IACR Communications in Cryptology
ISSN 3006-5496, Vol. 2, No. 1, 27 pages.

https://doi.org/10.62056/ak5wl86bm



# Fully Composable Homomorphic Encryption

Daniele Micciancio

UC San Diego, Computer Science and Engineering, La Jolla, USA

**Abstract.** The traditional definition of fully homomorphic encryption (FHE) is not composable, i.e., it does not guarantee that evaluating two (or more) homomorphic computations in a sequence produces correct results. We formally define and investigate a stronger notion of homomorphic encryption which we call "fully composable homomorphic encryption", or "composable FHE". The definition is both simple and powerful: it does not directly involve the evaluation of multiple functions, and yet it supports the arbitrary composition of homomorphic evaluations. On the technical side, we compare the new definition with other definitions proposed in the past, proving both implications and separations, and show how the "bootstrapping" technique of (Gentry, STOC 2009) can be formalized as a method to transform a (non-composable, circular secure) homomorphic encryption scheme into a fully composable one. We use this formalization of bootstrapping to formulate a number of conjectures and open problems.

**Keywords:** Fully homomorphic encryption · composability · circular security · functional bootstrapping

## 1 Introduction

A fully homomorphic encryption scheme is a cryptosystem that allows to perform arbitrary computations on encrypted data. More specifically, an encryption scheme with message space $\mathcal{M}$ is $\mathcal{F}$-*homomorphic* (for some set of functions[1] $\mathcal{F}$) if for any $f \in \mathcal{F}$ and input value $m \in \mathcal{M}$, given an encryption $c = \mathsf{Enc}(m)$ of that value, one can compute (publicly, without knowledge of the decryption key) a ciphertext $c' = \mathsf{Eval}(f, c)$ that decrypts to $f(m)$. (See Figure 1 for an illustration.) 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 (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 surveys [Hal17, Bra19].) While this definition closely models the intended use of homomorphic encryption schemes in typical applications, it has a shortcoming: homomorphic computations cannot be composed together, i.e., the result of computing $c = \mathsf{Enc}(m)$, $c' = \mathsf{Eval}(f, c)$ and then $c'' = \mathsf{Eval}(g, c')$ (for some $m \in \mathcal{M}$ and $f, g \colon \mathcal{M} \to \mathcal{M}$) is not guaranteed to produce a ciphertext $c''$ that decrypts to $g(f(m))$.

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], who observed that the ciphertexts accepted as input and produced as output by the evaluation function $\mathsf{Eval}_f$ can, in general, be very different. So, the output of $\mathsf{Eval}_f$ may not even be a syntactically valid input to $\mathsf{Eval}_g$. In order to address the composability problem, [GHV10] proposed a stronger notion of correctness,

Work supported in part by Intel Crypto Frontiers program.

E-mail: daniele@cs.ucsd.edu (Daniele Micciancio)

[1] For simplicity, in this introduction we focus on functions $f \colon \mathcal{M} \to \mathcal{M}$ of a single input. This is generalized to multi-input functions $f \colon \mathcal{M}^w \to \mathcal{M}$ in the rest of the paper.



Received: 2024-10-03 Accepted: 2025-03-11

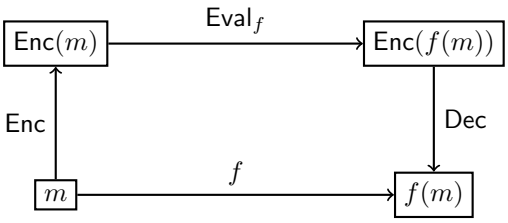


**Figure 1:** The standard correctness definition for homomorphic encryption. Encrypting a message $m$ to obtain a ciphertext $c = \mathsf{Enc}(m)$, performing a homomorphic computation $c' = \mathsf{Eval}(f, c)$, and then decrypting the final result $\mathsf{Dec}(c') = f(m)$ produces the same output as evaluating the function $f$ on the unencrypted message $m$.

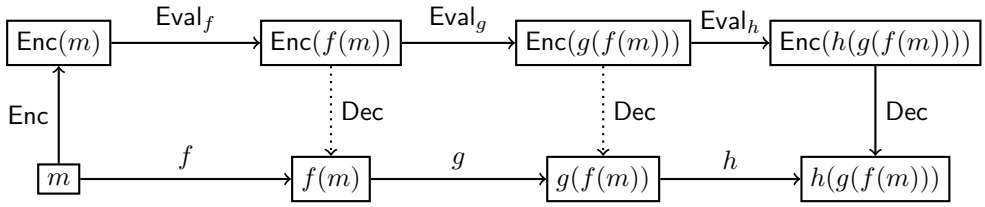


**Figure 2:** A 3-hop homomorphic encryption scheme supports the consecutive homomorphic evaluation of 3 functions, $f, g, h$. More generally, a $i$-hop encryption scheme allows to chain $i$ homomorphic computations.

called *$i$-hop homomorphic encryption*. In an $i$-hop encryption scheme [GHV10], one can sequentially evaluate up to $i$ functions[2] homomorphically on a ciphertext $c = \mathsf{Enc}(m)$, and the final result $c' = \mathsf{Eval}_{f_i}(\mathsf{Eval}_{f_{i-1}}(\cdots \mathsf{Eval}_{f_1}(c)))$ will be a ciphertext that decrypts to $f_i(f_{i-1}(\cdots f_1(m)))$. (See Figure 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 is $i$-hop homomorphic for all integers $i$, then it is called *multi-hop*.

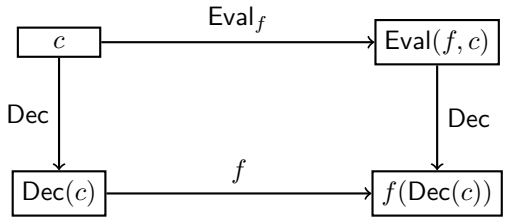


**Figure 3:** Fully composable homomorphic encryption. For any ciphertext $c$, applying the decryption function $\mathsf{Dec}(c)$ and then evaluating a function $f$ on the result produces the same output as first evaluating $f$ homomorphically on $c$, and then decrypting $\mathsf{Eval}(f, c)$.

**Contributions** The multi-hop property, while desirable, is somehow hard to check, as it requires considering (the homomorphic evaluation of) arbitrary sequences of functions $f_1, f_2, \ldots$. In this paper we investigate a different approach to achieve composability,

[2] As in [GHV10], for simplicity, in this introduction we only consider unary functions $\mathcal{F} \subseteq \mathcal{M} \to \mathcal{M}$, evaluated in sequence. In the main body of the paper we generalize unary functions to functions $f\colon \mathcal{M}^w \to \mathcal{M}$ with any number of arguments, combined into directed acyclic graphs.

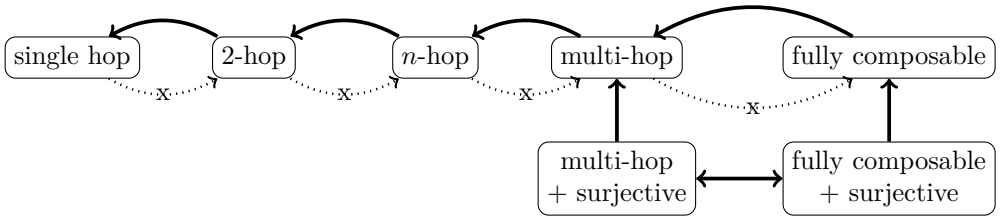


**Figure 4:** Relations between homomorphic encryption variants. "Single hop" or "1-hop" is the basic standard notion of homomorphic encryption. For any integers $n < m$, any $m$-hop homomorphic encryption scheme is also $n$-hop, but there are $n$-hop schemes that are not $m$-hop (Theorem 3). Fully composable schemes are multi-hop, i.e. $n$-hop for any $n$ (Theorem 2), but there are multi-hop schemes that are not fully composable (Theorem 4). When restricted to surjective schemes (Definition 9) with perfect correctness, the multi-hop and fully composable properties are equivalent (Corollary 1 and Theorem 5).

which we call *fully composable* homomorphic encryption (Definition 6). Technically, we say that a scheme is "fully composable" if the homomorphic evaluation function $\mathsf{Eval}_f$ commutes with the decryption function $\mathsf{Dec}$: evaluating a function $f$ homomorphically on a ciphertext $c$ and then decrypting $\mathsf{Eval}_f(c)$ should produce the same result as first decrypting $m = \mathsf{Dec}(c)$, and then computing $f(m)$ in the clear. (See Figure 3.)

Crucially, this is required for all ciphertexts $c$, not only those produced by the encryption function $\mathsf{Enc}(m)$. Notice that, syntactically, this definition is as simple as the standard notion of homomorphic encryption, involving the evaluation of a single function $f$. (See Figures 1 and 3 for a pictorial comparison of the two definitions.) Still, it achieves very strong composition properties.

In this paper we formally investigate this notion of full composability and its relation to previous 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 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 full 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. (See Theorem 3, Theorem 4, and Figure 4 for a pictorial summary of this and the next bullet.)

3. For a general class of homomorphic encryption schemes (satisfying a natural surjectivity property, see Definition 9) multi-hop correctness[3] 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.)

In addition to the above, in Section 5 we consider a number of extensions and optimizations. In particular,

[3]For technical reasons related to the definition of surjective encryption, this result required perfect correctness, i.e., zero probability of decryption errors in the multi-hop correctness definition. See discussion after Definition 9 for details.

- we propose a definition of "wire-bootstrappable" encryption scheme (Definition 12) which more closely corresponds to the way Gentry's bootstrapping technique is used in practice, and show that it is equivalent to full composability (Theorems 9 and 10), and

- we extend this notion to "functional bootstrapping", a more powerful operation which is at the core of FHEW-like homomorphic encryption schemes [DM15, CGGI20, MP21, LMK+23].

We emphasize that our contributions are mostly definitional, not algorithmic: we show how known algorithmic techniques commonly used in practice to speed up homomorphic encryption can be formulated in terms of correctness and composition properties, in the style of our full composability definition. Still, we think that our definitions can be useful to frame and further study these and other optimization techniques.

As a final remark, we note that Gentry's bootstrapping technique [Gen09b] is usually described (and understood) as a "noise reduction" mechanism in lattice-based cryptography. It is a somehow peculiar property of encryption schemes based on lattice problems that ciphertexts are "noisy", with higher levels of noise corresponding to lower quality ciphertexts. The noise grows during homomorphic computation, and if it surpasses a certain threshold then the ciphertext becomes undecryptable. So, in order to keep computing homomorphically on encrypted data one needs to periodically apply bootstrapping to bring the noise back to acceptable levels. As essentially all known fully homomorphic encryption constructions are based on lattices, this has often led to the question of whether "noisy ciphertexts" are somehow necessary to perform arbitrary computations on encrypted data. Our full composability definition provides a different perspective on bootstrapping, making no explicit mention of ciphertext noise. It describes the problem solved by bootstrapping in abstract terms, as a general transformation between encryption schemes achieving different notions of correctness, not specific to lattice-based cryptography. This allows to formulate interesting questions/conjectures about the role of bootstrapping in the construction of fully homomorphic encryption. For example, one may ask if the existence of fully homomorphic encryption schemes (supporting arbitrary computations on encrypted data) implies the existence of schemes that are fully composable, or if there are methods to achieve full composability other than bootstrapping.

**Related Work** As already mentioned in the introduction, the problem of composability of homomorphic computations was first explicitly posed in [GHV10], and our transformation from non-composable to composable FHE is essentially a formalization of the bootstrapping technique from [Gen09b]. Our definition of bootstrappable encryption scheme (Definition 12) differs from the one originally given in [Gen09b] and more closely corresponds to how bootstrapping is implemented in practice. To disambiguate between the two definitions we refer to the definition of [Gen09b] as "gate-bootstrappable", and our new definition as "wire-bootstrappable".

Our definition of full composability uses a universal quantification over input ciphertexts. (See Definition 6 and the discussion immediately after it for a motivating explanation.) A similar universal quantification over input ciphertexts was previously used in [AV21, AGHV25] in their definition of *circuit-privacy*$^+$, though for very different reasons. (The goal in [AV21, AGHV25] was to capture active attacks by a malicious server in the context of FHE-based client-server protocols.[4]) Still, there is a close technical relation between the two definitions. Informally, a homomorphic encryption scheme is circuit-private$^+$ if, for any function $f$ and ciphertext $c$, the result $\mathsf{Eval}_f(c)$ of a homomorphic computation is

[4]The reader is referred to [AGHV25, Section 1, *"Prior versions of this work"*] for a description of that line of research, and a detailed account of the contributions and timeline of the papers that lead to [AGHV25] and its conference version presented at TCC 2022.

statistically close to a fresh encryption $\mathsf{Enc}(f(\mathsf{Dec}(c)))$ of the final result of the computation. (See [AGHV25, Definition 12] for a formal definition.) Then, applying the decryption algorithm to both distributions gives $\mathsf{Dec}(\mathsf{Eval}_f(c)) = \mathsf{Dec}(\mathsf{Enc}(f(\mathsf{Dec}(c)))) = f(\mathsf{Dec}(c))$, i.e., decryption commutes with the evaluation of $f$. So, circuit-privacy$^+$ implies full composability. [AV21, AGHV25] also suggests that a homomorphic encryption scheme can be made circuit-private$^+$ (and therefore fully composable) using a sanitization algorithm [DS16], i.e., a randomized algorithm $\mathsf{San}$ mapping ciphertexts to ciphertexts that preserves the encrypted message $\mathsf{Dec}(\mathsf{San}(c)) = \mathsf{Dec}(c)$ and that outputs (almost) the same distribution $\mathsf{San}(c) \approx \mathsf{San}(c')$ on any two valid ciphertexts that decrypt to the same message $\mathsf{Dec}(c) = \mathsf{Dec}(c')$. In fact, if an encryption scheme satisfies the homomorphic correctness property $\mathsf{Dec}(\mathsf{Eval}_f(\mathsf{Enc}(m))) = f(m) = \mathsf{Dec}(\mathsf{Enc}(f(m)))$, then one can replace $\mathsf{Dec}$ with $\mathsf{San}$ in these expressions and obtain $\mathsf{San}(\mathsf{Eval}_f(\mathsf{Enc}(m))) \approx \mathsf{San}(\mathsf{Enc}(f(m)))$. So, it is very tempting to conclude that using modified encryption and evaluation algorithms $\mathsf{Enc}^{santz}(m) = \mathsf{San}(\mathsf{Enc}(m))$ and $\mathsf{Eval}_f^{santz}(c) = \mathsf{San}(\mathsf{Eval}_f(\mathsf{San}(c)))$ (as suggested in [AGHV25, Definition 8]) gives a homomorphic encryption scheme that satisfies circuit-privacy$^+$. The problem with this transformation is that the modified scheme is not, in general, homomorphically correct, and in Section 3 (Lemma 2) we give a concrete counterexample to show that. In fact, this problem is implicitly acknowledged and partially addressed in [AGHV25, Lemma 2] by *assuming* (rather than *proving*) that the modified scheme is homomorphically correct.[5] This may be a reasonable assumption *in practice* as many concrete homomorphic encryption schemes based on lattices have this property. However, it is not clear how to generically achieve circuit-privacy$^+$ (and full composability) using a sanitization algorithm.

The circular security assumption underlying the bootstrapping technique (and our construction in Section 4) has been extensively studied in a long sequence of previous works (e.g., see [BHHO08, ACPS09, BG10, BGK11, KRW15, KW16, AP16, GKW17b, GKW17a, HK17, KM20, MV24],) but is somehow orthogonal to the main concerns of this paper.

A scheme offering full composability (and a form of functional bootstrapping) as a core functionality was first presented in [DM15]. The composability advantages (and the power of functional bootstrapping) are clearly highlighted in that paper, though without putting forward a formal definition of full composability and then using it to prove an abstract composition theorem. Still, [DM15] served as a starting point for our definitional work, some of which was presented in preliminary form in a number of talks given by the author (e.g., see [Mic22a, Mic22b]) where the notion of composable FHE is explicitly formulated.

**Paper Outline** The rest of the paper is organized as follows. In Section 2 we recall previous (non-composable) definitions of homomorphic encryption schemes. In Section 3 we present our definition of fully composable homomorphic encryption (composable FHE), and study its relation to other definitions. In Section 4 we show how to formulate Gentry's bootstrapping technique in terms of full composability. Finally, in Section 5 we describe extensions of our definition to wire-bootstrappable schemes, and functional bootstrapping. Section 6 concludes with a discussion of open problems and directions for further research.

[5] The preliminary version of the paper [AV21, Lemma 1] contains no such assumption, but it appears that the proof contains a bug and the claim made in [AV21, Lemma 1] is incorrect, as demonstrated by our counterexample in Section 3, Lemma 2. We remark that [AGHV25, Lemma 2] (as well as the TCC 2022 conference version) states that the modified scheme is homomorphically correct as an *assumption*, making our counterexample inapplicable. So, [AGHV25, Lemma 2] is technically correct and contains no bug.

## 2 Definitions

In this section we recall the standard notion of (homomorphic) encryption scheme and (circular) security against chosen plaintext attacks. Cryptographic constructions typically make use of a security parameter $\kappa$, and their properties hold only with high probability, as a function of $\kappa$. We say that a probability $p(\kappa)$ is negligible is $p(\kappa) < 1/\kappa^c$ for all constant $c > 0$ and sufficiently large $\kappa$. A probability $p(\kappa)$ is overwhelming if $1 - p(\kappa)$ is negligible.

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

- The randomized *Key Generation* algorithm $\mathsf{Gen}$, on input a security parameter $\kappa$, outputs a pair of (secret and public) keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$.
- The randomized *Encryption* algorithm $\mathsf{Enc}$ on input key $\mathsf{pk}$ and message $m \in \mathcal{M}$, outputs a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ in $\mathcal{C}$.
- The deterministic[6] *Decryption* algorithm $\mathsf{Dec}(\mathsf{sk}, c)$, on input a secret key $\mathsf{sk}$ and ciphertext $c \in \mathcal{C}$, 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 $m \in \mathcal{M}$, with overwhelming probability over the choice of the keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$ and the randomness of $\mathsf{Enc}$.

We allow for a small (negligible) probability of decryption errors for generality, and because this is not uncommon in practical lattice-based (fully homomorphic) encryption schemes. But this does not play any significant role in our results. So, for simplicity the reader may assume that the schemes satisfy *perfect correctness*, i.e., (1) holds with probability 1, and similarly for other correctness definitions in the paper.

As commonly done both in theory and in practice, in the definition of correctness we quantified universally over the message $m$. This could be relaxed using a game-based definition of correctness where $m$ is chosen by an adversary as a function of the public key $\mathsf{pk}$. But, in any case, the probability of decryption failures should always be assumed to be negligible, as it is well known that decryption errors can easily lead to a complete loss of security. (E.g., see [HNP+03].)

We remark that 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. In particular, since $\perp \notin \mathcal{M}$, if $\mathsf{Dec}$ outputs $\perp$, then (1) is not satisfied, and in an encryption scheme with perfect correctness $\mathsf{Dec}$ should never output $\perp$ when given a properly computed ciphertext.

We focus on encryption schemes with a finite, fixed-length message space,[7] 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 component-wise:

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

[6] Without loss of generality, decryption can always be assumed to be deterministic, e.g., by emulating its randomness using a pseudo-random function with a key stored as part of $\mathsf{sk}$.

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

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.[8]
3. The adversary is given a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$ and outputs a value $b' \leftarrow \mathcal{A}(c)$ in $\{0, 1, \perp\}$.

The advantage of the adversary in the attack is defined as[9] $\delta = (\beta - \bar{\beta})^2/(\beta + \bar{\beta})$ where $\beta = \Pr\{b' = b\}$ and $\bar{\beta} = \Pr\{b' = 1 - b\}$.

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 extension $(\mathsf{Gen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ 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 $b = 1$).[10] 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*[11], 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. Following [MV24],

[8] 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 $k$.

[9] This is the definition of advantage given in [MW18] to capture the concrete bit-security level of a cryptographic primitive, and makes essential use of adversaries that may output a special symbol $\perp$ to express low confidence in their decision. For adversaries that always output a bit $b' \in \{0,1\}$, we have $\beta + \bar{\beta} = 1$, and $\delta$ equals (the square of) the distinguishing gap $\beta - \bar{\beta}$, as used in the traditional (asymptotic) treatment of security. Since this is a theoretical paper, the reader not familiar with the concrete bit-security notion of [MW18] can ignore the distinction between these two definitions.

[10] More specifically, we assume that, for any fixed value of the security parameter $\kappa$, 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 $\mathsf{pk}$ such that membership in $\mathcal{C}$ can be efficiently tested and the uniform (or other standard) distribution on $\mathcal{C}$ can be efficiently sampled.

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

circular security of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ can be formally 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 $\mathsf{Gen}^{\psi}$ 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 (possibly randomized) key encoding function $\psi\colon \mathcal{K} \to \mathcal{M}^w$, a public-key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $\psi$-circular IND-CPA secure if the scheme $(\mathsf{Gen}^{\psi}, \mathsf{Enc}^{\psi}, \mathsf{Dec})$ with modified key generation and encryption algorithms 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.

## 2.2 Homomorphic Encryption

A homomorphic encryption scheme allows to perform computations on encrypted data using a publicly computable *evaluation* algorithm $\mathsf{Eval}$.

**Definition 4** (Homomorphic encryption scheme, syntax)**.** A homomorphic encryption scheme with message and ciphertext spaces $\mathcal{M}, \mathcal{C}$ and functions $\mathcal{F} \subseteq \bigcup_{w \geq 0} \{f\colon \mathcal{M}^w \to \mathcal{M}\}$ is a encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the same message and ciphertext spaces $\mathcal{M}, \mathcal{C}$ together with a (possibly randomized) *evaluation* algorithm $\mathsf{Eval}$ that on input a public key $\mathsf{pk}$, a function $f\colon \mathcal{M}^w \to \mathcal{M}$ in $\mathcal{F}$, and a sequence $\mathbf{c} \in \mathcal{C}^w$, outputs a ciphertext $\mathsf{Eval}(\mathsf{pk}, f, \mathbf{c}) \in \mathcal{C}$.

Note that in Definition 4, the word "homomorphic" refers only to the syntax of the scheme (e.g., the existence of an evaluation algorithm $\mathsf{Eval}$,) without yet imposing any correctness requirement on it. In fact, we will consider several notions of homomorphic correctness, and further qualify the term "homomorphic encryption" to specify which notion of correctness is satisfied. For emphasis, when no notion of correctness is assumed, we may refer to $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ as a "syntactically homomorphic" encryption scheme.

The standard definition of correctness for homomorphic encryption schemes requires that for any function $f\colon \mathcal{M}^w \to \mathcal{M}$, the following holds true: 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. (See Figure 1.)

**Definition 5** (Homomorphic correctness)**.** A (syntactically) homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ (with messages, ciphertext and functions $\mathcal{M}, \mathcal{C}, \mathcal{F}$) satisfies *homomorphic correctness* if for any function $f\colon \mathcal{M}^w \to \mathcal{M}$ in $\mathcal{F}$, messages $\mathbf{m} \in \mathcal{M}^w$ and random keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$, we have

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) = f(\mathbf{m}) \tag{2}$$

with overwhelming probability over the randomness of $\mathsf{Gen}$, $\mathsf{Enc}$ and $\mathsf{Eval}$.

For brevity, we say that a scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is $\mathcal{F}$-homomorphic if it has $\mathcal{F}$ as its set of functions (as specified in Definition 4) and it satisfies the basic notion of homomorphic correctness given in Definition 5. (Stronger, alternative correctness notions will be defined in the next section.)

We remark that in order to run the evaluation algorithm $\mathsf{Eval}$ on $f \in \mathcal{F}$, one needs to provide $\mathsf{Eval}$ 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 $\mathsf{Eval}$. For example, if $\mathcal{M} = \{0,1\}$, then functions may be described by boolean circuits with $|f|$ gates with bounded fan-in. For simplicity, we identify a function $f$ with its description (using some standard encoding), and write $f(\mathbf{m})$ for the result of evaluating $f$ at $\mathbf{m}$, and $|f|$ for the size of the description of $f$.

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

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 $\ell$ as an auxiliary parameter, and runs in time polynomial in both $\kappa$ and $\ell$. In particular, this allows $\mathsf{Enc}$, $\mathsf{Dec}$ and $\mathsf{Eval}$ to also run in time polynomial in $\ell$. 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 $\ell$. 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.[12] 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. As a side note, function privacy is one of the main motivations for work on ciphertext sanitization [DS16], and the basis of the stronger notion of circuit-privacy$^+$ introduced in [AV21, AGHV25]. The reader interested in function privacy, the related notion of ciphertext sanitization, and other forms of security in the presence of active attacks is referred to [DS16, AV21, AGHV25] as a starting point for further reading.

# 3 Full Composability

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

**Definition 6** (Composable FHE)**.** A (syntactically) homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with messages, ciphertexts and functions $\mathcal{M}, \mathcal{C}, \mathcal{F}$ is *fully composable* if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a valid[13] encryption scheme, and for all functions $f\colon \mathcal{M}^w \to \mathcal{M}$ in $\mathcal{F}$, ciphertexts $\mathbf{c} \in \mathcal{C}^w$ such that $\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}) \in \mathcal{M}^w$, and random keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$, we

[12] 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 $f$. However, for schemes that are correct only in an approximate sense, the situation is more complex, and the security definition needs to use also the $\mathsf{Eval}$ and $\mathsf{Dec}$ functions [LM21].

[13] We recall that a scheme is valid if it satisfies the standard correctness property $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$ with overwhelming probability.

have

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) \tag{3}$$

with overwhelming probability over the randomness of Gen, Enc and Eval.

Note that in Definition 6 the input function $f$ and ciphertexts $\mathbf{c}$ are universally quantified. This is similar to (and in the same spirit of) the universal quantification over the messages $\mathbf{m}$ and function $f$ in Definitions 1 and 5. In particular, it should not be misinterpreted as an attempt to model security against some form of chosen ciphertext attack, e.g., as done in [AGHV25]. In this paper we only consider the basic notion of IND-CPA security (Definition 2) commonly assumed in the context of fully homomorphic encryption. We emphasize that Definition 6 (just like Definitions 1 and 5) specifies a *correctness* property. While the distinction between security and correctness properties may be a bit blurry (especially so in light of attacks [HNP+03, LM21] showing how failure of correctness can have dramatic effects on security,) there is a fundamental difference: correctness (even when probabilistic) is typically proved unconditionally, while security necessarily requires[14] computational hardness assumptions and can only hold against computationally bounded adversaries. For this reason, in the context of correctness properties, it is customary to make no distinction between computationally bounded and unbounded adversaries, and liberally use universal quantification on the inputs. But the motivation is just simplicity, rather than capturing security against more powerful attackers. The advantages should be clear and familiar to any working cryptographer: simpler definitions (as opposed to the interactive adversarial games required by computational security properties) and much simpler and direct proofs (instead of reduction arguments based on computational complexity assumptions). The use of ciphertexts $\mathbf{c}$ as input is somehow a peculiarity of Definition 6, as correctness properties typically use only plaintext values. Yet, the fact remains that it is a correctness property which can be reasonably expected (and proved) to hold unconditionally. Computational versions of correctness may still be useful in more specialized settings, and the interested reader is referred to [CHI+21, ABMP24] as examples of works that use game-based correctness definitions of homomorphic encryption with adversarially chosen computations.

For brevity, if the scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is a (syntactically) homomorphic encryption scheme satisfying full composability (Definition 6), we refer to it as a *Composable FHE* scheme. Note that Definition 6 imposes no requirements on the evaluation and decryption functions when the input ciphertexts are not valid. In particular, $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, \mathbf{c}))$ is not required to output $\perp$ when $\mathsf{Dec}(\mathsf{sk}, c_i) = \perp$ for some $i$. It is easy to see that any fully composable scheme satisfies the standard homomorphic correctness property from Definition 5.

**Theorem 1.** *For any (syntactically) homomorphic encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*, full composability (Definition 6) implies homomorphic correctness (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 $f\colon \mathcal{M}^w \to \mathcal{M}$ be any function in $\mathcal{F}$, $\mathbf{m} \in \mathcal{M}^w$, 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 $\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}) = \mathbf{m}$ with all but negligible probability. It follows from the full composability property that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) = f(\mathbf{m})$. This proves that the scheme satisfies Definition 5, again with all but negligible probability. □

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

[14] Short of proving that $P \neq NP$ and resolving other long standing open problems in computational complexity theory. This is true for all cryptographic primitives that are sufficiently complex to imply the existence of one-way functions. Unconditional security properties arise in the context of information theoretic cryptography, but are not considered in this paper.

postpone the formal proof of this statement as we will derive it as a corollary of a more general result. (See Corollary 2.) Instead, we first analyze the composition 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 homomorphic correctness (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 functions $\mathcal{F} \subseteq \bigcup_w (\mathcal{M}^w \to \mathcal{M})$, let $\mathcal{F}^{\leq d}$ be the set of all computations $F\colon \mathcal{M}^w \to \mathcal{M}$ 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 $\mathsf{Eval}$ of an $\mathcal{F}$-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 evaluate each $f$-labeled gate of $F$.

Sometimes it is useful to restrict the evaluation function $\mathsf{Eval}^*$ to "layered" circuits, i.e., circuits $C(x_1, \ldots, x_n) \in \mathcal{F}^*$ where gates are arranged into layers $\ell = 1, \ldots, d$. Gates in the first layer $\ell = 1$ are applied to the circuit input values $x_1, \ldots, x_n$, while gates in higher layers $\ell > 1$ take inputs from gates at layer $\ell - 1$. The output of the circuit is given by the gate(s) in the last layer $\ell = d$. We write $\mathcal{F}^\sharp$ for the set of layered circuits, and $\mathcal{F}^{\sharp d} = \mathcal{F}^\sharp \cap \mathcal{F}^{\leq d}$ for the layer circuits of depth bounded by $d$.

It easily follows by induction (and a union bound on all the correctness failure events) 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 $\mathsf{Dec}(\mathsf{sk}, c) = F(\mathbf{m})$ with overwhelming probability. Moreover, $\mathsf{Eval}^*$ is still fully composable. 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 (with function set $\mathcal{F}$), then $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is also fully composable (with function set $\mathcal{F}^*$). In particular, by Corollary 1, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is $\mathcal{F}^*$-homomorphic.*

*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 theorem statement is equivalent to the assumption that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is fully composable with function set $\mathcal{F}$.

For the inductive case, let $F\colon \mathcal{M}^w \to \mathcal{M}$ 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

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

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

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

with overwhelming probability. By a union bound on all the correctness failure events, the total probability is proportional to the number of gates in $F$ (times negligible.) So, it is still negligible. This completes the proof that $\mathsf{Eval}^*$ is fully composable, and, by Corollary 1, also $\mathcal{F}^*$-homomorphic. □

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

The property established in Theorem 2 is closely related to a (somehow weaker) notion of composition proposed in [GHV10] under the name of *multi-hop* homomorphic encryption. Using our notation, multi-hop homomorphic encryption can be equivalently[15] 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 $\mathcal{F}$. We say that the scheme is a $d$-hop (resp. *multi-hop*) $\mathcal{F}$-homomorphic if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is $\mathcal{F}^{\leq d}$-homomorphic (resp. $\mathcal{F}^*$-homomorphic). We say that the scheme is *layered* $d$-hop (resp. *layered multi-hop*) $\mathcal{F}$-homomorphic if $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is $\mathcal{F}^{\sharp d}$-homomorphic (resp. $\mathcal{F}^\sharp$-homomorphic.)

Note that just as in Definition 5, the multi-hop homomorphic correctness property allows for (*surjective*) to fail with nonzero (but negligible) probability. At the end of this section we will introduce a notion of surjective encryption scheme that makes sense only for schemes that satisfy (*surjective*) with probability 1 (over the randomness of $\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}$.) We refer to this property as *perfect correctness*, and make use of it in Lemma 1 and Theorem 5.

Notice that the definition of 1-hop homomorphic encryption scheme is the same as 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 homomorphic encryption scheme. By Theorem 2, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}^*)$ is $\mathcal{F}^*$-homomorphic. So, by definition, the scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is multi-hop $\mathcal{F}$-homomorphic. □

As a side remark, all results so far are easily adapted to the case of schemes with perfect correctness. So, for example, by Corollary 1, if a scheme is fully composable with perfect correctness, then it is also multi-hop with perfect correctness. Later in this section (Theorem 5) we will prove a partial converse of this statement, showing that for a certainly class of "surjective" encryption schemes (see Definition 9), perfect multi-hop correctness implies full composability (also with perfect correctness.) So, in the case of surjective encryption schemes, perfect multi-hop correctness is equivalent to perfectly correct full composability.

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 the evaluation of arbitrary circuits with gates in $\mathcal{F}^*$.

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

**Theorem 3.** *Under the minimal assumption that (secure) $d$-hop $\mathcal{F}$-homomorphic encryption schemes exist at all, there are (secure) $d$-hop $\mathcal{F}$-homomorphic encryption schemes that are not $(d+1)$-hop homomorphic.*

[15] 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 $f\colon \mathcal{M}^w \to \mathcal{M}$.

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

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

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 the modified scheme $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Eval}')$ is still $d$-hop homomorphic, but not $(d+1)$-hop homomorphic. □

**Corollary 2.** *Under the same minimal assumption as Theorem 3, for any $d$, there are (secure) $d$-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.** *For any nontrivial set of functions $\mathcal{F} \neq \emptyset$, under the minimal assumption that that (secure) multi-hop $\mathcal{F}$-homomorphic encryption schemes exist at all, 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

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

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

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

Now, let $f \in \mathcal{F}$ be a function and $\mathbf{m} \in \mathcal{M}^*$ be a sequence of input messages. The ciphertexts $c_i = (1, \mathsf{Enc}(\mathsf{pk}, m_i))$ satisfy

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

So, the scheme is not fully composable. □

So, full composability is a strictly stronger notion than multi-hop homomorphic 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 essentially 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 messages, ciphertexts and functions $\mathcal{M}, \mathcal{C}, \mathcal{F}$) 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 $\mathbf{m} \in \mathcal{M}^w$ such that

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

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

Notice that in the definition of surjective encryption scheme, the probability of each ciphertext $c$ is nonzero, but it can be (and it typically is) very small. In fact, for the scheme to be secure, each individual ciphertext cannot occur with more than negligible probability. Due to this technicality, the next lemma and theorem (or, more generally, all results making use of surjective property) are restricted to schemes with perfect correctness.

The lemma shows that the decryption function $\mathsf{Dec}$ of a surjective encryption scheme never outputs $\perp$, i.e., all possible ciphertexts $c \in \mathcal{C}$ are valid.

**Lemma 1.** *For any surjective homomorphic encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *with perfect multi-hop correctness, keys* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}$ *and ciphertext* $c \in \mathcal{C}$*, we have* $\mathsf{Dec}(\mathsf{sk}, c) \neq \perp$*.*

*Proof.* Let $c$ be an arbitrary ciphertext. By definition of surjective scheme, there are $F \in \mathcal{F}^*$ and $\mathbf{m} \in \mathcal{M}^w$ such that $c = \mathsf{Eval}^*(\mathsf{pk}, F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))$ with nonzero probability. We also know from the multi-hop homomorphic perfect correctness property that

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}^*(F, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}))) = F(\mathbf{m}) \neq \perp.$$

So, it must be $\mathsf{Dec}(\mathsf{sk}, c) \neq \perp$. □

Finally, we show that if we restrict our attention to surjective encryption schemes with perfect correctness, then full composability is equivalent to multi-hop homomorphism.

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

*Proof.* Let $f : \mathcal{M}^w \to \mathcal{M}$ be any function in $\mathcal{F}$, and $\mathbf{c} \in \mathcal{C}^w$ a vector of ciphertexts. We need to prove that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathbf{c})) = f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c}))$. 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 $\mathsf{Eval}(\mathsf{pk}, F_i, \mathsf{Enc}^*(\mathsf{pk}, \mathbf{m}_i)) = c_i$ with nonzero probability. If follows from the multi-hop homomorphic perfect correctness property that $\mathsf{Dec}(\mathsf{sk}, c_i) = F_i(\mathbf{m}_i)$. 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 multi-hop (perfectly) homomorphic, we have

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

with probability 1. But, by definition of $\mathsf{Eval}^*$ and $F$, we also have

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

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

Together with Corollary 1, this shows that for surjective encryption schemes, perfect multi-hop correctness is equivalent to perfect full composability. Note however that none of these results shows how to obtain a surjective encryption scheme, even from a fully composable one with perfect correctness. In fact, at present, Theorem 5 is mostly of theoretical interest, and there are gaps between the theorem assumptions, and current constructions of fully homomorphic encryption schemes. To start with, for efficiency reasons, practical FHE schemes often do not satisfy perfect correctness. This is not, by itself, a fundamental barrier, as perfect correctness can be easily achieved by a conservative choice of the scheme parameters of current lattice-based constructions. (E.g., using truncated Gaussian error distributions and worst-case bounds on noise growth during homomorphic operations.) However, since this results in (theoretically modest, but) practically significant decrease in performance, it is common practice to use more aggressive parameter settings and probabilistic bounds on decryption errors. A more fundamental gap in the applicability of Theorem 5 to current lattice-based FHE constructions is given by the assumption that the encryption scheme is surjective. In a sense, lattice-based encryption schemes are surjective, meaning that any ciphertext can be obtained with nonzero probability by running the encryption algorithm with a sufficiently wide "noise distribution", or using a sufficiently complex sequence of homomorphic operations. However, in doing so, one would almost certainly generate also ciphertexts that do not decrypt correctly. In order to keep the decryption error probability negligible, one would have to stop performing homomorphic operations before Definition 9 is satisfied. So, building a homomorphic encryption scheme that is both (at the same time) surjective and perfectly correct (or even achieve negligible correctness error) is an open problem.

**A counterexample to a generic construction based on ciphertext sanitization.** We conclude this section by giving a counterexample showing that modifying a homomorphic encryption scheme using a sanitization algorithm, as suggested in [AV21, AGHV25], does not necessarily result in an encryption scheme that is homomorphically correct. This shows that an appealing approach to achieve full composability based on the notion of circuit-privacy$^+$ introduced in [AV21, AGHV25] cannot be instantiated *generically* using an arbitrary homomorphic encryption scheme and sanitization algorithm. We recall from [DS16] that a sanitization algorithm for an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a publicly computable, randomized algorithm $\mathsf{San}(\mathsf{pk}, \cdot)$ mapping ciphertexts to ciphertexts, and satisfying the following two properties, with overwhelming probability over the choice of keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(\kappa)$:

- (Message preservation) For any ciphertext $c \in \mathcal{C}$, $\mathsf{Dec}(\mathsf{sk}, \mathsf{San}(\mathsf{pk}, c)) = \mathsf{Dec}(\mathsf{sk}, c)$, i.e., sanitization preserves the encrypted message.
- (Sanitization) For any two valid ciphertexts $c, c'$ with the same decryption $\mathsf{Dec}(\mathsf{sk}, c) = \mathsf{Dec}(\mathsf{sk}, c')$, the distributions $\mathsf{San}(\mathsf{pk}, c)$ and $\mathsf{San}(\mathsf{pk}, c')$ are statistically close, i.e., the sanitized distribution $\mathsf{San}(\mathsf{pk}, c)$ depends only on the decrypted value $\mathsf{Dec}(\mathsf{sk}, c)$.

**Lemma 2.** *There is a homomorphically correct encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *and sanitization algorithm* $\mathsf{San}$ *for* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that the modified homomorphic encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}^{santz}, \mathsf{Dec}, \mathsf{Eval}^{santz})$ *proposed in [AGHV25, Definition 8] is not homomorphically correct.*

*Proof.* Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a homomorphically correct encryption scheme and $\mathsf{San}$ a sanitization algorithm for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. (If no such schemes exist, then the construction proposed in [AGHV25, Definition 8] is uninstantiatable.) To start with, modify the schemes as follows:

- $\mathsf{Enc}'(\mathsf{pk}, m) = (0, \mathsf{Enc}(\mathsf{pk}, m))$

- $\mathsf{Dec}'(\mathsf{sk}, (b, c)) = \mathsf{Dec}(\mathsf{sk}, c)$
- $\mathsf{Eval}'(\mathsf{pk}, f, \{(b_i, c_i)_i\}) = (0, \mathsf{Eval}(\mathsf{pk}, f, \{c_i\}))$ if $\forall i. b_i = 0$, and output $\perp$ otherwise.
- $\mathsf{San}'(\mathsf{pk}, (b, c)) = (1, \mathsf{San}(\mathsf{pk}, c))$.

Informally, the modified scheme adds a bit $b \in \{0, 1\}$ to the ciphertexts to indicate if they were obtained using $\mathsf{Enc}$ and $\mathsf{Eval}$, or $\mathsf{San}$. It is easy to check that the modified scheme $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Eval}')$ is still homomorphically correct (because it does not use $\mathsf{San}$), and $\mathsf{San}'$ is a good sanitization algorithm for $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ (because $\mathsf{San}'(\mathsf{pk}, c)$ depends only only $\mathsf{Dec}(\mathsf{sk}, c)$.)

We recall that the modified scheme from [AGHV25, Definition 8] defines

- $\mathsf{Enc}^{santz}(\mathsf{pk}, m) = \mathsf{San}(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m))$, and
- $\mathsf{Eval}^{santz}(\mathsf{pk}, f, \mathbf{c}) = \mathsf{San}(\mathsf{pk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{San}(\mathsf{pk}, \mathbf{c})))$

while keeping $\mathsf{Gen}$ and $\mathsf{Dec}$ unmodified. We apply this construction to our modified scheme $(\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}', \mathsf{Eval}')$ and sanitization algorithm $\mathsf{San}'$. It is immediate to check that the resulting scheme is not homomorphically correct because $\mathsf{Eval}'$ outputs $\perp$ on the ciphertexts produced by $\mathsf{San}'$. □

# 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 by composition (Theorem 2). We begin by describing the bootstrapping construction of [Gen09b].

**Definition 10** (Bootstrapping construction)**.** Fix a set $\mathcal{F}$ of functions $f\colon \mathcal{M}^w \to \mathcal{M}$, 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 $f_{\mathbf{c}}^{\circlearrowleft}\colon \mathcal{M}^k \to \mathcal{M}$ indexed by $f \in \mathcal{F}$ and $\mathbf{c} \in \mathcal{C}^w$ defined as

$$f_{\mathbf{c}}^{\circlearrowleft}(\mathbf{x}) = \begin{cases} f(\mathsf{Dec}^*(\mathsf{sk}, \mathbf{c})) & \text{if } \mathbf{x} = \psi(\mathsf{sk}) \text{ for some } \mathsf{sk} \in \mathcal{K} \\ \perp & \text{otherwise.} \end{cases} \tag{4}$$

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

- $\mathsf{Gen}^\psi(\kappa) = (\mathsf{sk}, (\mathsf{pk}, \mathsf{Enc}^*(\mathsf{pk}, \psi(\mathsf{sk}))))$ and $\mathsf{Enc}^\psi((\mathsf{pk}, \mathsf{pk}'), m) = \mathsf{Enc}(\mathsf{pk}, m)$ are the key generation and encryption algorithms from Definition 3,
- the decryption algorithm $\mathsf{Dec}$ is the same as that of $\mathsf{FHE}$, and
- the evaluation function is $\mathsf{Eval}^{\circlearrowleft}((\mathsf{pk}, \mathsf{pk}'), f, \mathbf{c}) = \mathsf{Eval}(\mathsf{pk}, f_{\mathbf{c}}^{\circlearrowleft}, \mathsf{pk}')$.

In words, $\mathsf{FHE}^{\circlearrowleft}$ evaluates a function $f$ homomorphically on a ciphertext $\mathbf{c}$ by using $\mathbf{c}$ to select a function $f_{\mathbf{c}}^{\circlearrowleft}$ from $\mathcal{F}_\psi^{\circlearrowleft}$, and then evaluating this function (using $\mathsf{FHE}$) on a fixed ciphertext $\mathsf{pk}'$ which is part of the public key. The following theorem shows that this "bootstrapping" construction produces a fully composable homomorphic encryption scheme.

**Theorem 6.** *For any set $\mathcal{F}$ of functions $f\colon \mathcal{M}^w \to \mathcal{M}$ and encoding $\psi\colon \mathcal{K} \to \mathcal{M}^k$, let $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a valid $\mathcal{F}_\psi^{\circlearrowleft}$-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 $\mathsf{FHE}$ is $\psi$-circular IND-CPA secure, then $\mathsf{FHE}^{\circlearrowleft}$ is also ($\psi$-circular) IND-CPA secure.*

*Proof.* The IND-CPA security of $\mathsf{FHE}^{\circlearrowleft}$ immediately follows from the assumption that $\mathsf{FHE}$ is $\psi$-circular IND-CPA secure (Definition 3). Moreover, appending[16] $\mathsf{Enc}^{\psi *}((\mathsf{pk}, \mathsf{pk}'), \mathsf{sk}) = \mathsf{Enc}^*(\mathsf{pk}, \mathsf{sk}) = \mathsf{pk}'$ to the public key $(\mathsf{pk}, \mathsf{pk}')$ does not add any new information. So, $\mathsf{FHE}^{\circlearrowleft}$ is $\psi$-circular IND-CPA secure. 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 $m \in \mathcal{M}$, with all bu negligible probability over the key generation and encryption randomness. So, $\mathsf{FHE}^{\circlearrowleft}$ is a valid encryption scheme.

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

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

This proves the full composability property. □

Notice that Definition 10 and Theorem 6 transform a (non-composable) scheme $\mathsf{FHE}$ that supports only the evaluation of functions $f_{\mathbf{c}}^{\circlearrowleft}(\mathbf{x})$ with a *fixed* number of inputs $\mathbf{x} \in \mathcal{M}^k$ (determined by the encoding function $\psi$), into a scheme $\mathsf{FHE}^{\circlearrowleft}$ that supports the (composable) evaluation of functions $f$ with an arbitrary number of inputs $w$. The larger is the set of functions $\mathcal{F}$ we want $\mathsf{FHE}^{\circlearrowleft}$ to support, the larger is the set $\mathcal{F}_\psi^{\circlearrowleft}$ for which $\mathsf{FHE}$ is required to be homomorphic to start with. However, this is typically not necessary, and $\mathcal{F}$ is usually a small (finite) set of functions, 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 a valid, $\mathcal{F}_\psi^{\circlearrowleft}$-homomorphic, $\psi$-circular IND-CPA secure encryption scheme. Then $\mathsf{FHE}^{\circlearrowleft *}$ is valid, $\mathcal{F}^*$-homomorphic, fully composable, and $\psi$-circular secure.*

*Proof.* Follows from Theorem 6 and Theorem 2. □

The bootstrapping theorem, as stated above, requires the starting scheme $\mathsf{FHE}$ to be circular secure. If $\mathsf{FHE}$ is only IND-CPA secure, we can still achieve a limited form of composition using leveled bootstrapping. In the following construction, ciphertexts are tagged with an integer $\ell$ corresponding to their level in the homomorphic computation, starting from $\ell = 0$ for the input layer, all the way to the final output of a computation of depth $\ell = d$. Gates are evaluated similarly to Theorem 6, but using a different pair of keys for each layer of the computation.

[16] Technically, since $\mathsf{Enc}(\mathsf{pk}, \mathsf{sk})$ is randomized, this may produce ciphertext different from $\mathsf{pk}'$. So, one should assume that the original scheme is circular secure even when given multiple encryptions of $\mathsf{sk}$. In practice, this additional ciphertext serves no purpose, and can be omitted from the public key.

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

- $\mathsf{Gen}^{\sharp}_d(\kappa)$ runs $(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathsf{Gen}(\kappa)$ 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 $\ell$, 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* $\mathcal{F}^{\circlearrowleft}_{\psi}$*-homomorphic, then* $\mathsf{FHE}^{\sharp}$ *is leveled* $\mathcal{F}^{\sharp d}$*-homomorphic. 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. □

**Applicability to current FHE constructions** A natural question at this point may be if currently known lattice-based FHE constructions are already fully composable, or if they can be made so. A first cut answer to this question is "yes" for FHEW-like encryption schemes (e.g., [DM15, CGGI20, MP21, LMK+23]) which already incorporate bootstrapping (or "functional bootstrapping", see Section 5) as part of their evaluation procedure. While for BGV-style FHE schemes (e.g., [BV14, BGV14, Bra12]) the answer is "no, but the scheme can be made fully composable using bootstrapping" as formally proved in our Theorem 6". In fact, this is essentially what all these papers already do, even if without using a formal definition of full composability. However, when looking at the details of specific instantiations and implementations of these schemes, the answer is a little bit more complex. As it should already clear to the reader by now (if not, this is made even clearer in the Section 5 where we formalize the notion of wire-bootstrappable and functional bootstrapping), "bootstrapping" is nothing other than just a homomorphic evaluation of the decryption function $\mathsf{Dec}(\mathsf{sk}, c)$ on an encryption of $\mathsf{sk}$. If the bootstrapping algorithm of a specific FHE scheme faithfully implements the decryption function $\mathsf{sk} \mapsto \mathsf{Dec}(\mathsf{sk}, c)$ for any ciphertext $c \in \mathcal{C}$ in the ciphertext space, then, by Theorem 6 the resulting scheme is guaranteed to be fully composable. However (in practice, as an optimization) one may take advantage of the fact the current FHE schemes are not surjective (see discussion following Theorem 5), and try to optimize the scheme by implementing the bootstrapping function $\mathsf{sk} \mapsto \mathsf{Dec}(\mathsf{sk}, c)$ only for a subset of ciphertexts $c$, namely those that can actually occur as the output of a homomorphic computation. In this case, even when used with bootstrapping, the resulting scheme may not be fully composable. So, in practice, whether a concrete instantiation of an FHE scheme achieves full composability is often dependent on specific parameter settings and implementation details.

# 5 Optimizations

In this section we discuss some optimizations that are commonly used to improve the efficiency of (fully composable) encryption schemes. There is one important aspect in which Theorem 6 differs from the way Gentry's bootstrapping technique is used in practice. In

Theorem 6 (as well as [Gen09b, Theorem 3]) full composability is achieved by preprocessing each *input* to a gate $f : \mathcal{M}^w \to \mathcal{M}$ with a copy of the decryption function $\mathsf{Dec}(\mathsf{sk}, c_i)$ for $i = 1, \ldots, w$. When several gates are combined in a circuit to perform a larger homomorphic computation, if a ciphertext $c_i$ is used as input to multiple gates (i.e., if the gate producing $c_i$ has fan-out higher than 1), then $c_i$ will be decrypted (homomorphically) multiple times, once for each gate that takes $c_i$ as input. So, the total number of homomorphic decryptions performed to evaluate a circuit with $n$ gates with fan-in $k$ is $k \cdot n$. In practice, homomorphic decryption is performed only once for each gate, on the output wire, and then the same "bootstrapped" ciphertext is used as input to multiple gates without further preprocessing. This reduces the number of homomorphic decryptions from $k \cdot n$ to just $n$. This optimization is captured by the following definition. In order to distinguish the properties/assumptions used in Definition 10/Theorem 6, and those in the following definition, we will refer to the schemes of Definition 10 as *gate-bootstrappable*, and those in the next definition as *wire-bootstrappable*.

**Definition 12.** An $\mathcal{F}$-homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is *wire-bootstrappable* if there is an efficient algorithm $\mathsf{Boot}(\mathsf{pk}, c)$ that on input a public key $\mathsf{pk}$ and ciphertext $c$ outputs another ciphertext such that for all $f \in \mathcal{F}$, keys $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$, and valid[17] ciphertexts $\mathbf{c} \in \mathcal{C}^w$

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Boot}(\mathsf{pk}, c_1), \ldots, \mathsf{Boot}(\mathsf{pk}, c_w))) = f(\mathsf{Dec}(\mathsf{sk}, c_1), \ldots, \mathsf{Dec}(\mathsf{sk}, c_w))$$

with overwhelming probability over the randomness of $\mathsf{Gen}$,$\mathsf{Enc}$ and $\mathsf{Eval}$.

Using this definition, an wire-bootstrappable scheme can be used to homomorphically evaluate any circuit with gates in $\mathcal{F}$ in the obvious way, applying $\mathsf{Boot}$ to the output of each gate. This is formalized in the following theorem.

**Theorem 8.** *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be an encryption scheme which is* $\mathcal{F}$*-homomorphic and wire-bootstrappable, with evaluation procedure* $\mathsf{Eval}$ *and bootstrapping algorithm* $\mathsf{Boot}$*. Define a modified evaluation function* $\mathsf{Eval}'$ *that on input a circuit* $C(x_1, \ldots, x_n) \in \mathcal{F}^*$ *and* $n$ *input ciphertext* $c_1, \ldots, c_n$*, computes, for each circuit gate* $x_i = f(x_I)$*, the ciphertexts*

$$c'_i = \mathsf{Eval}(\mathsf{pk}, c_I), \qquad c_i = \mathsf{Boot}(\mathsf{pk}, c'_i).$$

*For each output gate* $x_i = f(x_I)$*,* $\mathsf{Eval}'$ *outputs* $c'_i$*. Then,* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}')$ *is* $\mathcal{F}^\sharp$*-homomorphic.*[18]

*Proof.* Let $x_i$ be the outputs of each gate of $C$ when the circuit is evaluated in the clear. It easily follows by induction that for all gates $i$, $\mathsf{Dec}_i(\mathsf{sk}, c'_i) = x_i$:

- For the first layer of gates, this follows from the $\mathcal{F}$-homomorphic property

$$\mathsf{Dec}(\mathsf{sk}, c'_i) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, c_I)) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Enc}(\mathsf{pk}, x_I))) = f(x_I)$$

- For all other gates, we have

$$\begin{aligned} \mathsf{Dec}(\mathsf{sk}, c'_i) &= \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, c_I)) \\ &= \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Boot}(\mathsf{pk}, c'_I))) = f(\mathsf{Dec}(\mathsf{sk}, c'_I)) = f(x_I). \end{aligned}$$

[17] We recall that a ciphertext $c$ is *valid* if $\mathsf{Dec}(\mathsf{sk}, c) \neq \perp$. A ciphertext vector $\mathbf{c}$ is valid if $\mathsf{Dec}(\mathsf{sk}, c_i) \neq \perp$ for all $i$. The restriction to valid ciphertexts is just a technicality, to make sure that the value of $f(x_1, \ldots, x_w)$ is well defined when $x_i = \mathsf{Dec}(\mathsf{sk}, c_i)$. The important part of the definition is that $\mathbf{c}$ is quantified over *all* possible ciphertexts in $\mathcal{C}$ (not necessarily in the support of $\mathsf{Enc}(\mathsf{pk}, \cdot)$,) with the only exception of those that are explicitly detected as invalid by the decryption function.

[18] For simplicity, we assumed the input circuit is layered. This is easily generalized to arbitrary circuits by combining Definition 4 and Definition 12 into a single property $\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, c_1, \ldots, c_w)) = f(x_1, \ldots, x_w)$ where each input $c_i$ is either a fresh ciphertext $\mathsf{Enc}(\mathsf{pk}, x_i)$, or $\mathsf{Boot}(\mathsf{pk}, c_i)$ for some $c_i$ such that $\mathsf{Dec}(\mathsf{sk}, c_i) = x_i$.

So, the final output of the homomorphic evaluation $c' = \mathsf{Eval}^\sharp(\mathsf{pk}, C, c_1, \ldots, c_n)$ satisfies $\mathsf{Dec}(\mathsf{sk}, c') = C(x_1, \ldots, x_n)$. As usual, the correctness property at each step may fail with negligible probability, and the overall failure probability is negligible by a union bound. □

We remark that Definition 12 is somehow different from [Gen09b, Definition 5], where a "bootstrappable" scheme is defined as a scheme supporting the construction in Definition 10, and which we call "gate-bootstrappable". However, since this optimization has the potential of speeding up homomorphic computations by a factor $k$ (equal to the gates' fan-in), this is how bootstrapping is implemented in practice.

Wire-bootstrappability is easily related (at least in theory) to full composability, as shown in the next simple theorems.

**Theorem 9.** *Any fully composable encryption scheme is wire-bootstrappable.*

*Proof.* Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be any fully composable encryption scheme, and let $\mathsf{Boot}(\mathsf{pk}, c) = c$ be the identity function. Then, Definition 12 becomes equivalent to Definition 6. □

**Theorem 10.** *Any wire-bootstrappable encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *can be turned into a fully composable one* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}')$ *supporting the same set of functions* $\mathcal{F}$*.*

*Proof.* Let $\mathsf{Boot}$ be the bootstrapping algorithm from Definition 12, and define

$$\mathsf{Eval}'(\mathsf{pk}, f, c_1, \ldots, c_w) = \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Boot}(\mathsf{pk}, c_1), \ldots, \mathsf{Boot}(\mathsf{pk}, c_w)).$$

Then, by definition of wire-bootstrappability, $\mathsf{Eval}'$ satisfies Definition 6. □

Naturally, turning an wire-bootstrappable encryption scheme into a fully composable one (using Theorem 10) and then evaluating a circuit homomorphically (using Theorem 2) is unnecessarily inefficient, computing $\mathsf{Boot}$ multiple times for each output wire. In order to save a factor $k$ one needs to make direct use of $\mathsf{Boot}$ as described above.

Algorithm $\mathsf{Boot}$ can be thought of as evaluating the identity function homomorphically. In fact, assuming without loss of generality that $\mathsf{Eval}(\mathsf{pk}, \mathsf{id}, c) = c$, Definition 12 with $f = \mathsf{id}$ reduces to

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Boot}(\mathsf{pk}, c)) = \mathsf{Dec}(\mathsf{sk}, c).$$

(Note however that $\mathsf{Boot}$ is generally not the identity function on ciphertexts.) There is no need to restrict $\mathsf{Boot}$ to the evaluation of the identity function, and one can define a more general notion of *functional bootstrapping*. We remark that functional bootstrapping always refers to an wire-bootstrappability property of the type described in Definition 12.

**Definition 13** (Functional Bootstrapping)**.** A $\mathcal{F}$-homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ supports *functional bootstrapping* with function set $\mathcal{G} \subseteq \mathcal{M} \to \mathcal{M}$ if there is an efficient algorithm $\mathsf{Boot}^g(\mathsf{pk}, c)$ such that for all $f \in \mathcal{F}$, $g_1, \ldots, g_w \in \mathcal{G}$, valid ciphertexts $\mathbf{c} \in \mathcal{C}^w$ and random $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}(\kappa)$,

$$\begin{aligned}\mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, f, \mathsf{Boot}^{g_1}(\mathsf{pk}, c_1), \ldots, \mathsf{Boot}^{g_w}(\mathsf{pk}, c_w))\\ = f(g_1(\mathsf{Dec}(\mathsf{sk}, c_1)), \ldots, g_w(\mathsf{Dec}(\mathsf{sk}, c_w))).\end{aligned}$$

Definition 12 is a special case of Definition 13 with the trivial set $\mathcal{G} = \{\mathsf{id}\}$. Functional bootstrapping can be used to homomorphically evaluate circuits with gates in $\mathcal{G} \circ \mathcal{F} = \{g \circ f \colon f \in \mathcal{F}, g \in \mathcal{G}\}$, where $(g \circ f)(\mathbf{x}) = g(f(\mathbf{x}))$ is the standard function composition operation. Circuits with gates in $\mathcal{G} \circ \mathcal{F}$ are evaluated as follows:

- input wires are labeled with the corresponding ciphertext,

- for each gate $g \circ f$ with input $c_1, \ldots, c_w$, compute the ciphertext

$$c' = \mathsf{Boot}^g(\mathsf{pk}, \mathsf{Eval}(\mathsf{pk}, f, c_1, \ldots, c_w))$$

  and label the gate output wire with $c'$.

The use of functional bootstrapping may seem redundant at first, as one can assume Eval already supports the evaluation of functions in $\mathcal{G} \circ \mathcal{F}$.[19] The motivation for functional bootstrapping is again practical: functional bootstrapping can lead to substantial efficiency gains. The idea was first introduced by the FHEW cryptosystem in [DM15], where it is observed that the NAND boolean gate (as well as any other symmetric[20] boolean function) can be expressed as addition modulo a small $p > 2$ followed by a nonlinear operation mapping $\{0, 1, 2\} \subseteq \mathbb{Z}_p$ to $\{0, 1\} \subset \mathbb{Z}_p$. This allows to perform arbitrary homomorphic computations starting from an encryption scheme (Gen, Enc, Dec, Eval) that is only linearly homomorphic, i.e., with $\mathcal{F} = \{+\}$. Lattice-based encryption schemes (which are the basis of essentially all known FHE constructions) are naturally linearly homomorphic, and schemes supporting just the addition operation are much simpler than those also supporting the homomorphic evaluation of multiplication (or other non-linear operations). This results in much smaller parameters and computationally simpler decryption function Dec. Since Boot is typically implemented as a homomorphic evaluation of Dec on the encryption of the secret key (e.g., see Definition 10), this translates to a much simpler and faster bootstrapping procedure Boot. Moreover, the bootstrapping algorithm underlying [DM15] supports functional bootstrapping for free, at essentially no additional cost. The end result is a very fast bootstrapping procedure, fast enough that it becomes feasible to perform (functional) bootstrapping after each (boolean) gate. As an added benefit, the resulting scheme is fully composable, supporting a much simpler model of computation (where programs are arbitrary boolean circuits) than previous practical schemes which reduced the *amortized* cost of bootstrapping by batching many (often tens of thousands) computations together. The efficiency of the bootstrapping algorithm of [DM15] has been further improved in several other so-called "FHEW-like" cryptosystems [CGGI20, MP21, LMK+23] which, following [DM15], combine a linearly homomorphic base encryption scheme with a (non-linear) functional bootstrapping procedure.

# 6 Conclusions and Open Problems

We have presented a definition of homomorphic encryption that allows the arbitrary composition of homomorphic computations, and investigated its relation to the traditional (non-composable) FHE definition [Gen09b] as well as other forms of composition considered in the past [GHV10]. Then we showed that this definition allows to formalize the bootstrapping technique of [Gen09b] (at the core of essentially all known FHE constructions) as a method to turn a non-composable FHE scheme into a composable one. We also gave similar definitions that more closely correspond to the way bootstrapping is used in practice. Beside providing a possible avenue to the construction of FHE schemes (e.g., as already done by FHEW-like cryptosystems [DM15, CGGI20, MP21, LMK+23],) we believe the new definition may prove useful to investigate questions that are central to the theory of homomorphic computations as we now explain.

All known constructions of fully homomorphic encryption schemes (aside from proposals relying on indistinguishability obfuscation [CLTV15]) make use of lattice-based

[19] One may also ask why start from a homomorphic encryption at all, when functional bootstrapping can already support the evaluation of arbitrary functions. The reason is that, by definition, $\mathcal{G}$ only contains unary functions. In order to combine multiple ciphertexts together, one needs $\mathcal{F}$ to contain at least one binary function $f \colon \mathcal{M}^2 \to \mathcal{M}$.

[20] Non symmetric gates are also easily handled by mapping input bits $x_0, x_1 \in \{0, 1\}$ to $x_0 + 2 \cdot x_1 \in \{0, 1, 2, 3\} \subseteq \mathbb{Z}_p$ for $p \geq 4$.

cryptography, which is inherently noisy. This requires the use of bootstrapping as a noise reduction technique for lattice-based ciphertexts, and circular security assumptions to implement bootstrapping. For this reason, a recurring question in the study of fully homomorphic encryption has been whether noise (with bootstrapping and circular security along with it) is necessary to achieve fully homomorphic encryption, or it is possible to build an FHE scheme which is "noiseless". However, it should be remarked that being "noisy" or "noiseless" is not an abstract property (which may or may not be satisfied by any encryption scheme,) but a peculiar characteristic of specific constructions (such as lattice based cryptography) where the encryption randomness can be naturally be interpreted as a noise term. So, the question of whether noise (and bootstrapping and circular security along with it) is necessary to build fully homomorphic encryption schemes is not really well posed. One possible way to formalize the question could be to consider homomorphic encryption schemes such that $\mathsf{Enc}_{\mathsf{pk}}(m; r \in \mathcal{R}_0)$ is secure even when the encryption randomness is restricted to a set $\mathcal{R}_0$, and such that $\mathsf{Eval}_{\mathsf{pk}}(f, \mathsf{Enc}(m; \mathcal{R}_i)) \subseteq \mathsf{Enc}_{\mathsf{pk}}(f(m); \mathcal{R}_{i+1})$ for larger and larger sets $\mathcal{R}_0 \subset \mathcal{R}_1 \subset \ldots$.

Our description of bootstrapping as a method to transform a (non-composable) homomorphic encryption scheme into a fully composable one (Theorem 6) offers a different framework to properly formalize and investigate this type of questions, completely bypassing the notion of "noisy" encryption scheme. For example one may ask:

> **Question:** Is circular security necessary to achieve full composability?

Note that the question does not make any reference to encryption noise, and all concepts (circular security, homomorphic encryption and full composability) have well formalized abstract cryptographic definitions. A possible way to address this question could be to show the following.

> **Conjecture 1:** Any fully composable homomorphic encryption scheme can be modified into a circular secure one.

In fact, one could ask if any fully composable homomorphic encryption scheme is already circular secure, but this is most likely false as one can adapt the simple counterexamples demonstrating the existence of circular insecure encryption schemes to the fully composable setting. So, the circular secure scheme of Conjecture 1 can be different from (but still depend on) the original composable FHE scheme. Given that circular security implies full composability (Theorem 6), proving the conjecture would show that circular security and full composability are essentially equivalent (assuming the existence of a non-composable encryption scheme with limited homomorphic properties, as those that can be built from lattices.)

We also remark that there are forms of circular security that seem sufficient to achieve full composability (extending Theorem 6), but are not covered by known separation results [KRW15, KW16, AP16, GKW17b, GKW17a, HK17]. Specifically, Theorem 6 makes a scheme $\mathsf{Enc}$ fully composable using an encryption cycle $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk})$ of length 1, but is easily generalized to longer encryption cycles $\mathsf{Enc}_{\mathsf{pk}_1}(\mathsf{sk}_2), \mathsf{Enc}_{\mathsf{pk}_2}(\mathsf{sk}_3), \ldots, \mathsf{Enc}_{\mathsf{pk}_n}(\mathsf{sk}_1)$, where $(\mathsf{pk}_i, \mathsf{sk}_i)$ are independently generated pairs of public/secret keys. Previous results have shown how to build encryption schemes $\mathsf{Enc}$ such that publishing such a cycle is insecure. So, one cannot achieve full composability generically by publishing such an encryption cycle for any scheme $\mathsf{Enc}$.

However, for the purpose of applying (a generalization of) Theorem 6 it is not necessary to use the same encryption scheme at every step of the cycle. So, for example, in order to make a scheme $\mathsf{Enc}$ fully composable it would be enough to show that there exists some other (possibly different) encryption scheme $\mathsf{Enc}'$ such that one can securely publish a cycle $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}'), \mathsf{Enc}'_{\mathsf{pk}'}(\mathsf{sk})$ that combines the two schemes. Note that the new (existentially quantified) scheme $\mathsf{Enc}'$ is not required to be homomorphic. In fact, given ciphertexts

$c = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}'), c' = \mathsf{Enc}'_{\mathsf{pk}'}(\mathsf{sk})$ (as key material), and an input ciphertext $c'' = \mathsf{Enc}_{\mathsf{pk}}(m)$ (to be bootstrapped), one can bootstrap $c''$ by first computing $\mathsf{sk}' \mapsto \mathsf{Dec}_{\mathsf{Dec}_{\mathsf{sk}'}(c')}(c'') = m$ homomorphically on $c$. Naturally, for this to be useful (in Theorem 6) the original (non-composable) scheme should support the homomorphic evaluation of this more complex function. Effectively, this is turning the 2-cycle $(c, c')$ into a simple cycle that encrypts $\mathsf{Dec}_{\mathsf{sk}'}(c') = \mathsf{sk}$ under $\mathsf{Enc}_{\mathsf{pk}}$, and then use that to bootstrap $c''$. However, this is different from computing a simple cycle $\mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk})$ directly, because the evaluated ciphertext follows a different distribution. So, it is not ruled out by previous separation results, and we conjecture the following.

> **Conjecture 2:** For any (public key) encryption scheme $\mathsf{Enc}$ there is a (possibly different) encryption scheme $\mathsf{Enc}'$ such that $\mathsf{Enc}(\mathsf{pk}, \cdot)$ is secure in the presence of side information $\mathsf{Enc}'_{\mathsf{pk}'}(\mathsf{sk}), \mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}')$ for a randomly chosen key pair $(\mathsf{pk}', \mathsf{sk}')$.

Several variants of this conjecture are possible. For example, one may consider cycles of length greater than 2, or set $\mathsf{Enc}'$ to a private key encryption scheme where the side information is $\mathsf{Enc}'_{\mathsf{sk}'}(\mathsf{sk}), \mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}')$, or consider the special case of encryption schemes that encrypt their messages bit by bit. Note that this Conjecture does not by itself imply the existence of composable FHE schemes. The reason is that for any homomorphic encryption scheme $\mathsf{Enc}$ (capable of evaluating functions in a given set $\mathcal{F}$), one may select a scheme $\mathsf{Enc}'$ such that the required computation $\mathsf{sk}' \mapsto \mathsf{Dec}_{\mathsf{Dec}_{\mathsf{sk}'}(c')}(c'')$ is not in $\mathcal{F}$. Still, proving that the conjecture is true would provide interesting information about the feasibility of achieving circular security in a generic way. In particular, if the starting scheme $\mathsf{Enc}$ is (non-composable) fully homomorphic (i.e., $\mathcal{F}$ is the set of all possible functions), this would be enough to achieve full composability.

## 7 Acknowledgments

The author wishes to thank Intel for its support through the Crypto Frontiers program, and the anonymous reviewers for their feedback and comments.

## References

[ABMP24] Andreea Alexandru, Ahmad Al Badawi, Daniele Micciancio, and Yuriy Polyakov. Application-aware approximate homomorphic encryption: Configuring FHE for practical use. *IACR Cryptol. ePrint Arch.*, page 203, 2024. URL: https://eprint.iacr.org/2024/203.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009. doi:10.1007/978-3-642-03356-8_35.

[AGHV25] Adi Akavia, Craig Gentry, Shai Halevi, and Margarita Vald. Achievable CCA2 relaxation for homomorphic encryption. *J. Cryptol.*, 38(1):5, 2025. doi:10.1007/S00145-024-09526-1.

[AP16] Navid Alamati and Chris Peikert. Three's compromised too: Circular insecurity for any cycle length from (Ring-)LWE. In 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*, pages 659–680. Springer, 2016. doi:10.1007/978-3-662-53008-5_23.

[AV21] Adi Akavia and Margarita Vald. On the privacy of protocols based on CPA-secure homomorphic encryption. *IACR Cryptol. ePrint Arch.*, page 803, 2021. URL: https://eprint.iacr.org/2021/803.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). 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 1–20. Springer, 2010. doi:10.1007/978-3-642-14623-7_1.

[BGK11] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In Yuval Ishai, editor, *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*, volume 6597 of *Lecture Notes in Computer Science*, pages 201–218. Springer, 2011. doi:10.1007/978-3-642-19571-6_13.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3):13:1–13:36, 2014. doi:10.1145/2633600.

[BHHO08] Dan Boneh, Shai Halevi, Michael Hamburg, and Rafail Ostrovsky. Circular-secure encryption from Decision Diffie-Hellman. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2008. doi:10.1007/978-3-540-85174-5_7.

[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 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012. doi:10.1007/978-3-642-32009-5_50.

[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. doi:10.1145/3335741.3335762.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014. Preliminary version in FOCS 2011. doi:10.1137/120868669.

[CGGI20] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. *J. Cryptol.*, 33(1):34–91, 2020. doi:10.1007/s00145-019-09319-x.

[CHI+21] Megan Chen, Carmit Hazay, Yuval Ishai, Yuriy Kashnikov, Daniele Micciancio, Tarik Riviere, Abhi Shelat, Muthuramakrishnan Venkitasubramaniam, and Ruihan Wang. Diogenes: Lightweight scalable RSA modulus generation with a dishonest majority. In *42nd IEEE Symposium on Security and Privacy, SP*

*2021, San Francisco, CA, USA, 24-27 May 2021*, pages 590–607. IEEE, 2021. doi:10.1109/SP40001.2021.00025.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 468–497. Springer, 2015. doi:10.1007/978-3-662-46497-7_19.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015. doi:10.1007/978-3-662-46800-5_24.

[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. doi:10.1007/978-3-662-49890-3_12.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, USA, 2009. URL: https://searchworks.stanford.edu/view/8493082.

[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. doi:10.1145/1536414.1536440.

[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. doi:10.1007/978-3-642-14623-7_9.

[GKW17a] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating IND-CPA and circular security for unbounded length key cycles. In Serge Fehr, editor, *Public-Key Cryptography - PKC 2017 - 20th IACR International Conference on Practice and Theory in Public-Key Cryptography, Amsterdam, The Netherlands, March 28-31, 2017, Proceedings, Part I*, volume 10174 of *Lecture Notes in Computer Science*, pages 232–246. Springer, 2017. doi:10.1007/978-3-662-54365-8_10.

[GKW17b] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture*

*Notes in Computer Science*, pages 528–557, 2017. doi:10.1007/978-3-319-56614-6_18.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

[Hal17] Shai Halevi. Homomorphic encryption. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 219–276. Springer International Publishing, 2017. doi:10.1007/978-3-319-57048-8_5.

[HK17] Mohammad Hajiabadi and Bruce M. Kapron. Toward fine-grained blackbox separations between semantic and circular-security notions. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 561–591, 2017. doi:10.1007/978-3-319-56614-6_19.

[HNP+03] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. The impact of decryption failures on the security of NTRU encryption. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 226–246. Springer, 2003. doi:10.1007/978-3-540-45146-4_14.

[KM20] Fuyuki Kitagawa and Takahiro Matsuda. Circular security is complete for KDM security. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 253–285. Springer, 2020. doi:10.1007/978-3-030-64837-4_9.

[KRW15] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, volume 9015 of *Lecture Notes in Computer Science*, pages 378–400. Springer, 2015. doi:10.1007/978-3-662-46497-7_15.

[KW16] Venkata Koppula and Brent Waters. Circular security separations for arbitrary length cycles from LWE. In 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*, pages 681–700. Springer, 2016. doi:10.1007/978-3-662-53008-5_24.

[LM21] Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 648–677. Springer, 2021. doi:10.1007/978-3-030-77870-5_23.

[LMK+23] Yongwoo Lee, Daniele Micciancio, Andrey Kim, Rakyong Choi, Maxim Deryabin, Jieun Eom, and Donghoon Yoo. Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part III*, volume 14006 of *Lecture Notes in Computer Science*, pages 227–256. Springer, 2023. doi:10.1007/978-3-031-30620-4_8.

[Mic22a] Daniele Micciancio. Fully homomorphic encryption 10 years later: definitions and open problems. Presentation at Simons Institute, May 2022. URL: https://www.youtube.com/watch?v=HIJad2TS1iM.

[Mic22b] Daniele Micciancio. Fully homomorphic encryption: Definitional issues and open problems. Presentation at FHE.org, May 2022. URL: https://fhe.org/conferences/conference-2022/resources.html.

[MP21] Daniele Micciancio and Yuriy Polyakov. Bootstrapping in FHEW-like cryptosystems. In *WAHC '21: Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography, Virtual Event, Korea, 15 November 2021*, pages 17–28. WAHC@ACM, 2021. doi:10.1145/3474366.3486924.

[MV24] Daniele Micciancio and Vinod Vaikuntanathan. Sok: Learning with errors, circular security, and fully homomorphic encryption. In Qiang Tang and Vanessa Teague, editors, *Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part IV*, volume 14604 of *Lecture Notes in Computer Science*, pages 291–321. Springer, 2024. doi:10.1007/978-3-031-57728-4_10.

[MW18] Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 3–28. Springer, 2018. doi:10.1007/978-3-319-78381-9_1.