[3]

[1] As usual, we write $x \sqsupseteq y$ if $y \sqsubseteq x$.

[2] For example, $\mathcal{M} = \{0,1\}$ for single bit messages. The set $\mathcal{M}$ may still depend on the security parameter κ, e.g., $\mathcal{M} = \{0,1\}^\kappa$ for the set of bitstrings of fixed length κ.

[3] If $\mathcal{M}$ is a fixed-length message space, then this requirement is trivially satisfied. If $m_0, m_1 \in \mathcal{M}^*$ are variable length messages, then it must be $m_0, m_1 \in \mathcal{M}^k$ for the same k.

3. The adversary is given a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$ *and outputs a bit* $b' \leftarrow \mathcal{A}(c)$

The adversary is successful if $b' = b$*, and the advantage* δ *is defined as* $2\beta - 1$*, where* β *is the probability that* $b' = b$*.*

It easily follows by a standard hybrid argument that if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is IND-CPA secure for fixed length messages $\mathcal{M}$, then its extesion $(\mathsf{Gen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ to variable length messages $\mathcal{M}^*$ is also IND-CPA secure.

A slightly stronger definition (*Pseudorandomness under Chosen Plaintext Attack*, or RND-CPA) has the adversary select a single message $m \leftarrow \mathcal{A}(\mathsf{pk})$, and receive either its encryption $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ (if $b = 0$) or a randomly chosen ciphertext $c \leftarrow \mathcal{C}$ (if $b = 1$).[4] It is easy to show that any RND-CPA secure encryption scheme is also IND-CPA secure, but the converse is not necessarily true: any IND-CPA secure encryption scheme can be easily modified to make it RND-CPA *insecure*[5], while preserving IND-CPA security. RND-CPA security not only hides the encrypted message, but also provides some form of anonymity, as the set $\mathcal{C}$ does not depend on the value of the keys $(\mathsf{pk}, \mathsf{sk})$. Lattice-based encryption schemes (and, with them, virtually all known fully homomorphic encryption constructions) typically satisfy this slightly stronger definition of security. For simplicity we restrict our attention to the standard IND-CPA security definition, but all definitions and proofs can be easily adapted to RND-CPA security as well.

2.1 Circular security

An encryption scheme satisfies *circular security* if it remains secure even against adversaries that are given an encryption of the secret key, or, more precisely, an encoding of the secret key $\psi(\mathsf{sk}) \in \mathcal{M}^w$ as a sequence of elements in the message space. Formally, circular security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ can be defined in terms of the (standard) IND-CPA security of a scheme with modified key generation and encryption algorithms as follows:

- $\mathsf{Gen}^\psi(\kappa) = (\mathsf{sk}, (\mathsf{pk}, \mathsf{pk}'))$ where $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$ and $\mathsf{pk}' \leftarrow \mathsf{Enc}^*(\mathsf{pk}, \psi(\mathsf{sk}))$
- $\mathsf{Enc}^\psi((\mathsf{pk}, \mathsf{pk}'), m) = \mathsf{Enc}(\mathsf{pk}, m)$

Informally, the new key generation algorithm Gen^ψ appends an encryption of $\psi(\mathsf{sk})$ to the public key. This extra information is ignored by the encryption function, but is available to an adversary attacking the scheme.

Definition 3 *For any key encoding function* $\psi\colon \mathcal{K} \to \mathcal{M}^w$*, a public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* ψ*-circular IND-CPA secure if the modified scheme* $(\mathsf{Gen}^\psi, \mathsf{Enc}^\psi, \mathsf{Dec})$ *is IND-CPA secure.*

The definition of circular security and the results in this paper are easily extended to encryption cycles of length longer than one, or even arbitrary encryption graphs $G = (V, E)$ with a pair of keys $(\mathsf{sk}_v, \mathsf{pk}_v)$ associated to every node $v \in V$ and a public ciphertext $\mathsf{Enc}^*(\mathsf{pk}_v, \psi_e(\mathsf{sk}_u))$ associated to every edge $e = (u, v) \in E$. But for simplicity, we focus on simple loops involving a single secret key.

[4] More specifically, we assume that, for any fixed value of the security parameter κ, and for all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$, we have $\mathsf{Enc}(\mathsf{pk}, \mathcal{M}) \subseteq \mathcal{C}$ for some set $\mathcal{C}$ independent of the encryption key pk such that membership in $\mathcal{C}$ can be efficiently tested and the uniform (or other standard) distribution on $\mathcal{C}$ can be efficiently sampled.

[5] E.g., simply let the encryption algorithm add a fixed prefix to the output ciphertext.

2.2 Homomorphic Encryption

A homomorphic encryption scheme allows to perform computations on encrypted data using a publicly computable *evaluation* algorithm Eval. We define a simple, but powerful generalization of the standard notion of homomorphic encryption, with functions $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ over the extended message space $\mathcal{M}_\perp = \mathcal{M} \cup \{\perp\}$. More specifically, we consider *morphisms* $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$, i.e., functions between partially ordered sets[6] such that $f(\perp, \ldots, \perp) = \perp$ and $f(\mathbf{x}) \sqsubseteq f(\mathbf{y})$ for all $\mathbf{x} \sqsubseteq \mathbf{y}$.

Definition 4 *A homomorphic encryption scheme with message space $\mathcal{M}$ and morphisms $\mathcal{F} \subseteq \bigcup_{w\geq 0}\{f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp\}$ is a encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space $\mathcal{M}$ together with an* evaluation *algorithm* Eval *that on input a public key* pk*, a morphism $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ in $\mathcal{F}$, and a sequence $\mathbf{c} \in \mathcal{C}^w$, outputs a ciphertext* $\mathsf{Eval}(\mathsf{pk}, f, \mathbf{c}) \in \mathcal{C}$.

The standard definition of correctness for homomorphic encryption schemes requires that for any function $f\colon \mathcal{M}^w \to \mathcal{M}$, encrypting some data $\mathbf{c} = \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m})$, evaluating the function f homomorphically $c = \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})$, and decrypting the final result $\mathsf{Dec}(\mathsf{sk}, c)$, produces the same value as computing $f(\mathbf{m})$ in the clear. This is extended to morphisms $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ as follows.

Definition 5 (Homomorphic correctness) *An encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is $\mathcal{F}$-homomorphic if for any keys* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$*, morphism $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ in $\mathcal{F}$, and messages $\mathbf{m} \in \mathcal{M}_\perp^w$, we have*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) \sqsupseteq f(\mathbf{m}). \tag{3}$$

Notice that when $f(\mathbf{m}) = \perp$, condition (3) says nothing about the output of Dec. We argue that this is the most natural generalization of the standard notion of correctness. Informally, the decryption algorithm is not required to detect if f was evaluated homomorphically on the encryption of an invalid input $\mathbf{m}$. In order to compare these generalized homomorphic computations to previous definitions, for any function $f\colon \mathcal{M}^w \to \mathcal{M}$, define its extension to a morphism $f_\perp\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ in the obvious way, setting $f_\perp(\mathbf{m}) = f(\mathbf{m})$ if $\mathbf{m} \in \mathcal{M}^w$, and $f_\perp(\mathbf{m}) = \perp$ otherwise. In other words, $f_\perp(m_1, \ldots, m_w)$ requires all of its input to be $m_i \neq \perp$, and outputs $\perp$ otherwise.[7] Then, the Eval function from Definitions 4 and 5 is applied to f by computing $\mathsf{Eval}(\mathsf{pk}, f_\perp, \mathbf{c})$. The next lemma shows that, when specialized to regular functions $f\colon \mathcal{M}^w \to \mathcal{M}$, the correctness condition (3) becomes an equality, matching the familiar definition of homomorphic encryption used in previous works.

Lemma 1 *For any set $\mathcal{F}$ of functions $f\colon \mathcal{M}^w \to \mathcal{M}$, an encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is $\mathcal{F}$-homomorphic if and only if for any* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$*, $f \in \mathcal{F}$, and $\mathbf{m} \in \mathcal{M}^w$,*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) = f(\mathbf{m}). \tag{4}$$

Proof Notice that for all $\mathbf{m} \in \mathcal{M}^w$, $f_\perp(\mathbf{m}) = f(\mathbf{m}) \neq \perp$. So, $f_\perp(\mathbf{m}) \sqsubseteq x$ is equivalent to $f_\perp(\mathbf{m}) = x$, and (3) becomes an equality. In the other direction, if $\mathbf{m} \notin \mathcal{M}^w$, then $f_\perp(\mathbf{m}) = \perp$ and (3) is trivially satisfied. □

Morphisms $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ can be used to model several types of computations of practical interest like the following:

[6] The flat partial ordering of $\mathcal{M}_\perp$ is extended to $\mathcal{M}_\perp^w$ componentwise, i.e., $\mathbf{x} \sqsubseteq \mathbf{y}$ if $x_i \sqsubseteq y_i$ for all i.

[7] We remark that not all morphisms $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ are required to satisfy this property. See, for example, the comments after Lemma 1.

- Computations with restricted domain $f\colon \mathcal{X} \to \mathcal{M}$ where $\mathcal{X} \subset \mathcal{M}^w$. Consider for example bounded integer arithmetic on $\mathcal{M} = \{-B, \ldots, B\} \subset \mathbb{Z}$, where the result of an arithmetic operation (e.g., $x + y$ or $x \cdot y$ for $x, y \in \mathcal{M}$) is set to $\perp$ if it falls outside the range $\mathcal{M}$. This is different from modular arithmetics, because in case of overflow, the function evaluates to $\perp$, and the output of the homomorphic computation $\mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})$ can be arbitrary.

- Computations that may recover from partially invalid inputs. For example, consider an "error correcting" function such that $f(m_1, m_2, m_3) = m$ if at least two of the three inputs equal m, even if the third input may be $m_i = \perp$ or a different value $m_i \neq m$.

So, for generality, we will present all our results in terms of morphisms. However, for comparison with previous work and definitions, we will focus on functions $f\colon \mathcal{M}^w \to \mathcal{M}$, using their standard extension $f_\perp\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$. So, we say that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is *fully homomorphic* if $\mathcal{F}$ contains all morphisms $f_\perp$, where $f\colon \mathcal{M}^w \to \mathcal{M}$. As usual, these morphisms are represented using some standard encoding. For example, if $\mathcal{M} = \{0, 1\}$, then functions may be described by boolean circuits with $|f|$ gates with bounded fan-in.

All our definitions can be further extended to morphisms $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp^v$ with multiple outputs. Efficiency aside, this is equivalent to morphisms with output in $\mathcal{M}_\perp$, as any other $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp^v$ can be expressed as v separate morphisms $f_i\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ such that $f(\mathbf{m}) = (f_1(\mathbf{m}), \ldots, f_v(\mathbf{m}))$. So, for notational simplicity, we focus on morphisms with a single output $f(\mathbf{m}) \in \mathcal{M}_\perp$.

We remark that in order to run the evaluation algorithm Eval on $f \in \mathcal{F}$, one needs to provide Eval with a concrete *description* of f, so that one can talk about the *size* of (the description of) f, and how this size and the details of the encoding affect the running time of Eval. For simplicity, we identify f with its description, and write $f(\mathbf{m})$ for the result of evaluating f at $\mathbf{m}$, and $|f|$ for the size of the description of f.

A weaker form of general purpose homomorphic computation is provided by *leveled homomorphic* encryption schemes, which can be formally defined as a sequence $(\mathsf{Gen}_\ell, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ (for $\ell = 1, 2, \ldots$) of homomorphic encryption schemes with function sets $\mathcal{F}_\ell$ such that $\mathsf{Gen}_\ell(\kappa) = \mathsf{Gen}(\kappa, \ell)$ is a key generation algorithm that takes ℓ as an auxiliary parameter, and runs in time polynomial in both κ and ℓ. In particular, this allows Enc, Dec and Eval to also run in time polynomial in ℓ. The standard example, for $\mathcal{M} = \{0, 1\}$, is to let $\mathcal{F}_\ell$ be the set of all functions computable by a boolean circuit of depth at most ℓ. We say that $(\mathsf{Gen}_\ell, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is *Leveled Fully Homomorphic* if the union $\bigcup_\ell \mathcal{F}_\ell = \{f\colon \mathcal{M}^w \to \mathcal{M} \mid w \geq 0\}$ is the set of all functions.

The definition of IND-CPA security applies to homomorphic encryption schemes unmodified, just considering the underlying scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, without taking into account the evaluation algorithm.[8] This is the basic notion of security typically used for homomorphic encryption. Stronger definitions of security are possible, e.g., hiding not only the messages, but also the computation performed on them. In this paper we focus on the basic definition of security (without function privacy), and strengthen the schemes in a different direction, making the homomorphic correctness condition composable.

[8] This is justified by the fact that $\mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})$ can be publicly computed, and does not provide additional information to an adversary that already knows $\mathbf{c}$ and f.

3 Full Composability

We propose a stronger, but compatible, definition of fully homomorphic encryption that focuses on the fact that computations in $\mathcal{F}$ can be *arbitrarily composed*.

Definition 6 (Composable FHE) $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is a* fully composable $\mathcal{F}$-homomorphic encryption *scheme (or "Composable FHE") if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a valid encryption scheme, and*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) \sqsupseteq f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) \tag{5}$$

for all keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$*, morphism* $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ *in* $\mathcal{F}$*, and ciphertexts* $\mathbf{c} \in \mathcal{C}^w$*.*

Similarly to the definition of correctness, condition (5) only requires an inequality ($\sqsupseteq$), and we assume it holds with probability 1. The next lemma shows that, when specialized to functions $f\colon \mathcal{M}^w \to \mathcal{M}$ that do not use the $\perp$ symbol, condition (5) becomes an equality ($=$), and it is required only for valid input ciphertexts $\mathbf{c}$. (By constrast, notice that Definition 6 should hold for any $\mathbf{c} \in \mathcal{C}^w$, even if $\mathsf{Dec}(\mathsf{sk}, c_i) = \perp$ for some i.

Lemma 2 *For any set of functions* $\mathcal{F} \subseteq \bigcup_w (\mathcal{M}^w \to \mathcal{M})$*,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is a fully composable* $\mathcal{F}$*-homomorphic encryption scheme if and only if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a valid encryption scheme and*

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) \tag{6}$$

for all keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$*, function* $f\colon \mathcal{M}^w \to \mathcal{M}$ *in* $\mathcal{F}$*, and ciphertexts* $\mathbf{c} \in \mathcal{C}^w$ *such that* $\mathsf{Dec}(\mathsf{sk}, c_i) \neq \perp$ *for all* i*.*

Proof For any $\mathbf{c} \in \mathcal{C}^w$ such that $\mathsf{Dec}(\mathsf{sk}, c_i) \neq \perp$ for all i, we have $\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}) \in \mathcal{M}^w$ and $f_\perp(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) \neq \perp$. So, (5) becomes an equality. In the other direction, if $\mathsf{Dec}(\mathsf{sk}, c_i) = \perp$ for some i, then $\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}) \notin \mathcal{M}^w$ and $f_\perp(\mathsf{Dec}(\mathsf{sk}, \mathbf{c})) = \perp$. So, (5) is trivially satisfied. □

It is easy to see that any fully composable homomorphic encryption scheme is also homomorphic in the sense of Definition 5.

Theorem 1 *For any set of morphisms* $\mathcal{F}$*, any fully composable* $\mathcal{F}$*-homomorphic encryption scheme (Definition 6) is also* $\mathcal{F}$*-homomorphic (Definition 5).*

Proof Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully composable homomorphic encryption scheme with function set $\mathcal{F}$. Let $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ be any morphism in $\mathcal{F}$, $\mathbf{m} \in \mathcal{M}_\perp^w$, select $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$, and compute $\mathbf{c} = \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m})$. Since $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a valid encryption scheme, we have $\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}) \sqsupseteq \mathbf{m}$. It follows from the full composability property that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) \sqsupseteq f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) \sqsupseteq f(\mathbf{m})$, where in the last step we have used the monotonicity of f. This proves that the scheme satisfies Definition 5. □

In fact, full composability is a strictly stronger notion than the standard homomorphic correctness, i.e., there are $\mathcal{F}$-homomorphic schemes that are not fully composable. We postpone the formal proof of this statement as we will derive it as a corollary of more general result. (See Corollary 2.) Instead, we first analyze the composability properties of Definition 6. There is a fundamental

difference between the two homomorphic correctness definitions: full composability (Definition 6) allows arbitrary composition of functions in $\mathcal{F}$, while $\mathcal{F}$-homomorphism (Definition 5) does not. The composability properties of Definition 6 are easily formulated as a transformation on the set of functions $\mathcal{F}$ supported by the homomorphic encryption scheme.

Definition 7 *For any (typically finite) set of morphisms $\mathcal{F} \subseteq \bigcup_w (\mathcal{M}_\perp^w \to \mathcal{M}_\perp)$, let $\mathcal{F}^{\leq d}$ be the set of all computations $F\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ described by a circuit of depth $\leq d$ with gates in $\mathcal{F}$, and let $\mathcal{F}^* = \bigcup_d \mathcal{F}^{\leq d}$ be the set of computations described by a circuit without any depth restriction.*

The evaluation function Eval *of a $\mathcal{F}$-homomorphic encryption scheme is extended to $F \in \mathcal{F}^*$ in the obvious way, mapping input ciphertexts $\mathbf{c} \in \mathcal{C}^w$ to a final output $\mathsf{Eval}^*(\mathsf{pk}, F, \mathbf{c})$, using $\mathsf{Eval}(\mathsf{pk}, f, \cdots)$ to evalute each f-labeled gate of F. The restriction of Eval^* to computations $F \in \mathcal{F}^{\leq d}$ is denoted $\mathsf{Eval}^{\leq d}$.*

It easily follows by induction that, for any *fully composable* homomorphic encryption scheme, the final output $c = \mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))$ of a homomorphic computation $F \in \mathcal{F}^*$ decrypts to the correct message $\mathsf{Dec}(\mathsf{sk}, c) = F(\mathbf{m})$. This is formalized in the following theorem, showing that the set of functions supported by a fully composable homomorphic encryption scheme can be extended from $\mathcal{F}$ to $\mathcal{F}^*$, i.e., fully composable homomorphic encryption schemes support the evaluation of arbitrary (polynomial size) circuits with gates in $\mathcal{F}$.

Theorem 2 *For any set of functions $\mathcal{F}$, if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is a fully composable $\mathcal{F}$-homomorphic encryption scheme, then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ *is a $\mathcal{F}^*$-homomorphic encryption scheme. Moreover,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ *is fully composable.*

Proof By induction on the depth of F. In the base case, F is a circuit of depth 1 (i.e., a single gate $F \in \mathcal{F}$), and the statement follows from the assumption that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is fully composable $\mathcal{F}$-homomorphic.

For the inductive case, let $F\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ be any circuit of depth $d+1$, and let f be the output gate. Then, we can write $F(\mathbf{m}) = f(F_1(\mathbf{m}), \ldots, F_w(\mathbf{m}))$ for w circuits $F_1, \ldots, F_w$ of depth d. By induction hypothesis, for any $\mathbf{c} \in \mathcal{C}^w$, we have

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(\mathsf{pk}, F_i, \mathbf{c})) \sqsupseteq F_i(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))$$

for all i. It follows from the definition of Eval^* and the assumption that Eval is f-homomorphic that

$$\begin{aligned}
\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(\mathsf{pk}, F, \mathbf{c})) &= \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \{\mathsf{Eval}^*(\mathsf{pk}, F_i, \mathbf{c})\}_i)) \\
&\sqsupseteq f(\{\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(\mathsf{pk}, F_i, \mathbf{c}))\}_i) \\
&\sqsupseteq f(\{F_i(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))\}_i) \\
&= F(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))
\end{aligned}$$

where, as usual, we have used the monotonicity of f. This completes the proof that Eval^* is F-homomorphic. □

Notice that the transformation from Eval to Eval^* preserves the security of the scheme because IND-CPA security only depends on Gen and Enc, which are not modified.

The property established in Theorem 2 is closely related to a (somehow weaker) notion of composability proposed in [GHV10] under the name of *multi-hop* homomorphic encryption. Using our notation, multi-hop homomorphic encryption can be equivalently[9] defined as follows.

Definition 8 (Multi-hop Homomorphic Encryption [GHV10]) *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a homomorphic encryption scheme with message space* $\mathcal{M}$ *and set of functions* $\mathcal{F}$*.* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is a* d*-hop* $\mathcal{F}$*-homomorphic encryption scheme if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ *is* $\mathcal{F}^{\leq d}$*-homomorphic. The scheme is* multi-hop $\mathcal{F}$*-homomorphic if* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ *is* $\mathcal{F}^*$*-homomorphic.*

Notice that the definition of 1-hop homomorphic encryption scheme is the same as $\mathcal{F}$-homomorphic correctness (Definition 5). So, schemes satisfying Definition 5 are also called *single-hop* homomorphic. Moreover, since $\mathcal{F}^{\leq d} \subseteq \mathcal{F}^*$, multi-hop homomorphic schemes are d-hop homomorphic for any d. Finally, it easily follows from Theorem 2 that any fully composable homomorphic encryption scheme is also multi-hop homomorphic.

Corollary 1 *Any fully composable homomorphic encryption scheme is multi-hop homomorphic.*

Proof Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully composable $\mathcal{F}$-homomorphic encryption scheme. By Theorem 2, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is fully composable $\mathcal{F}^*$-homomorphic, and, by Theorem 1, also $\mathcal{F}^*$-homomorphic. So, by definition, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is multi-hop $\mathcal{F}$-homomorphic. □

In summary, both fully composable and multi-hop homomorphic encryption schemes support the homomorphic evaluation of arbitrary circuits with gates in $\mathcal{F}$. But notice the difference between Corollary 1 and Definition 8: in the definition of multi-hop homomorphic encryption, the ability to evaluate any function in $\mathcal{F}^*$ is *assumed*, while for fully composable schemes it is derived from a simpler correctness property (Definition 6) that does not directly involve function composition.

It turns out that all inclusions between d-hop, multi-hop and fully composable homomorphic encryption schemes are strict.

Theorem 3 *Under minimal assumotions*[10] *there are (secure)* d*-hop homomorphic encryption schemes that are not* $(d+1)$*-hop homomorphic.*

Proof Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be d-hop $\mathcal{F}$-homomorphic. Define a new scheme where

$$\begin{aligned}
\mathsf{Enc}'(\mathsf{pk}, m) &= (d, \mathsf{Enc}(\mathsf{pk}, m)) \\
\mathsf{Dec}'(\mathsf{sk}, (l, c)) &= \begin{cases} \mathsf{Dec}(\mathsf{sk}, c) & \text{if } l \geq 0 \\ \bot & \textit{otherwise} \end{cases} \\
\mathsf{Eval}'(\mathsf{pk}, f, \{(l_i, c_i)\}_i) &= (\min_i l_i - 1, \mathsf{Eval}(\mathsf{pk}, f, \{c_i\}_i)).
\end{aligned}$$

The transformation preserves IND-CPA security because the encryption function simply adds a known value d to the ciphertexts. Moreover, it is easy to see that $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Eval}')$ is still d-hop homomorphic, but not $(d+1)$-hop homomorphic. □

[9] Technically, [GHV10] defines multi-hop homomorphic encryption only for unary functions $f\colon \mathcal{M} \to \mathcal{M}$, but the definition is easily adapted to arbitrary $f\colon \mathcal{M}_\bot^w \to \mathcal{M}_\bot$.

[10] Specifically, assuming that (secure) d-hop $\mathcal{F}$-homomorphic encryption schemes exist at all.

Corollary 2 *Under minimal assumptions, for any d, there are d-hop homomorphic encryption schemes that are not fully composable.*

Proof Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a scheme that is d-hop homomorphic, but not $(d+1)$-hop (or multi-hop) homomorphic, as given by Theorem 3. It follows by Corollary 1 that the scheme cannot be fully composable. □

The separation of Corollary 2 can be strengthened showing that even multi-hop encryption schemes may fail to be fully composable.

Theorem 4 *Under minimal assumptions*[11] *for any set of functions $\mathcal{F}$, there are (secure) multi-hop homomorphic encryption schemes that are not fully composable.*

Proof Assume $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is multi-hop homomorphic, and define a new scheme where

$$\begin{aligned}
\mathsf{Enc}'(\mathsf{pk}, m) &= (0, \mathsf{Enc}(\mathsf{pk}, m)) \\
\mathsf{Dec}'(\mathsf{sk}, (l, c)) &= \begin{cases} \mathsf{Dec}(\mathsf{sk}, c) & \text{if } l \leq 1 \\ \bot & \textit{otherwise} \end{cases} \\
\mathsf{Eval}'(\mathsf{pk}, f, \{(l_i, c_i)\}_i) &= (2 \cdot \max_i l_i, \mathsf{Eval}(\mathsf{pk}, f, \{c_i\})).
\end{aligned}$$

It is easy to see that $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Eval}')$ is still multi-hop homomorphic because for all $F \in \mathcal{F}^*$,

$$\begin{aligned}
\mathsf{Dec}'(\mathsf{sk}, (\mathsf{Eval}')^*(\mathsf{pk}, F, (\mathsf{Enc}')^*(\mathsf{pk}, \mathbf{m}))) &= \mathsf{Dec}'(0, \mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) \\
&= \mathsf{Dec}(\mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) \sqsupseteq F(\mathbf{m}).
\end{aligned}$$

Now, let $f \in \mathcal{F}$ and $\mathbf{m}$ be such that $f(\mathbf{m}) \neq \bot$. The ciphertexts $c_i = (1, \mathsf{Enc}(\mathsf{pk}, m_i))$ satisfy

$$\begin{aligned}
f((\mathsf{Dec}')^*(\mathsf{sk}, \mathbf{c})) &= f(\mathsf{Dec}^*(\mathsf{sk}, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) \sqsupseteq f(\mathbf{m}) \neq \bot \\
\mathsf{Dec}'(\mathsf{sk}, \mathsf{Eval}'(\mathsf{pk}, f, \mathbf{c})) &= \mathsf{Dec}'(\mathsf{sk}, (2, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c}))) = \bot.
\end{aligned}$$

So, the scheme is not fully composable. □

So, full composability is a strictly stronger notion than multi-hop homorphic correctness. However, the ciphertexts used in the proof of Theorem 4 are pathological, in the sense that they cannot be produced by repeated application of the encryption and evaluation functions. In fact, this is the only way in which a multi-hop homomorphic encryption scheme may fail to be fully composable. In order to bridge the gap between the two definitions, let's consider a subclass of homomorphic encryption schemes that do not contain such useless ciphertexts.

Definition 9 *A homomorphic encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *(with message space $\mathcal{M}$ and functions $\mathcal{F}$) is* surjective *if for any key $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$, and ciphertext $c \in \mathcal{C}$, there is a function $F \in \mathcal{F}^*$ and message vector $\mathbf{m} \in \mathcal{M}_\bot^w$ such that $F(\mathbf{m}) \neq \bot$ and*

$$\Pr\{\mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m})) = c\} > 0$$

i.e., the ciphertext c can be obtained as the result of a valid homomorphic computation with nonzero probability.

[11] Specifically, assuming that (secure) multi-hop $\mathcal{F}$-homomorphic encryption schemes exist at all, and $\mathcal{F}$ contains at least one non-trivial function $f\colon \mathcal{M}_\bot^w \to \mathcal{M}_\bot$ such that $f(\mathbf{m}) \neq \bot$ for some $\mathbf{m}$.

Theorem 5 *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a surjective multi-hop homomorphic encryption scheme. Then* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is fully composable.*

Proof Let $f : \mathcal{M}_\perp^w \to \mathcal{M}_\perp$ be any function in $\mathcal{F}$, and $\mathbf{c} \in \mathcal{C}^w$ a vector of ciphertexts. We need to prove that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) \sqsupseteq f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))$. Since the scheme is surjective, for any c_i there are $F_i \in \mathcal{F}^*$ and $\mathbf{m}_i \in \mathcal{M}^*$ such that $F_i(\mathbf{m}_i) \neq \perp$ and $\mathsf{Eval}(\mathsf{pk}, F_i, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}_i)) = c_i$ with nonzero probability. If follows from the multi-hop homomorphic property that $\mathsf{Dec}(\mathsf{sk}, c_i) \sqsupseteq F_i(\mathbf{m}_i) \neq \perp$, and, therefore $\mathsf{Dec}(\mathsf{sk}, c_i) = F_i(\mathbf{m}_i)$. Now, consider the function

$$F(\mathbf{m}_1, \ldots, \mathbf{m}_w) = f(F_1(\mathbf{m}_1), \ldots, F_w(\mathbf{m}_w)) \in \mathcal{F}^*.$$

Since the encryption scheme is d-hop homomorphic, we have

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}_1, \ldots, \mathbf{m}_w))) &\sqsupseteq F(\mathbf{m}_1, \ldots, \mathbf{m}_w) \\ &= f(F_1(\mathbf{m}_1), \ldots, F_w(\mathbf{m}_w)) \\ &= f(\mathsf{Dec}(\mathsf{sk}, c_1), \ldots, \mathsf{Dec}(\mathsf{sk}, c_w)) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})). \end{aligned}$$

with probability 1. But, by definition of Eval^* and F, we also have

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}_1, \ldots, \mathbf{m}_w))) &= \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \{\mathsf{Eval}^*(\mathsf{pk}, F_i, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}_i))\})) \\ &= \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) \end{aligned}$$

with nonzero probability. Therefore it must be that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(f, \mathbf{c})) \sqsupseteq f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))$. □

4 Bootstrapping

The following theorem is in essence a formalization of Gentry's bootstrapping technique [Gen09b] presented in terms of our full composability definition. Instead of directly showing that a bootstrapped scheme supports the evaluation of arbitrary circuits, we show that it is *fully composable*. The ability to evaluate arbitrary circuits homomorphically then follows from Theorem 2.

We begin by describing the bootstrapping construction of [Gen09b], genelarized to morphisms $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$.

Definition 10 (Bootstrapping) *Fix a set* $\mathcal{F}$ *of functions* $f\colon \mathcal{M}_\perp^w \to \mathcal{M}_\perp$*, and an (injective) encoding* $\psi\colon \mathcal{K} \to \mathcal{M}^k$*. Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a* $\mathcal{F}_\psi^{\circlearrowleft}$*-homomorphic encryption scheme with message space* $\mathcal{M}$*, ciphertext space* $\mathcal{C}$*, and secret key space* $\mathcal{K}$*, where* $\mathcal{F}_\psi^{\circlearrowleft}$ *is the set of all functions* $f_{\mathbf{c}}^{\circlearrowleft}\colon \mathcal{M}_\perp^k \to \mathcal{M}_\perp$ *indexed by* $f \in \mathcal{F}$ *and* $\mathbf{c} \in \mathcal{C}^w$ *defined as*

$$f_{\mathbf{c}}^{\circlearrowleft}(\mathbf{x}) = \begin{cases} f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) & \text{if } \mathbf{x} = \psi(\mathsf{sk}) \text{ for some } \mathsf{sk} \in \mathcal{K} \\ \perp & \text{otherwise.} \end{cases} \tag{7}$$

The bootstrapped encryption scheme $\mathsf{FHE}^{\circlearrowleft} \stackrel{def}{=} (\mathsf{Gen}^\psi, \mathsf{Enc}^\psi, \mathsf{Dec}, \mathsf{Eval}^{\circlearrowleft})$ *consists of the following algorithms:*

- $\mathsf{Gen}^\psi, \mathsf{Enc}^\psi$ *are the key generation and encryption algorithms from Definition 3,*

- *the decryption algorithm* Dec *is the same as that of* FHE*, and*
- *the evaluation function is* $\mathsf{Eval}^{\circlearrowleft}((\mathsf{pk},\mathsf{pk}'),f,\mathbf{c}) = \mathsf{Eval}(\mathsf{pk}, f^{\circlearrowleft}_{\mathbf{c}}, \mathsf{pk}')$.

The following theorem shows that bootstrapping produces a fully composable homomorphic encryption scheme.

Theorem 6 *For any set* $\mathcal{F}$ *of morphisms* $f\colon \mathcal{M}_{\perp}^{w} \to \mathcal{M}_{\perp}$ *and encoding* $\psi\colon \mathcal{K} \to \mathcal{M}^k$*, let* $\mathsf{FHE} = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ *be a valid* $\mathcal{F}^{\circlearrowleft}_{\psi}$*-homomorphic encryption scheme with secret key space* $\mathcal{K}$*, message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$*. Then, the bootstrapped scheme* $\mathsf{FHE}^{\circlearrowleft}$ *from Definition 10 is valid,* $\mathcal{F}$*-homomorphic, and* fully composable*. Moreover, if* FHE *is* ψ*-circular IND-CPA secure, then* $\mathsf{FHE}^{\circlearrowleft}$ *is also (*ψ*-circular) IND-CPA secure.*

Proof The IND-CPA security of $\mathsf{FHE}^{\circlearrowleft}$ immediately follows from the assumption that FHE is ψ-circular IND-CPA secure, and Definition 3. Moreover, appending $\mathsf{pk}' = \mathsf{Enc}^{\psi *}(\mathsf{pk},\mathsf{sk})$ to the public key $(\mathsf{pk},\mathsf{pk}')$ does not add any new information. So, $\mathsf{FHE}^{\circlearrowleft}$ is ψ-circular IND-CPA security. We also have

$$\mathsf{Dec}(\mathsf{sk},\mathsf{Enc}^{\psi}((\mathsf{pk},\mathsf{pk}'),m)) = \mathsf{Dec}(\mathsf{sk},\mathsf{Enc}(\mathsf{pk},m)) = m$$

for all $m \in \mathcal{M}$. So, $\mathsf{FHE}^{\circlearrowleft}$ is a valid encryption scheme.

In order to prove that $\mathsf{FHE}^{\circlearrowleft}$ is fully composable, let $f\colon \mathcal{M}_{\perp}^{w} \to \mathcal{M}_{\perp}$ be any morphism in $\mathcal{F}$, and $\mathbf{c} \in \mathcal{C}^w$ a sequence of ciphertexts. Then, since FHE is $\mathcal{F}^{\circlearrowleft}_{\psi}$-homomorphic, we have

$$\begin{aligned}
\mathsf{Dec}(\mathsf{sk},\mathsf{Eval}^{\circlearrowleft}((\mathsf{pk},\mathsf{pk}'),f,\mathbf{c})) &= \mathsf{Dec}(\mathsf{sk},\mathsf{Eval}(\mathsf{pk},f^{\circlearrowleft}_{\mathbf{c}},\mathsf{pk}')) \\
&\sqsupseteq f^{\circlearrowleft}_{\mathbf{c}}(\mathsf{Dec}^{*}(\mathsf{sk},\mathsf{pk}')) \\
&= f^{\circlearrowleft}_{\mathbf{c}}(\mathsf{Dec}^{*}(\mathsf{sk},\mathsf{Enc}^{*}(\mathsf{pk},\psi(\mathsf{sk})))) \\
&= f^{\circlearrowleft}_{\mathbf{c}}(\psi(\mathsf{sk})) = f(\mathsf{Dec}^{*}(\mathsf{sk},\mathbf{c})).
\end{aligned}$$

This proves the full composability property. □

Notice that Definition 10 and Theorem 6 transform a (non-composable) scheme FHE that supports only the evaluation of morphismss $f^{\circlearrowleft}_{\mathbf{c}}(\mathbf{x})$ with a *fixed* number of inputs $\mathbf{x} \in \mathcal{M}^k$ (determined by the encoding function ψ), to a scheme $\mathsf{FHE}^{\circlearrowleft}$ that supports the (composable) evaluation of morphisms f with any number of inputs w. However, the larger is the set of morphisms $\mathcal{F}$ we want $\mathsf{FHE}^{\circlearrowleft}$ to support, the larger is the set $\mathcal{F}^{\circlearrowleft}_{\psi}$ for which FHE is required to be $\mathcal{F}^{\circlearrowleft}_{\psi}$-homomorphic. In practice, $\mathcal{F}$ is usually a small (finite) set of morphisms, with a fixed number of inputs. For example, for boolean messages $\mathcal{M} = \{0,1\}$, one may use a set $\mathcal{F} = \{f_{\text{NAND}}\}$ consisting of a single function $f_{\text{NAND}}\colon \mathcal{M}^2 \to \mathcal{M}$ implementing the NAND gate $f_{\text{NAND}}(x_0,x_1) = 1 - x_0 \cdot x_1$, which is universal for boolean computations. Then, using the fact that $\mathsf{FHE}^{\circlearrowleft}$ is fully composable, and Theorem 2, conclude that $\mathsf{FHE}^{\circlearrowleft *}$ (i.e., the same scheme with evaluation function $\mathsf{Eval}^{\circlearrowleft *}$ extended to circuits with gates in $\mathcal{F}$) is $\mathcal{F}^*$-homomorphic, i.e., it supports the homomorphic evaluation of arbitrary boolean functions $F\colon \mathcal{M}^w \to \mathcal{M}$, expressed as boolean circuits.

Corollary 3 *Let* $\mathsf{FHE} = (\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ *be* ψ*-circular IND-CPA secure* $\mathcal{F}^{\circlearrowleft}_{\psi}$*-homomorphic encryption scheme. Then* $\mathsf{FHE}^{\circlearrowleft *}$ *is a valid,* ψ*-circular secure, fully composable* $\mathcal{F}^*$*-homomorphic encryption scheme.*

The bootstrapping theorem, as stated above, requires the starting scheme FHE to be circular secure. If FHE is only IND-CPA secure, we can still achieve a limited form of composition using leveled bootstrapping.

Definition 11 (Leveled Bootstrapping) *Let* $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*,* $\psi \colon \mathcal{K} \to \mathcal{M}^k$*,* $\mathcal{F}$ *and* $\mathcal{F}_\psi^{\circlearrowleft}$ *be as in Theorem 6. The Leveled homomorphic encryption scheme* $\mathsf{FHE}^\sharp = (\mathsf{Gen}_d^\sharp, \mathsf{Enc}^\sharp, \mathsf{Dec}^\sharp, \mathsf{Eval}^\sharp)$ *is defined by the following algorithms*

- $\mathsf{Gen}_d^\sharp(\kappa)$ *runs* $\mathsf{pk}_i \in (\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{Gen}(\kappa)$ *for* $i = 0, \ldots, d$*, computes* $\mathsf{pk}'_i \leftarrow \mathsf{Enc}^*(\mathsf{pk}_i, \psi(\mathsf{sk}_{i-1})))$ *for* $i = 1, \ldots, d$*, and outputs* $(\{\mathsf{sk}_i\}_{i\geq 0}, (\{\mathsf{pk}_i\}_{i\geq 0}, \{\mathsf{pk}'_i\}_{i\geq 1}))$.
- $\mathsf{Enc}^\sharp((\{\mathsf{pk}_i\}_i, \ldots), m) = (0, \mathsf{Enc}(\mathsf{pk}_0, m))$
- $\mathsf{Dec}^\sharp(\{\mathsf{sk}_i\}_{i\geq 0}, (\ell, c)) = \mathsf{Dec}(\mathsf{sk}_\ell, c)$
- $\mathsf{Eval}^\sharp((\{\mathsf{pk}_i\}_i, \{\mathsf{pk}'_i\}_i), f, \hat{\mathbf{c}})$ *checks that* $\hat{c}_i = (\ell, c_i)$ *for all* i *and some (common) value* ℓ*, and outputs* $(\ell + 1, \mathsf{Eval}(\mathsf{pk}_{\ell+1}, f^{\circlearrowleft}_{\mathbf{c}}, \mathsf{pk}'_{\ell+1}))$*. Otherwise,* $\mathsf{Eval}^\sharp$ *outputs* $\perp$.

Theorem 7 *If* $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is* $\mathcal{F}_\psi^{\circlearrowleft}$*-homomorphic, then* $\mathsf{FHE}^\sharp$ *is a leveled homomorphic encryption scheme with functions* $\mathcal{F}_d^\sharp$*, the set of layered*[12] *circuits of depth bounded by* d*. Moreover, if* FHE *is IND-CPA secure, then* $\mathsf{FHE}^\sharp$ *is also IND-CPA secure.*

Proof The proof of homomorphic correctness is similar to the proof of full composability of Theorem 6 and Corollary 1. Security is proved by a standard hybrid argument. □

References

[BPMW16] Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. FHE circuit privacy almost for free. In Robshaw and Katz [RK16], pages 62–89.

[Bra19] Zvika Brakerski. Fundamentals of fully homomorphic encryption. In Oded Goldreich, editor, *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 543–563. ACM, 2019.

[DD22] Nico Döttling and Jesko Dujmovic. Maliciously circuit-private FHE from information-theoretic principles. In Dana Dachman-Soled, editor, *3rd Conference on Information-Theoretic Cryptography, ITC 2022, July 5-7, 2022, Cambridge, MA, USA*, volume 230 of *LIPIcs*, pages 4:1–4:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[DS16] Léo Ducas and Damien Stehlé. Sanitization of FHE ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 294–310. Springer, 2016.

[12] A circuit $F(\mathbf{x})$ is layered if it is either a single gate (when $d = 1$), or equals $F(\mathbf{x}) = f(F_1(\mathbf{x}), \ldots, F_w(\mathbf{x}))$ for some gate $f \colon \mathcal{M}^w \to \mathcal{M}$ and layered circuits $F_1, \ldots, F_w \in \mathcal{F}_{d-1}^\sharp$. Any circuit $F \in \mathcal{F}^{\leq d}$ can be efficiently transformed into a layered one of the same depth, assuming $\mathcal{F}$ contains the identity function $f(x) = x$.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, USA, 2009.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. i-hop homomorphic encryption and rerandomizable yao circuits. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 155–172. Springer, 2010.

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

[Hal17] Shai Halevi. Homomorphic encryption. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 219–276. Springer International Publishing, 2017.

[OPP14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2014.

[RK16] Matthew Robshaw and Jonathan Katz, editors. *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*. Springer, 2016.