IACR Communications in Cryptology
ISSN 3006-5496, Vol. 2, No. 1, 35 pages.

https://doi.org/10.62056/a69qgyl7s

On Circuit Private, Multikey and Threshold Approximate Homomorphic Encryption

Kamil Kluczniak[1] and Giacomo Santato[2,3]

[1] Independent Researcher, Munich, Germany
[2] CISPA - Helmholtz Center for Information Security, Saarbrücken, Germany
[3] Saarland University, Saarbrücken, Germany

Abstract. Homomorphic encryption for approximate arithmetic allows one to encrypt discretized real/complex numbers and evaluate arithmetic circuits over them. The first scheme, called CKKS, was introduced by Cheon et al. (Asiacrypt 2017) and gained tremendous attention. The enthusiasm for CKKS-type encryption stems from its potential to be used in inference or multiparty computation tasks that do not require an exact output.

A desirable property for homomorphic encryption is circuit privacy, which requires that a ciphertext leaks no information on the computation performed to obtain it. Despite numerous improvements directed toward improving efficiency, the question of circuit privacy for approximate homomorphic encryption remains open.

In this paper, we give the first formal study of circuit privacy for homomorphic encryption over approximate arithmetic. We introduce formal models that allow us to reason about circuit privacy. Then, we show that approximate homomorphic encryption can be made circuit private using tools from differential privacy with appropriately chosen parameters. In particular, we show that by applying an exponential (in the security parameter) Gaussian noise on the evaluated ciphertext, we remove useful information on the circuit from the ciphertext. Crucially, we show that the noise parameter is tight, and taking a lower one leads to an efficient adversary against such a system.

We expand our definitions and analysis to the case of multikey and threshold homomorphic encryption for approximate arithmetic. Such schemes allow users to evaluate a function on their combined inputs and learn the output without leaking anything on the inputs. A special case of multikey and threshold encryption schemes defines a so-called partial decryption algorithm where each user publishes a "masked" version of its secret key, allowing all users to decrypt a ciphertext. Similarly, in this case, we show that applying a proper differentially private mechanism gives us IND-CPA-style security where the adversary additionally gets as input the partial decryptions. This is the first security analysis of approximate homomorphic encryption schemes that consider the knowledge of partial decryptions. We show lower bounds on the differential privacy noise that needs to be applied to retain security. Analogously, in the case of circuit privacy, the noise must be exponential in the security parameter. We conclude by showing the impact of the noise on the precision of CKKS-type schemes.

Keywords: Approximate Homomorphic Encryption · Circuit Privacy

Authors were partially supported by the German Ministry for Education and Research through funding for the project CISPA-Stanford Center for Cybersecurity (Funding number: 16KIS0927).

E-mail: kamil.kluczniak@gmail.com (Kamil Kluczniak), giacomo.santato@cispa.de (Giacomo Santato)

Received: 2025-01-08 Accepted: 2025-03-11

1 Introduction

Fully Homomorphic Encryption (FHE) allows for computations to be performed on encrypted data. A client encrypts a message m and sends the ciphertext to a server, which, given a circuit F, returns a ciphertext that decrypts to $F(m)$. The first fully homomorphic encryption scheme was introduced by Gentry [Gen09b].

FHE has numerous applications in cryptography. Among others, it is used to build private information retrieval [AMBFK15, ALP$^+$21, ACLS18, GH19, CHK22, MW22, HHCG$^+$23], secure function delegation [QWW18] and obfuscation schemes [BDGM20, GP21]. Note, however, that the security of fully homomorphic encryption protects only the encrypted message and, in particular, does not offer any protection for the server's computation. In other words, the ciphertexts that a server returns may completely leak the circuit F.

Circuit privacy, sometimes called function privacy, is a critical property in FHE, where the ciphertext produced by the server, computing a circuit F on encrypted data, should not reveal any information about F, except for the fact that the ciphertext decrypts to $F(m)$. Circuit private FHE enables semi-honest two-party computation with optimal communication, requiring only one round of communication, and its communication complexity is independent of the size of the computation. Furthermore, the ciphertexts produced by the evaluation process can be reused, making FHE suitable for applications such as private set intersection [HFH99, Mea86, CLR17], neural network inference [DGBL$^+$16, CdWM$^+$17, LJLA17, JKLS18, JVC18, BGGJ20, ABSdV19, CDKS19, RSC$^+$19, BGPG20, KS22], analysis of genomic data [KSK$^+$18, KSK$^+$20, BGPG20], and many more.

Multikey and Threshold Homomorphic Encryption. Extensions of homomorphic encryption like multikey [LTV12, CM15, BP16, MW16, CZW17, CCS19, CDKS19, AJJM20] or threshold homomorphic [BGG$^+$18] encryption allow computing on ciphertexts that come from different parties, but require a subset of secret keys of the different parties to decrypt the outcome of the computation. In particular, many variants of these schemes introduce a so-called partial decryption algorithm, in which each party publishes a secret key capable to "remove an encryption layer" from the evaluated ciphertext. Multikey or threshold homomorphic encryption schemes seem to be related to circuit private encryption schemes, as both give us the means to build two-round multiparty computation if the homomorphic encryption satisfies the right security notion. Namely, whether IND-CPA holds against an adversary that is given partial decryptions of non-corrupted parties. In fact, there is a folklore construction ([Sma22]) of a circuit private scheme from a multikey homomorphic encryption scheme for at least two keys.

Homomorphic Encryption for Approximate Arithmetic. While we have seen significant advancements in the practical efficiency of fully homomorphic encryption (FHE) schemes and their circuit private versions, realizing practical instances of neural network inference, data analysis problems, or collaborative learning is still relatively slow. In their seminal paper [CKKS17] Cheon et al. noticed that many of these problems do not require the computation on the encrypted data to be exact. In particular, in many applications, it is sufficient for the homomorphic computation to return an approximation of $F(m)$. As a result, they design a homomorphic encryption scheme with a plaintext space of approximations of real or complex numbers.

Due to its native support of real or complex numbers, CKKS-style schemes are believed to be the most competitive solutions for private machine learning inference problems, data analytics, and even training of machine learning models. The focus of researchers is to make CKKS more efficient and increase its plaintext precision. For example, [CDKS19] introduces an efficient multikey version of [CKKS17]. However, it is not clear whether the application is secure and with respect to which security notion. In particular, [CDKS19] states the standard IND-CPA definition, but in applications of multikey homomorphic encryption, we need to make sure that IND-CPA holds even when given partial decryptions.

On the other hand, we may argue that, running an MPC protocol computing the decryption circuit by inputting the secret keys of all users, can solve the problem. After all, the solution solves the decryption problem in the case of "exact" homomorphic encryption, since the MPC protocol reveals nothing aside from the result of the homomorphic computation. But, unfortunately, in the approximate setting, the decryption gives only an approximation of the exact result, where the approximation error may carry information on the plaintexts of other parties. This means that we need to be careful when trying to apply techniques from the "exact" setting in the approximate setting.

1.1 Our Contributions

In this work, we are the first to formally address the issue of circuit privacy and ciphertext sanitization for homomorphic encryption over approximate arithmetic. Our contributions are as follows.

Formal Definitions. We introduce formal definitions that allow us to reason about circuit privacy for approximate homomorphic encryption. In particular, we expand on some formalism introduced by Li et al. [LMSS22] with regard to the approximate correctness of the computation on ciphertexts. After that, we introduce an indistinguishability-based definition. We note that this is the first indistinguishability-based definition for circuit/function privacy; previously, all definitions were simulation-based, and this also applies in the case of "exact" homomorphic encryption. In particular, the simulation-based definitions imply ours, but ours is more convenient when dealing with approximate homomorphic computation and showing lower bounds.

Circuit Privacy and Lower Bounds. We give an analysis based on Kullback-Leibler divergence, showing that applying a differentially private mechanism with appropriate parameters gives us circuit privacy. In particular, we can use the Gaussian mechanism to "flood" the approximation errors in a ciphertext. Noise flooding is a known technique, and in particular, [LMSS22] analyzed it in the context of $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security [LM21]. Our analysis is inspired by [LMSS22], but we stress that our setting is different in many ways and comes with its own technical challenges which we discuss in the main body of the paper when having the right context. Importantly, we show that the applied noise must be exponential in the security parameter. In particular, we show that, if we apply only a subexponential noise, then there exists an efficient adversary that breaks circuit privacy with non-negligible probability.

Multikey and Threshold Approximate Homomorphic Encryption. We give the first formal study of multikey and threshold homomorphic encryption for approximate arithmetic. There are constructions of such schemes based on CKKS [CDKS19, KKL$^+$23]. However, none of them addresses the relevant security properties. We introduce definitions for indistinguishability security, where an adversary obtains partial decryptions. First, we show that our definitions are meaningful, and multikey and threshold homomorphic encryption satisfying our security notion imply homomorphic encryption satisfying our notion of circuit privacy. Then, we give a Kullback-Leibler-divergence-based proof that applying the Gaussian differential-privacy mechanism in partial decryptions with exponential Gaussian noise is sufficient to satisfy our security notion. On the downside, we show that the applied noise parameters are tight, and using smaller parameters leads to the break of the relevant security property. Our result in this manner is especially relevant due to the following.

- There is a folklore belief that achieving circuit privacy through multikey (F)HE leads to better parameters than sanitizing a single-key homomorphic evaluation of the function. Indeed, multikey (F)HE can be used to construct a circuit-private (F)HE scheme. The core idea is that the server encrypts the circuit using its own key, while the client encrypts the query with its respective key. The server then evaluates a universal circuit over both ciphertexts and returns a partially decrypted result of the evaluated ciphertext to the client. If multikey or threshold evaluation with partial decryptions yields a secure MPC protocol, then this approach appears to be sound. This claim is supported, for instance, by the construction presented in [MW16].

  However, we argue that the required parameters remain the same as those dictated by the noise flooding of a single-key evaluation. The misconception that multikey (F)HE improves parameter efficiency likely stems from the fact that, in this construction, the circuit itself is encrypted, creating the impression that it is inherently "secured" in some way. This perspective often overlooks the critical role of noise flooding in the partial decryption phase. By omitting explicit discussion of noise flooding, this narrative may encourage the use of schemes with smaller noise parameters.

  Our analysis shows that encrypting the circuit does not significantly reduce the required flooding noise compared to simply sanitizing a single-key homomorphic encryption. Furthermore, we conclude that circuit privacy is primarily ensured by noise flooding with exponential noise, rather than by the mere fact that computation is performed on a multikey ciphertext. Thus, we resolve the “myth” that multikey homomorphic computation alone guarantees circuit privacy and instead emphasize that it is the specific way partial decryptions are implemented that gives us this property.

- Papers introducing multikey constructions often present parameters that consider only the correctness of the evaluation and the security of the underlying LWE encryption. Specifically, the paper introducing MK-CKKS [CDKS19] simply states to "set the noise flood distribution to have a larger variance than the standard error distribution." We believe this guidance could lead practitioners to adopt overly optimistic parameters compared to those required in practice or those commonly used in the single-key setting. By providing a formal analysis and establishing a lower bound on the noise, we aim to encourage practitioners to carefully consider their parameter choices.

1.2 Related Work on Circuit Privacy and Multikey Homomorphic Encryption

Circuit privacy, or sometimes called function or server privacy, was studied before the first secure fully homomorphic encryption schemes were proposed [IP07, Gen09a]. There are two ways to build a circuit private homomorphic encryption scheme. The first is to use a multiparty computation protocol to compute the decryption circuit on the ciphertext [IP07, GHV10, CO17]. Another way is to sanitize a ciphertext from any information on the circuit. In other words, we apply a random process to the ciphertext in order to make its distribution independent of the circuit. Current approaches to sanitize a ciphertext include noise flooding [Gen09a], repeated bootstrapping [DS16], and re-randomizing computation [BDPMW16, Klu22]. Note that all of these mechanisms apply to “exact” homomorphic encryption. In particular, there is no formal treatment on circuit privacy for approximate homomorphic encryption [CKKS17].

Multikey fully homomorphic encryption was first introduced in [LTV12], and the related concept of threshold homomorphic encryption was introduced in [BGG$^+$18]. For the case of approximate arithmetic, [CDKS19] gave an efficient construction for the multikey setting based on [CKKS17]. They propose to use noise flooding for partially decrypting ciphertexts. However, there is no security proof or even formal definition of what it means for such encryption scheme to be secure aside of IND-CPA security that does not consider adversaries with knowledge of partial decryptions. Mukherjee and Wichs [MW16] define a simulator for partial decryptions in the setting of "exact" GSW [GSW13] encryption to capture the security properties needed to build multiparty computation protocols. Note that such a definition often requires that the homomorphic encryption scheme evaluates the exact circuit, as opposed to approximate. Unfortunately, it is not clear whether we can use such definitions for approximate homomorphic encryption.

2 Preliminaries

We denote an n dimensional column vector as $[f(.,i)]_{i=1}^{n}$, where $f(.,i)$ defines the i-th coordinate. For brevity, we will also denote as $[n]$ the vector $[i]_{i=1}^{n}$.For a random variable $x \in \mathbb{Z}$ we denote as $\mathsf{Var}(x)$ the variance of x, as $\mathsf{stddev}(x)$ its standard deviation and as $\mathsf{E}(x)$ its expectation. By $\mathsf{Ham}(\vec{a})$ we denote the Hamming weight of the vector $\vec{a}$, i.e., the number of non-zero coordinates of $\vec{a}$.

We say that an algorithm is **PPT** if it is a probabilistic polynomial-time algorithm. We denote any polynomial as $\mathtt{poly}(.)$. We denote as $\mathtt{negl}(\lambda)$ a negligible function in $\lambda \in \mathbb{N}$. That is, for any positive polynomial $\mathtt{poly}(.)$ there exists $c \in \mathbb{N}$ such that for all $\lambda \geq c$ we have $\mathtt{negl}(\lambda) \leq \frac{1}{\mathtt{poly}(\lambda)}$. Given two distributions X, Y over a finite domain D, their statistical distance is defined as $\Delta(X,Y) = \frac{1}{2}\sum_{v \in D} |X(v) - Y(v)|$. We say that two distributions are statistically close if their statistical distance is negligible.

Usually, we assume that a probabilistic algorithm $\mathsf{Alg}(x)$ chooses its random coins internally. However, sometimes we write $\mathsf{Alg}(x;r)$ to denote that the random coins $r \xleftarrow{\$} \mathcal{U}$ are used as a seed for Alg, and $\mathsf{Alg}(x;r)$ is deterministic.

2.1 Homomorphic Encryption

We review the definition of Homomorphic Encryption in the public key setting with a particular focus on classical and (static) approximate correctness.

Definition 1 (Homomorphic Encryption)**.** We define a homomorphic encryption scheme HE for a class of circuits $\mathcal{L}$ as a tuple of four algorithms $\mathsf{HE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{Dec})$ with the following syntax.

$\mathtt{KeyGen}(\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** Given a security parameter λ, returns a public key pk and a secret key sk.

$\mathtt{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$**:** Given a public key pk and a message m, returns a ciphertext ct.

$\mathtt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k) \to \mathsf{ct}$**:** Given a public key pk, a circuit $C \in \mathcal{L}$ and ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$, returns a ciphertext ct.

$\mathtt{Dec}(\mathsf{sk}, \mathsf{ct}) \to m$**:** Given a secret key sk and a ciphertext ct, returns a message m.

We denote as $\mathcal{M}$ the message space, $\mathcal{C}$ the ciphertext space and $\mathcal{L}$ the class of circuits.

In this paper, we consider different notions of correctness. In particular, we consider the classical correctness definition and approximate correctness that was recently introduced in [LMSS22] to reason about approximate homomorphic encryption schemes.

Definition 2 (Correctness)**.** We say that a homomorphic encryption scheme $\mathsf{HE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ is correct if for all $C \in \mathcal{L}$, all $m_1, \ldots, m_k \in \mathcal{M}$, all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{KeyGen}(\lambda)$ and for all $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ such that $m_i = \texttt{Dec}(\mathsf{sk}, \mathsf{ct}_i)$ for $i \in [k]$, we have that

$$\Pr[\texttt{Dec}(\mathsf{sk}, \texttt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)) \neq C(m_1, \ldots, m_k)] \leq \texttt{negl}(\lambda).$$

Below we recall the definition of approximate correctness from [LMSS22]. First, however, we need to formally define the notion of a ciphertext error.

Definition 3 (Ciphertext Error)**.** Let $\mathsf{HE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ be an homomorphic encryption scheme with message space $\mathcal{M}$. Furthermore, let $\mathcal{M}$ be a normed space with norm $||\cdot|| : \mathcal{M} \mapsto \mathbb{R}_{\geq 0}$. For all public/secret key pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{KeyGen}(\lambda)$, any ciphertext $\mathsf{ct} \in \mathcal{C}$ and message $m \in \mathcal{M}$ the ciphertext error is defined as

$$\texttt{Error}(\mathsf{sk}, \mathsf{ct}, m) = ||\texttt{Dec}(\mathsf{sk}, \mathsf{ct}) - m||.$$

We can now introduce the approximate correctness notion for approximate HE schemes.

Definition 4 (Approximate Correctness [LMSS22])**.** Let $\mathsf{HE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ be a homomorphic encryption scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$ that is a normed space with norm $|| \cdot || : \widetilde{\mathcal{M}} \mapsto \mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ be the class of circuits, $\mathcal{L}_k \subseteq \mathcal{L}$ be the subset of circuits with k input wires, and let $\texttt{Estimate} : \bigsqcup_{k \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}^k_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ be an efficiently computable function. We call HE an approximate homomorphic encryption scheme (w.r.t. $\texttt{Estimate}$) if for all $k \in \mathbb{N}$, for all $C \in \mathcal{L}_k$, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{KeyGen}(\lambda)$, if $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ and $m_1, \ldots, m_k$ are such that $\texttt{Error}(\mathsf{sk}, \mathsf{ct}_i, m_i) \leq t_i$, for some $t_1, \ldots, t_k \in \mathbb{R}_{\geq 0}$, then

$$\texttt{Error}(\mathsf{sk}, \texttt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k), C(m_1, \ldots, m_k)) \leq \texttt{Estimate}(C, t_1, \ldots, t_k).$$

To compute $\texttt{Estimate}$, we only need the circuit C and upper bounds t_i on the ciphertext errors. This means that the function is publicly and efficiently computable without needing a secret key.

To keep track of the errors when computing on encrypted data, we associate a tag with every ciphertext. In particular, we define a *tagged ciphertext* $\mathsf{ct} = (\ldots, t)$ where $t \in \mathbb{R}_{\geq 0}$ is an extension of an ordinary ciphertext that also stores t, a *provable upper bound* estimate of the ciphertext error. The noise bound is set to t_{fresh} by $\texttt{Enc}$ when a ciphertext ct is created. After that, the value of $\mathsf{ct}.t$ is updated using $\texttt{Estimate}$ every time that a circuit is homomorphically evaluated on ct.

We also recall the definition of IND-CPA security for HE schemes.

Definition 5 (IND-CPA security game)**.** Let $\mathsf{HE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{Dec})$ be a homomorphic encryption scheme. We define the IND-CPA game as the experiment $\mathsf{Exp}_b^{\mathsf{IND\text{-}CPA}}$, where $b \in \{0, 1\}$ is a bit and $\mathcal{A}$ is an adversary. The experiment is defined as follow:

$$\begin{aligned} \mathsf{Exp}_b^{\mathsf{IND\text{-}CPA}}[\mathcal{A}](\lambda) :\ & (\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{KeyGen}(\lambda), \\ & b' \leftarrow \mathcal{A}^{\mathsf{E}^b(\mathsf{pk}, \cdot, \cdot)}(\mathsf{pk}), \\ & \textbf{return } b', \end{aligned}$$

where the adversary has access to an encryption oracle $\mathsf{E}^b(\mathsf{pk}, \cdot, \cdot)$ that takes as input $m_0, m_1 \in \mathcal{M}$ and returns $\texttt{Enc}(\mathsf{pk}, m_b)$.

2.2 The CKKS Approximate HE Scheme

We recall the definition of the CKKS approximate HE scheme following the notation used in [LMSS22]. A more detailed description of CKKS can be found in [CKKS17].

We denote by $\mathcal{R}$ the ring $\mathbb{Z}[X]/(\Phi_N(X))$ and by $\mathcal{R}_Q$ the ring $\mathbb{Z}_Q[X]/(\Phi_N(X))$, where $\mathbb{Z}_Q$ is the ring of integers modulo Q and Φ_N is the N-th cyclotomic polynomial.

$\mathsf{CKKS}.\texttt{KeyGen}(\lambda)$**:** Given the security parameter λ choose $p \in \mathbb{N}$ and $Q \in \mathbb{N}$, the ring $\mathcal{R}$ and the noise distribution χ. Sample $s \in \mathcal{R}_{pQ}$ by sampling each coefficient uniformly from $\{-1, 0, 1\}$ and set $\mathsf{sk} \leftarrow s$. Sample $\mathsf{pk}.a \overset{\$}{\leftarrow} \mathcal{R}_Q$, $e \overset{\$}{\leftarrow} \chi$ and compute $\mathsf{pk}.b \leftarrow -\mathsf{sk} \cdot \mathsf{pk}.a + e$. Then sample $\mathsf{pk}.a' \overset{\$}{\leftarrow} \mathcal{R}_Q$, $e' \overset{\$}{\leftarrow} \chi$ and compute $\mathsf{pk}.b' \leftarrow -\mathsf{sk} \cdot \mathsf{pk}.a' + e' + \mathsf{sk}^2$.

$\mathsf{CKKS}.\texttt{Enc}(\mathsf{pk}, m \in \mathcal{R}_Q)$**:** Choose $r \in \mathcal{R}$ such that every coefficient (chosen independently) has probability 1/4 to be 1 and -1, and probability 1/2 to be 0. Sample $e_0, e_1 \leftarrow \chi$. Set $\mathsf{ct}.a \leftarrow r\mathsf{pk}.a + e_0$, $\mathsf{ct}.b \leftarrow r\mathsf{pk}.b + e_1 + m$ and return ct.

$\mathsf{CKKS}.\texttt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \dots, \mathsf{ct}_k)$ **:** The algorithm evaluates the arithmetic circuit C by means of addition and multiplication:

> $\mathsf{CKKS}.\texttt{Add}(\mathsf{pk}, \mathsf{ct}_0, \mathsf{ct}_1 \in \mathcal{R}_Q)$**:** Set $\mathsf{ct}.a \leftarrow \mathsf{ct}_0.a + \mathsf{ct}_1.a$, $\mathsf{ct}.b \leftarrow \mathsf{ct}_0.b + \mathsf{ct}_1.b$ and return ct.
>
> $\mathsf{CKKS}.\texttt{Mul}(\mathsf{pk}, \mathsf{ct}_0, \mathsf{ct}_1 \in \mathcal{R}_Q)$**:** Set $\mathsf{ct}.b \leftarrow \mathsf{ct}_0.b \cdot \mathsf{ct}_1.b + \lfloor (\mathsf{ct}_0.a \cdot \mathsf{ct}_1.a \cdot \mathsf{pk}.b')/p \rceil$, and $\mathsf{ct}.a \leftarrow \mathsf{ct}_0.a \cdot \mathsf{ct}_1.b + \mathsf{ct}_1.a \cdot \mathsf{ct}_0.b + \lfloor (\mathsf{ct}_0.a \cdot \mathsf{ct}_1.a \cdot \mathsf{pk}.a')/p \rceil$. Return ct.

$\mathsf{CKKS}.\texttt{Dec}(\mathsf{sk}, \mathsf{ct})$**:** Return $\mathsf{ct}.b + \mathsf{ct}.a \cdot \mathsf{sk}$.

When mentioning a noiseless CKKS encryption $\texttt{Enc}_\mathsf{n}$, we refer to a ciphertext obtained as $\mathsf{CKKS}.\texttt{Enc}(\mathsf{pk}, m)$ where $e_0 = e_1 = 0$. This is not a secure encryption but will help us when describing the rerandomization process in the proof of Theorem 5.

We also recall the basic expressions of noise growth during addition and multiplication in CKKS.

Lemma 1 (Lemma 3 of [CKKS17])**.** *Let* $\mathsf{ct}_i = \mathsf{CKKS}.\texttt{Enc}(\mathsf{pk}, m_i)$ *for* $i \in \{0, 1\}$ *and their ciphertext error be, respectively,* $\texttt{Error}(\mathsf{sk}, ct_i, m_i) = e_i$*. The ciphertext error of the sum of both ciphertexts is equal to* $e_0 + e_1$ *and the ciphertext error their product is equal to* $m_0e_1 + m_1e_0 + e_0e_1 + e_{\mathsf{mult}}$*, where the term* e_{mult} *depends on the parameters of the scheme and on the two ciphertexts* $\mathsf{ct}_0, \mathsf{ct}_1$*.*

We now give a brief explanation on how the tagged ciphertext and the $\texttt{Estimate}$ function are handled by the algorithms of the CKKS scheme. $\mathsf{CKKS}.\texttt{Enc}$ assigns to the returned ciphertext an upper bound of the ciphertext error for fresh encryptions. $\mathsf{CKKS}.\texttt{Add}$ and $\mathsf{CKKS}.\texttt{Mul}$ follow the noise growth rules of Lemma 1 to assign to the returned ciphertext a noise estimate. More generally, when homomorphic evaluating a circuit C in CKKS by computing $\texttt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \dots, \mathsf{ct}_k)$, it is always possible to publicly compute the resulting noise estimate by combining the two noise growth rules for sum and product using as an input only the description of C and the noise estimates on the input ciphertexts.

2.3 Probability

A probability ensemble $(\mathcal{P}_\theta)_\theta$ is a family of probability distributions parameterized by a variable θ. The KL Divergence is a useful tool to handle probability distributions. In particular, it gives us a way to understand how close (or far) are two distributions from each other.

Definition 6 (KL divergence)**.** Let $\mathcal{P}$ and $\mathcal{Q}$ be two probability distributions with common support X. The Kullback-Leibler Divergence between $\mathcal{P}$ and $\mathcal{Q}$ is $D(\mathcal{P}||\mathcal{Q}) := \sum_{x \in X} \Pr[\mathcal{P} = x] \ln\left(\frac{\Pr[\mathcal{P}=x]}{\Pr[\mathcal{Q}=x]}\right)$.

Lemma 2 (Subadditivity of KL divergence for Joint Distributions, Theorem 2.2 of [PW25])**.** *If* $(\mathcal{X}_0, \mathcal{X}_1)$ *and* $(\mathcal{Y}_0, \mathcal{Y}_1)$ *are pairs of (possibly dependent) random variables, then*

$$D((\mathcal{X}_0, \mathcal{X}_1)||(\mathcal{Y}_0, \mathcal{Y}_1)) \leq \max_x D((\mathcal{X}_1|x)||(\mathcal{Y}_1||x)) + D(\mathcal{X}_0, \mathcal{Y}_0)$$

Computing the advantages of adversaries from Subsection 4.3 and from Subsection 5.4 will require the following inequality about the total variation distance between two Gaussian distributions.

Definition 7 (Gaussian distribution)**.** Let $\mu \in \mathbb{R}$ and $\sigma > 0$. The (continuous) Gaussian of parameters μ, σ (written $\mathcal{N}(\mu, \sigma^2)$) is the probability distribution supported on $\mathbb{R}$ with p.m.f. $p(x) \propto \exp(-(x-\mu)^2/2\sigma^2)$.

Definition 8 (Discrete Gaussian distribution)**.** Let $n \in \mathbb{Z}$, $n \geq 1$, $\mu \in \mathbb{Z}^n$ and $\sigma > 0$. The discrete Gaussian of parameters μ, σ (written $\mathcal{N}_{\mathbb{Z}^n}(\mu, \sigma^2)$) is the probability distribution supported on $\mathbb{Z}^n$ with p.m.f. $p(x) \propto \exp(-\|x-\mu\|^2/2\sigma^2)$.

Theorem 1 (Theorem 1.3 of [DMR18])**.** *Let $\sigma_0, \sigma_1 > 0$. Then*

$$\Delta(\mathcal{N}(\mu_0, \sigma_0^2), \mathcal{N}(\mu_1, \sigma_1^2)) \geq \frac{1}{200} \min\{1, \frac{40|\mu_0 - \mu_1|}{\sigma_0}\}.$$

2.4 KL Differential Privacy

In [LMSS22], Li et al. introduce the new notion of *Norm Rényi Differential Privacy* by generalizing the notion of Rényi differential privacy [Mir17] to different norms. This innovative technique aims to address the primary technical challenges encountered when applying differential privacy in environments with arbitrary norms. Specifically, within the context of Differential Privacy, the concept of "adjacent" values is commonly assessed using the Hamming norm, whereas Approximate HE revolves around Euclidean and Infinity norms. In this paper, we will focus exclusively on the specific instance of this definition that uses KL divergence.

Definition 9 (Norm KL Diff. Privacy, Definition 14 of [LMSS22])**.** For $t \in \mathbb{R}_{\geq 0}$, let $M_t : B \to C$ be a family of randomized algorithms, where B is a normed space with norm $|| \cdot || : B \to \mathbb{R}_{\geq 0}$. Let $\rho \in \mathbb{R}$ be a privacy bound. We say that the family M_t is ρ-KL differentially private (ρ-KLDP) if, for all $x, x' \in B$ with $||x - x'|| \leq t$,

$$D(M_t(x) || M_t(x')) \leq \rho.$$

Definition 10. Let $\rho > 0$ and $n \in \mathbb{N}$. Define the (discrete) Gaussian Mechanism $M_t : \mathbb{Z}^n \to \mathbb{Z}^n$ be the mechanism that, on input $x \in \mathbb{Z}^n$, outputs a sample from $\mathcal{N}_{\mathbb{Z}^n}(x, \frac{t^2}{2\rho} I_n)$.

Theorem 2 (Lemma 6, [LMSS22])**.** *For any $\rho > 0$, $n \in \mathbb{N}$, the Gaussian mechanism is ρ-KLDP.*

2.5 Bit security

One of the original motivations of this work was to extend the security analysis beyond the use of statistical distance in the hope of providing tighter noise bounds and improving the parameters. Using Rényi divergence when studying decisional problems is an important technique introduced in [BLR$^+$18], and that has been proved useful in lattice-based cryptography to obtain a tighter security analysis and to improve the parameters. Finally, we choose to analyze bit security due to the technical synergies with KL divergence (Theorem 3) and KL Differential Privacy (Theorem 4).

We briefly recall the notion of bit security from [MW18].

Definition 11 (Indistinguishability Game)**.** Let $\{\mathcal{D}_\theta^0\}$ and $\{\mathcal{D}_\theta^1\}$ be two distributions ensembles. The indistinguishability game is defined as follows: the challenger C chooses $b \leftarrow \mathcal{U}(\{0,1\})$. At any time after that, the adversary $\mathcal{A}$ may send (adaptively chosen) query strings θ_i to C and obtain samples $c_i \leftarrow \mathcal{D}_{\theta_i}^b$. The goal of the adversary is to output $b' = b$.

Definition 12 (Bit Security)**.** For any adversary $\mathcal{A}$ playing an indistinguishability game $\mathcal{G}$, we define its

- output probability as $\alpha^{\mathcal{A}} = \Pr[\mathcal{A} \neq \perp]$ and its
- conditional success probability as $\beta^{\mathcal{A}} = \Pr[b' = b | \mathcal{A} \neq \perp]$.

where the probabilities are taken over the randomness of the entire indistinguishability game (including the internal randomness of $\mathcal{A}$). We also define $\mathcal{A}$'s

- conditional distinguishing advantage as $\delta^{\mathcal{A}} = 2\beta^{\mathcal{A}} - 1$ and
- the advantage of $\mathcal{A}$ as $\mathsf{adv}^{\mathcal{A}} = \alpha^{\mathcal{A}}(\delta^{\mathcal{A}})^2$.

The bit security of the indistinguishability game is $\min_{\mathcal{A}} \log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}}$, where $T(\mathcal{A})$ is the running time of $\mathcal{A}$.

We can use bit security on the indistinguishability game from Definition 5.

Definition 13 ($\mathsf{IND\text{-}CPA}$-security)**.** A homomorphic encryption scheme HE is said to be λ-bit $\mathsf{IND\text{-}CPA}$-secure if, for any adversary $\mathcal{A}$ in the $\mathsf{IND\text{-}CPA}$ security game, we have that $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}}$, where $\mathsf{adv}^{\mathcal{A}}$ is defined as in Definition 12.

Theorem 3. *[Theorem 1 of [LMSS22]] Let $\mathcal{G}^{\mathcal{P}}$ be an indistinguishability game with black-box access to a probability ensemble $\mathcal{P}_\theta$. If $\mathcal{G}^{\mathcal{P}_\theta}$ is k-bit secure, and also* $\max_\theta D(\mathcal{P}_\theta || \mathcal{Q}_\theta) \leq 2^{-k+1}$*, then $\mathcal{G}^{\mathcal{Q}_\theta}$ is $(k-8)$-bit secure.*

Theorem 4. *[Lemma 5 of [LMSS22]] Let $\mathcal{G}$ be the indistinguishability game instantiated with distribution ensembles $\{\mathcal{X}_\theta\}_\theta$ and $\{\mathcal{Y}_\theta\}_\theta$, where $\theta \in \Theta$. Let $q \in \mathbb{N}$. Then, for any (potentially computationally unbounded) adversary $\mathcal{A}$ making at most q queries to its oracle, we have that*

$$\mathsf{adv}^{\mathcal{A}} \leq \frac{q}{2} \max_{\theta \in \Theta} D(\mathcal{X}_\theta || \mathcal{Y}_\theta).$$

3 Defining Circuit Privacy for Approximate HE

In this section, we recall the (classic) simulation-based definition of circuit privacy introduced by Gentry [Gen09a]. Then we give our relaxed indistinguishability definition.

We start by stating Gentry's [Gen09a] simulation-based definition below.

Definition 14 (Circuit Privacy)**.** A homomorphic encryption scheme HE for a class of circuits $\mathcal{L}$ is said to be circuit private if there exists a **PPT** simulator $\mathtt{Sim}$ such that, for any $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ valid ciphertexts,

$$\Delta(\mathtt{Sim}(\mathsf{pk}, m_{\mathsf{out}}), \mathtt{Eval}(\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_k, C)) \leq \mathtt{negl}(\lambda),$$

where $C \in \mathcal{L}$, $[m_i \leftarrow \mathtt{Dec}(\mathsf{sk}, \mathsf{ct}_i)]_{i=1}^k$, $m_{\mathsf{out}} \leftarrow C(m_1, \ldots, m_k)$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathtt{KeyGen}(\lambda)$.

Definition 14 gives us a very strong privacy guarantee. In particular, the simulator should produce a ciphertext that is statistically indistinguishable from the homomorphic computation while obtaining only the outcome of an evaluation. This means that the evaluation process reveals no information on the circuit aside from the output of the circuit evaluation. On the other hand, as we discussed in Section 2, homomorphic encryption for approximate arithmetic introduces errors to the outcome of the evaluation. Consequently, the output of the computation may depend somehow on the evaluated circuit. For instance, already the magnitude of the error reveals the size of the circuit or its topology. Finally, note that the simulation definition implicitly induces a requirement that the homomorphic computation is exact. In particular, using $m_{\mathsf{out}} \leftarrow C(m_1, \ldots, m_k)$ to simulate a ciphertext completely ignores the fact that the homomorphic evaluation is approximate, and that the resulting ciphertext $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathtt{Eval}(\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ is now encrypting the message $\mathtt{Dec}(\mathsf{sk}, \mathsf{ct}_{\mathsf{res}})$, that is different from m_{out}. Unfortunately, due to this correctness requirement, we cannot use such a definition to reason about circuit privacy for approximate homomorphic encryption. This state of affairs motivates us to state a relaxed definition of circuit privacy which is sufficient for many applications and gives us a framework to analyze circuit privacy in the case of approximate homomorphic encryption.

We give our definition below.

Definition 15 (Indistinguishability against Chosen Function Attack)**.** Let $\mathsf{HE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{Dec})$ be a homomorphic encryption scheme for circuits in $\mathcal{L}$. We define the experiment $\mathtt{Exp}_b^{\mathsf{IND\text{-}CFA}}[\mathcal{A}]$, where $b \in \{0, 1\}$ is a bit and $\mathcal{A}$ is an adversary. The experiment is defined as follow:

$$\begin{aligned} \mathtt{Exp}_b^{\mathsf{IND\text{-}CFA}}[\mathcal{A}](\lambda): \quad & r, r_1, \ldots, r_n \overset{\$}{\leftarrow} \mathcal{U}, \\ & m_1, \ldots, m_n, C_0, C_1, \mathsf{st} \leftarrow \mathcal{A}(\lambda, r, r_1, \ldots, r_n), \\ & (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KeyGen}(\lambda; r), \\ & [\mathsf{ct}_i \leftarrow \mathtt{Enc}(\mathsf{pk}, m_i; r_i)]_{i=1}^n, \\ & \mathsf{ct} \leftarrow \mathtt{Eval}(\mathsf{pk}, C_b, \mathsf{ct}_1, \ldots, \mathsf{ct}_n), \\ & b' \leftarrow \mathcal{A}(\mathsf{st}, \mathsf{ct}), \\ & \textbf{return } b'. \end{aligned}$$

where $C_0, C_1 \in \mathcal{L}$ and $C_0(m_1, \ldots, m_n) = C_1(m_1, \ldots, m_n)$. The scheme HE is said to be λ-bit $\mathsf{IND\text{-}CFA}$-secure if, for any adversary $\mathcal{A}$, we have that $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}}$, where $\mathsf{adv}^{\mathcal{A}}$ is defined as in Definition 12.

In this definition, the adversary receives the random coins used by the $\mathtt{KeyGen}$ and the $\mathtt{Enc}$ algorithms. Therefore, sk,pk and the ct_i are honestly generated, and the adversary can compute sk and pk.

Remark 1. The acronym IND-CFA bears a resemblance to the acronym IND-CPA. We want to emphasize that they are two different security notions. In particular, IND-CFA can be seen as a computational version of the Circuit Privacy as defined in Definition 14.

4 Circuit Privacy in CKKS

In Subsection 4.1 we present a modification of the CKKS approximate homomorphic encryption scheme that satisfies indistinguishability circuit privacy as given by Definition 15. In particular, we show that re-randomized CKKS ciphertexts are circuit private when we apply an appropriate differential privacy mechanism that floods the ciphertexts noise with an exponential Gaussian sample. In Subsection 4.2 we show how to choose parameters for the differential privacy mechanism for the class of circuits that consists of multivariate polynomials of bounded degree. Finally, in Subsection 4.3, we show that the parameters are tight. Namely, the Gaussian noise must be exponential in the security parameter, and a significantly lower noise parameter leads to an efficient adversary against IND-CFA-security.

4.1 IND-CFA-secure CKKS

To get circuit privacy we modify the CKKS.Eval algorithm, which we describe at Algorithm 1. The main idea is to post-process the ciphertext after evaluation. Namely, we re-randomize the ciphertext with a freshly sampled encryption of zero, and we apply a proper differential privacy mechanism.

Note that to run the discrete Gaussian mechanism we need to redefine the Estimate algorithm such that it outputs an upper bound which depends on a class of circuits instead of just the noise upper bound for a given circuit. Concretely we estimate the noise tag as $\max_{C\in\mathcal{L}}\{\mathtt{Estimate}(C, t_{\mathsf{fresh}}, \dots, t_{\mathsf{fresh}})\}$ for a class of circuits $\mathcal{L}$; we refer to this noise estimate as $T_{\max}$.

Algorithm 1: The modified CKKS evaluation $\mathtt{Eval}_{\mathcal{L}}$

```
Data: A public key pk, circuit $C \in \mathcal{L}$, a vector of ciphertexts $\mathsf{ct}_1, \dots, \mathsf{ct}_k$.
begin
    $\mathsf{ct} \leftarrow \mathtt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \dots, \mathsf{ct}_k)$ ;
    $\mathsf{ct}.t \leftarrow \max_{D\in\mathcal{L}}\{\mathtt{Estimate}(D, \mathsf{ct}_1.t, \dots, \mathsf{ct}_k.t)\}$ ;
    $\mathsf{ct} \leftarrow \mathsf{CKKS}.\mathtt{Add}(\mathsf{pk}, \mathsf{ct}, \mathtt{Enc}(\mathsf{pk}, 0))$ ;
    $\mathsf{ct}.b \leftarrow M_{\mathsf{ct}.t}(\mathsf{ct}.b)$ ;
    return ct ;
```

Theorem 5. *Let* $\mathsf{CKKS} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{Dec})$ *be the CKKS approximate encryption scheme, with the normed plaintext space* $\mathcal{R}$ *and estimate function* Estimate. *Let* M_t *be a* ρ*-KLDP mechanism on* $\mathcal{R}$ *where* $\rho \leq 2^{-\lambda-7}$. *Then,* CKKS *with the modified* $\mathtt{Eval}_{\mathcal{L}}$ *given by Algorithm 1 is* λ*-bit secure in the* IND-CFA *game for the circuit space* $\mathcal{L}$.

Proof. We give a brief overview of the structure of the proof. First, we construct a new $(\lambda + 8)$-bit secure indistinguishability game. After that, we consider the output to any adversary's query in this game and in the IND-CFA game, and we study the KL-divergence between them. In order to bound the KL-divergence, we compute the difference of some entries in the outputs, upper-bound their norm, and then use subadditivity (Lemma 2) and differential privacy (Definition 9). Finally, once we have obtained a bound on the KL-divergence, we can link the bit security of the two games and conclude the proof.

We start by describing the two indistinguishability games.

- $\mathcal{G}_0$: the CKKS scheme with the evaluation algorithm given by Algorithm 1 in the IND-CFA game with circuit space $\mathcal{L}$.
- $\mathcal{G}_1$: the original CKKS scheme in a variant of the IND-CFA game where the challenger returns a fresh noiseless encryption (that we denote as $\mathtt{Enc}_{\mathsf{n}}$) of the result $m_{\mathsf{res}} = C_0(m_1, \dots, m_k) = C_1(m_1, \dots, m_k)$. Furthermore, $\mathsf{ct}.b$ is post-processed with a differential privacy mechanism that uses the same noise tag obtained in the game $\mathcal{G}_0$. More formally, we consider the following experiment:

$$\begin{aligned}
\mathtt{Exp}_b^{\mathcal{G}_1}[\mathcal{A}](\lambda) : \; & r \xleftarrow{\$} \mathcal{U}, \\
& (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KeyGen}(\lambda; r), \\
& m_1, \dots, m_k, C_0, C_1, \mathsf{st} \leftarrow \mathcal{A}(\lambda; r), \\
& m_{\mathsf{res}} \leftarrow C_0(m_1, \dots, m_n), \\
& \mathsf{ct} \leftarrow \mathtt{Enc}_{\mathsf{n}}(\mathsf{pk}, m_{\mathsf{res}}), \\
& \mathsf{ct}.t \leftarrow \max_{D\in\mathcal{L}}\{\mathtt{Estimate}(D, t_{\mathsf{fresh}}, \dots, t_{\mathsf{fresh}})\} + t_{\mathsf{fresh}}, \\
& \mathsf{ct} \leftarrow (\mathsf{ct}.a, M_{\mathsf{ct}.t}(\mathsf{ct}.b)), \\
& b' \leftarrow \mathcal{A}(\mathsf{st}, \mathsf{ct}), \\
& \textbf{return } b'.
\end{aligned}$$

We want to compare these two games and, in particular, analyze the ciphertext the adversary receives from the challenger in each game. In $\mathcal{G}_0$, the ciphertext is obtained by actually homomorphically evaluating the chosen circuit and then by post-processing it with the re-randomization and with a differential privacy mechanism on the second component. In $\mathcal{G}_1$, the ciphertext is simulated by encrypting the plaintext result of the evaluation, without performing any homomorphic evaluation. We will refer to the ciphertexts returned by $\mathcal{G}_0$ and $\mathcal{G}_1$, respectively, as ct_0 and ct_1.

While assuming that $\mathsf{ct}_0.a = \mathsf{ct}_1.a = a$, we compute the norm of the difference between $\mathsf{ct}_0.b$ and $\mathsf{ct}_1.b$, which are the first components of the ciphertexts before applying the differential privacy mechanism.

$$\begin{aligned}\|\mathsf{ct}_0.b - \mathsf{ct}_1.b\| &= \|(\mathsf{ct}_0.b + a \cdot \mathsf{sk}) - (\mathsf{ct}_1.b + a \cdot \mathsf{sk})\| \\ &= \|(m + e_0) - (m)\| = \|e_0\|,\end{aligned}$$

where e_0 is the original error of the ciphertext ct_0, before post-processing. By definition of approximate correctness of CKKS we know that the error e_0 is smaller than the ciphertext noise tag $\mathsf{ct}_0.t$. Therefore,

$$\|\mathsf{ct}_0.b - \mathsf{ct}_1.b\| = \|e_0\| \leq \mathsf{ct}.t$$

Since we were able to bound $\|\mathsf{ct}_0.b - \mathsf{ct}_1.b\|$ with $\mathsf{ct}.t$ we can now use Definition 9 to bound their KL divergence after post-processing

$$D\left((M_{\mathsf{ct}.t}(\mathsf{ct}_0.b)|\mathsf{ct}_0.a = a) \;||\; (M_{\mathsf{ct}.t}(\mathsf{ct}_1.b)|\mathsf{ct}_1.a = a)\right) \leq \rho.$$

We now use Lemma 2 to obtain the following inequality.

$$\begin{aligned}&D(M_{\mathsf{ct}.t}(\mathsf{ct}_0.b), \mathsf{ct}_0.a || M_{\mathsf{ct}.t}(\mathsf{ct}_1.b), \mathsf{ct}_1.a) \\ &\leq \max_a D(M_{\mathsf{ct}.t}(\mathsf{ct}_0.b)|\mathsf{ct}_0.a = a || M_{\mathsf{ct}.t}(\mathsf{ct}_1.b)|\mathsf{ct}_1.a = a) + D(\mathsf{ct}_0.a||\mathsf{ct}_1.a).\end{aligned}$$

It is easy to show that $\mathsf{ct}_0.a$ is uniform random in $\mathcal{R}$ because we re-randomized it by adding $\mathtt{Enc}(\mathsf{pk}, 0)$ to ct. Also $\mathsf{ct}_1.a$ is uniform random in $\mathcal{R}$ because it is obtained as a fresh encryption. This implies that the KL divergence $D(\mathsf{ct}_0.a||\mathsf{ct}_1.a) = 0$. We have already shown that ρ is an upper bound for the remaining term, for every a. This means that the upper bound can be rewritten as follows.

$$D(M_{\mathsf{ct}.t}(\mathsf{ct}_0.b), \mathsf{ct}_0.a || M_{\mathsf{ct}.t}(\mathsf{ct}_1.b), \mathsf{ct}_1.a) \leq \rho.$$

Then, since the KL-divergence between these two indistinguishability games is smaller than a fixed value ρ and provided that $\rho/2 \leq 2^{-\lambda-8}$, we can use Theorem 4 to relate the bit security of $\mathcal{G}_0$ with the bit security of $\mathcal{G}_1$ and we obtain that $\mathcal{G}_0$ is λ-bit $\mathsf{IND\text{-}CFA}$-secure. □

Analysis of the post-processing noise. We give an analysis of the precision lost when modifying the CKKS scheme as in Theorem 5. We instantiate the differential privacy mechanism from Definition 10 with $\rho = 2^{-\lambda-7}$. Considering that the static estimate $\mathsf{ct}.t$ is expressed in the infinity canonical norm and not in the euclidean norm, we obtain that a Gaussian noise of standard deviation $8\sqrt{n 2^\lambda} T_{\max}$ is added to each coordinate, where n is the dimension of the ring. We obtain that the bits of precision lost are $\lambda/2 + 3 + \log_2(T_{\max}) + \log_2(\sqrt{n})$.

Parameters for Machine Learning Inference. Tables 1 gives parameters for one of the most common applications of FHE which benefits from circuit privacy: privacy-preserving machine learning inference on a model with depth d and width w. For the base CKKS scheme, we consider parameters such as ring dimension and ciphertext modulus from [ACC+18]. In particular, we set the ring dimension to be smaller or equal to 2^{15} and the standard deviation for fresh encryption σ_{fresh} to be 3.2.

Table 1: Bits of additional Gaussian noise added in the modified CKKS of Theorem 5 to achieve 128-bits IND-CFA-security. We use the estimates on $T_{\max}$ from subsection 4.2 with message bound $B = 2^{10}$.

		`width`			
		$w = 1$	$w = 2^3$	$w = 2^5$	$w = 2^8$
`depth`	$d = 1$	85.50	87.67	89.54	92.50
	$d = 2$	97.08	100.99	104.63	110.51
	$d = 3$	108.08	113.45	118.76	127.53

4.2 Managing and obtaining $T_{\max}$

In this section, we will show how to set the noise bound $T_{\max}$ for the differential privacy mechanism. Recall that the usual noise estimation algorithm estimates the noise based on the circuit, which is enough for $\mathsf{IND\text{-}CPA^D}$-security when post-processing decryption as in [LMSS22]. To obtain circuit privacy, instead, we estimate the noise as the maximum noise over all circuits in a given class of circuits. In particular, we run $T_{\max} := \max_{D \in \mathcal{L}}\{\mathtt{Estimate}(D, t_{\mathsf{fresh}}, \ldots, t_{\mathsf{fresh}})\}$. Note that the estimation algorithm depends on the class of circuits; hence the evaluation process may still leak some information on the computation, like the multiplicative depth of the circuit. Below we show how to estimate the noise tag for the class of multivariate polynomials of degree bounded by some $d \in \mathbb{N}$.

Theorem 6. *Let $k, d \in \mathbb{N}$. Let $C(x_1, \ldots, x_k)$ be a multivariate polynomial of degree smaller or equal to d. Let $B \in \mathbb{N}$ such that $\|m_i\|_{\mathsf{can}} \leq B$ for $i \in [k]$, then*

$$\mathtt{Estimate}(\mathsf{sk}, \mathsf{CKKS}.\mathtt{Eval}(\mathsf{pk}, C, [\mathsf{ct}_i]_{i\in[k]}), C([m_i]_{i\in[k]})) = d\binom{k+d}{d}O(B^d t_{\mathsf{fresh}})$$

where $\mathsf{ct}_i \leftarrow \mathtt{Enc}(\mathsf{pk}, m_i)$ for $i \in [k]$.

To prove Theorem 6 we need to recall a heuristic on e_{mult}. More accurate noise analysis on $\mathsf{CKKS}.\mathtt{Eval}$ (like [CCH+24], Heuristic 8) can be found in the literature; although, considering the scope of this paper and the asymptotical nature of our results, using the following Heuristic will be enough.

Heuristic 1 (Appendix A.5 of [GHS12])**.** *Let w be the hamming weight of the secret key sk (i.e., the number of non-zero coordinates of sk) and n be the plaintext ring dimension. Then e_{mult} behaves like a random variable with mean zero and variance $O(wn)$.*

Proof. In this proof we denote $\mathtt{Estimate}(f(x), t_{\mathsf{fresh}})$ as $\mathtt{Est}(f(x))$. Also we omit the subscript can when using the canonical norm since it is the only norm used in this proof.

First, we want to prove that $\mathtt{Est}(x^d) = O(dB^{d-1}t_{\mathsf{fresh}})$ by strong induction. This is trivially true for $d = 1$. We now study the statement for $d > 1$. $\mathtt{Est}(x^d) = \mathtt{Estimate}(x^a \cdot x^b) = \|m^a e_b + m^b e_a + e_a e_b + e_{\mathsf{mult}}\|$ where e_a and e_b are, respectively, the resulting errors from the evaluation of the polynomials x^a and x^b, with $a + b = d$. We can bound this quantity from above by using the triangular inequality $\mathtt{Est}(x^d) \leq B^a\|e_b\| + B^b\|e_a\| + \|e_a e_b + e_{\mathsf{mult}}\|$. Using the strong inductive hypothesis $\|e_a\| = O(aB^{a-1}t_{\mathsf{fresh}})$ and $\|e_b\| = O(bB^{b-1}t_{\mathsf{fresh}})$, we can rewrite this quantity as $\mathtt{Est}(x^d) = O(B^a bB^{b-1}t_{\mathsf{fresh}} + B^b aB^{a-1}t_{\mathsf{fresh}}) + \|e_a e_b + e_{\mathsf{mult}}\|$. Since $\|e_a e_b + e_{\mathsf{mult}}\| \ll B^{d-1}$ we can just conclude that $\mathtt{Est}(x^d) = O(dB^{d-1}t_{\mathsf{fresh}})$.

We can now extend our study to monomials $x_1^{i_1} \ldots x_k^{i_k}$. We prove by induction on k that $\mathtt{Est}(x_1^{i_1} \ldots x_k^{i_k}) = O(dB^{d-1}t_{\mathsf{fresh}})$, where $d = i_1 + \cdots + i_k$. We already showed that it is true for $k = 1$. We now study the statement for $k > 1$. $\mathtt{Est}(x_1^{i_1} \dots x_{k-1}^{i_{k-1}} \cdot x_k^{i_k}) = \|(m_1^{i_1} \dots m_{k-1}^{i_{k-1}})e_k + m_k^{i_k} e_{k-1} + e_{k-1}e_k + e_{\mathsf{mult}}\|$, where e_{k-1} and e_k are, respectively, the resulting error from the evaluations of the monomials $x_1^{i_1} \dots x_{k-1}^{i_{k-1}}$ and $x_k^{i_k}$. We can bound this quantity from above by using the triangular inequality $\mathtt{Est}(x_1^{i_1} \dots x_k^{i_k}) \leq B^{i_1+\cdots+i_{k-1}}\|e_k\| + B^{i_k}\|e_{k-1}\| + \|e_{k-1}e_k + e_{\mathsf{mult}}\|$. Using the inductive hypothesis on e_{k-1} and e_k, we can rewrite this quantity as $\mathtt{Est}(x_1^{i_1} \dots x_k^{i_k}) = O(B^{i_1+\cdots+i_{k-1}} i_k B^{i_k-1} t_{\mathsf{fresh}} + B^{i_k}(i_1 + \cdots + i_{k-1})B^{i_1+\cdots+i_{k-1}-1} t_{\mathsf{fresh}}) + \|e_{k-1}e_k + e_{\mathsf{mult}}\|$. Since $\|e_{k-1}e_k + e_{\mathsf{mult}}\| \ll B^d$ we can just conclude that $\mathtt{Est}(x_1^{i_1} \dots x_k^{i_k}) = O(dB^{d-1}t_{\mathsf{fresh}})$ where $d = i_1 + \cdots + i_k$. Finally, we analyze a generic multivariate polynomial with k variables and degree smaller or equal to d.

$$\begin{aligned}\mathtt{Est}(\sum_{\substack{0 \leq i_1+\cdots+i_k \leq d \\ 0 < i_1,\dots,i_k \leq d}} a_{i_1,\dots,i_k} x_1^{i_1} \cdot \dots \cdot x_k^{i_k}) &\leq B\binom{k+d}{d}\mathtt{Est}(x_1^{i_1} \cdot \dots \cdot x_k^{i_k}) \\ &= B\binom{k+d}{d} O(dB^{d-1}t_{\mathsf{fresh}}) \\ &= d\binom{k+d}{d} O(B^d t_{\mathsf{fresh}}).\end{aligned}$$

□

4.3 Tightness of the Differential Privacy Parameters

As shown by Theorem 5, the proposed modified version of CKKS achieves λ-bit IND-CFA-security by applying a differentially private mechanism on the outcome of the evaluation algorithm. In practice, we instantiate the differential privacy mechanism by the Gaussian mechanism with Gaussian noise of variance $\sigma_{\mathsf{max}} \leftarrow \frac{T_{\max}^2}{2\rho}$. Recall that $\rho \leq 2^{-\lambda-7}$ is the privacy bound for ρ-KL differential privacy (Definition 9), and $T_{\max}$ is the noise upper bound for the class of circuits. We show that trying to use an appreciably smaller variance $\sigma_{\mathsf{s}} \ll \sigma_{\mathsf{max}}$ leads to the existence of an adversary that wins the IND-CFA game with a non-negligible advantage. In other words, we show that the noise parameters are tight when using the Gaussian mechanism, and the added Gaussian noise must be exponential in the security parameter.

Theorem 7. *Let $\sigma_{\mathsf{s}} > 0$. Let $\mathtt{Eval}_{\mathcal{L}_d}^{\sigma_{\mathsf{s}}}$ be the modified* CKKS *evaluation given by Algorithm 1 but where the post-processing noise is sampled from the discrete Gaussian $\mathcal{N}_{\mathbb{Z}^n}(0, \sigma_{\mathsf{s}}^2 T_{\max}^2 I_n)$. Then there exists an adversary $\mathcal{A}$ (Algorithm 2) against $\mathsf{CKKS}_{\mathcal{L}_d}^{\sigma_{\mathsf{s}}}$ in the* IND-CFA*-game such that $\mathsf{adv}^{\mathcal{A}} = \Omega(\frac{1}{\sigma_{\mathsf{s}}^2 B^2 t_{\mathsf{fresh}}^2})$, where B is an upper bound on the messages norm modulus and t_{fresh} is the noise tag associated to freshly encrypted messages.*

To prove Theorem 7 we need the following inequality that we can derive, for this case, from Theorem 1.

Lemma 3 (Theorem 1.3 of [DMR18])**.** *Let $\sigma > 0$. Then*

$$\Delta(\mathcal{N}(\mu_0, \sigma^2), \mathcal{N}(\mu_1, \sigma^2)) \geq \frac{1}{50}\frac{|\mu_0 - \mu_1|}{\sigma}.$$

Again, to prove Theorem 7 we need the following lemma.

Lemma 4. *Let $d \in \mathbb{N}$. Let B be the plaintext modulus and $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{pk}, B)$, then*

$$\mathtt{Dec}(\mathsf{sk}, \mathtt{Eval}(x^d, \mathsf{ct})) - B^d = dB^{d-1}\mathsf{ct}.e + f$$

where $\|f\|_{\mathsf{can}} = O(B^{d-1})$.

Algorithm 2: Adversary $\mathcal{A}(\lambda)$.

```
Data: A security parameter $\lambda$. The adversary has oracle access to $\mathtt{Eval}^{\sigma_s}_{\mathcal{L}_d}$.
begin
    $r \xleftarrow{\$} \mathcal{U}$;
    $r_1 \xleftarrow{\$} \mathcal{U}$;
    $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathtt{KeyGen}(\lambda; r)$;
    $m \leftarrow B$;
    $C_0 \leftarrow x^d$;
    $C_1 \leftarrow x^d + Bx^{d-1} - B^d$;
    $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{pk}, m; r_1)$;
    $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathcal{O}^{\mathtt{Eval}^{\sigma_s}_{\mathcal{L}_d}(\mathsf{pk},\cdot,\cdot,\mathsf{ct})}(C_0, C_1)$;
    $e_0 \leftarrow \mathtt{Dec}(\mathsf{sk}, \mathtt{Eval}(C_0, \mathsf{ct})) - B^d$;
    $e_1 \leftarrow \mathtt{Dec}(\mathsf{sk}, \mathtt{Eval}(C_1, \mathsf{ct})) - B^d$;
    $e_{\mathsf{res}} \leftarrow \mathtt{Dec}(\mathsf{sk}, \mathsf{ct}_{\mathsf{res}}) - B^d$;
    Choose $i \in \{0, \ldots, n-1\}$ such that $|e_{0,i} - e_{1,i}|$ is maximal;
    If $|e_{\mathsf{res},i} - e_{0,i}| \leq |e_{\mathsf{res},i} - e_{1,i}|$ then return 0. Otherwise output 1;
```

Remark 2 (On the order of operations when evaluating a polynomial)*.* When homomorphically evaluating a polynomial, the associated noise growth does not only depend from the noise of the starting ciphertexts and the polynomial itself. In particular, in CKKS, another relevant factor is how we write the polynomial as a sequence of $\mathsf{CKKS}.\mathtt{Add}$ and $\mathsf{CKKS}.\mathtt{Mul}$. For example, computing a polynomial with the *double-and-add* technique or computing it directly as $x \cdot x \cdot \cdots \cdot x$ results in two different error growths. In this theorem, we analyze the direct method. Our focus on this method simplifies the derivation of the lower bound on the noise growth. We specifically consider this case because the primary objective of this theorem in our paper is to estimate the advantage of Adversary 2 who can freely choose the order of operations for the homomorphic evaluation of the polynomial.

Proof. We define e_d as the left-hand term of the equation, therefore as

$$e_d := \mathtt{Dec}(\mathsf{sk}, \mathtt{Eval}(x^d, \mathsf{ct})) - B^d.$$

In the special case of $d = 1$ we have that $e_1 = \mathsf{ct}.e$

We now prove the result by performing an induction on the degree d. This is trivially true for $d = 2$, since

$$\mathtt{Dec}(\mathsf{sk}, \mathtt{Eval}(x^2, \mathtt{Enc}(\mathsf{pk}, B))) - B^2 = 2B\mathsf{ct}.e + \mathsf{ct}.e^2 + e_{\mathsf{mult}},$$

and $f := \mathsf{ct}.e^2 + e_{\mathsf{mult}}$ is such that $\|f\|_{\mathsf{can}} = O(B)$.

We now study the statement for $d > 2$. By computing x^d as $x^{d-1} \cdot x$, and by using CKKS noise growth rule (Lemma 1), we obtain that

$$e_d = e_{d-1}B + e_1 B^{d-1} + e_{d-1}e_1 + e_{\mathsf{mult}}.$$

Using inductive hypothesis we obtain that

$$\begin{aligned} e_d &= \left((d-1)B^{d-2}\mathsf{ct}.e + f_{d-1}\right) \cdot B + \mathsf{ct}.eB^{d-1} + \left((d-1)B^{d-1}\mathsf{ct}.e + f_{d-1}\right)\mathsf{ct}.e + e_{\mathsf{mult}} \\ &= (d-1)B^{d-1}\mathsf{ct}.e + B^{d-1}\mathsf{ct}.e + f_{d-1}B + \left((d-1)B^{d-1}\mathsf{ct}.e + f_{d-1}\right)\mathsf{ct}.e + e_{\mathsf{mult}} \\ &= dB^{d-1}\mathsf{ct}.e + f_d, \end{aligned}$$

where $f_d := f_{d-1}B + \left((d-1)B^{d-1}\mathsf{ct}.e + f_{d-1}\right)\mathsf{ct}.e + e_{\mathsf{mult}}$ and $\|f_d\|_{\mathsf{can}} = O(B^{d-1})$. □

Proof of Theorem 7. We give a brief description of the high-level idea of this proof. First, the adversary computes the ciphertext errors after the homomorphic evaluation of each circuit but before the post-processing phase of the challenger. Then, we rewrite each ciphertext error after the post-processing as a sample of a Gaussian distribution, where mean and variance only depend from the chosen circuit and variables known by the challenger. Finally, we compute the statistical distance between the two Gaussian distributions linked to the two possible circuits and use this distance to obtain a lower bound on the adversary's advantage.

The adversary knows $e := \mathsf{ct}.e$, receives the resulting error e_{res} after decrypting the oracle output and can compute the errors e_0 and e_1 obtained after the standard CKKS evaluation of C_0 and C_1 on ct. The oracle computes $\mathsf{ct}_{\mathsf{res}}$ as $\mathsf{CKKS}.\mathtt{Eval}(C_b, \mathsf{ct}) + e_{\mathsf{sm}}$, where e_{sm} is sampled from $\mathcal{N}_{\mathbb{Z}^n}(0, \sigma_{\mathsf{s}}^2 T_{\max}^2 I_n)$. This means that the adversary sees e_{res} that is a sample of $\mathcal{N}_{\mathbb{Z}^n}(e_b, \sigma_{\mathsf{s}}^2 T_{\max}^2 I_n)$. Then, the adversary analyzes the polynomial $e_0 - e_1$ and chooses i as the component where the difference of the i-th coefficients of the polynomials e_0 and e_1 is maximal in absolute value. After this, the adversary focuses on the i-th coefficient of e_{res}. This is a sample of $\mathcal{N}_{\mathbb{Z}}(e_{b,i}, \sigma_{\mathsf{s}}^2 T_{\max}^2)$. Obtaining that $|e_{\mathsf{res},i} - e_{0,i}| < |e_{\mathsf{res},i} - e_{1,i}|$ is more likely when $b = 0$ while if $|e_{\mathsf{res},i} - e_{0,i}| \geq |e_{\mathsf{res},i} - e_{1,i}|$ it is at least more likely that $b = 1$ rather then $b = 0$. To analyze the adversary's advantage in distinguishing these distributions, we first study the total variation distance between them. Computing this quantity for discrete Gaussians is not an easy task, therefore we will approximate it by considering their counterparts on the real numbers. By Lemma 3 and Lemma 4 we have that

$$\Delta(\mathcal{N}(e_{0,i}, \sigma_{\mathsf{s}}^2 T_{\max}^2), \mathcal{N}(e_{1,i}, \sigma_{\mathsf{s}}^2 T_{\max}^2)) \geq \frac{1}{50} \frac{|e_{0,i} - e_{1,i}|}{\sigma_{\mathsf{s}} T_{\max}} = \Theta\left(\frac{B^{d-1}|e_i|}{\sigma_{\mathsf{s}} T_{\max}}\right).$$

Theorem 6 gives us that $T_{\max} = d(d-1)O(B^d t_{\mathsf{fresh}})$ and $|e_i| \geq 1$ with high probability. We can now rewrite the right hand term of the past equation as $\Omega(\frac{1}{\sigma_{\mathsf{s}} B t_{\mathsf{fresh}}})$. The adversary's advantage in the $\mathsf{IND\text{-}CFA}$ game for this scheme is the square of the total variation distance we just estimated, therefore $\Omega(\frac{1}{\sigma_{\mathsf{s}}^2 B^2 t_{\mathsf{fresh}}^2})$.

□

Theorem 8. *If the* CKKS *scheme with the modified evaluation* $\mathtt{Eval}_{\mathcal{L}_d}^{\sigma_{\mathsf{s}}}$ *is λ-bit* $\mathsf{IND\text{-}CFA}$*-secure, then* $\sigma_{\mathsf{s}} = \Omega(2^{\lambda/2}/(B^2 t_{\mathsf{fresh}}^2))$. *This implies that one must add at least* $\lambda/2 - \log_2 \widetilde{\Omega}(B^2 t_{\mathsf{fresh}}^2)$ *bits of additional Gaussian noise to the standard* CKKS *evaluation in order to achieve* $\mathsf{IND\text{-}CFA}$ *security.*

Proof. By using the definition of bit security, we know that

$$\lambda \leq \log_2 O(\frac{T(A)}{\mathsf{adv}^A}) \leq \log_2 O(\sigma_s^2 B^2 t_{\mathsf{fresh}}^2);$$

this immediately implies that $\sigma_s \geq 2^{\lambda/2}/(B^2 t_{\mathsf{fresh}}^2)$ and $\lambda/2-$ $\log_2 \Omega(B^2 t_{\mathsf{fresh}}^2) \leq \log_2 \sigma_s$. □

5 Threshold FHE and MPC

In Subsection 5.1, we give definitions for threshold homomorphic encryption over approximate arithmetic. In Subsection 5.2, we give definitions for multikey homomorphic encryption over approximate arithmetic. In Subsection 5.3 we present a modification of the $\mathsf{MK\text{-}CKKS}$ multikey homomorphic encryption scheme that satisfies the indistinguishability security definition as given by Definition 23. In particular, we show that re-randomized $\mathsf{MK\text{-}CKKS}$ ciphertexts and decryption shares does not reveal information about messages and secret keys of non-corrupted parties when we apply an appropriate differential privacy mechanism that floods them with an exponential Gaussian sample. Finally, in Subsection

5.4, we show that the parameters are tight. Namely, the Gaussian noise must be exponential in the security parameter, and a significantly lower noise parameter leads to an efficient adversary against IND-MKHE-security.

5.1 Threshold Homomorphic Encryption

We base our definition for threshold approximate homomorphic encryption on the definition introduced by [BGG$^+$18]. We have the same syntax and we have the same indistinguishability definition as [BGG$^+$18], but we redefine the correctness definition for the case of approximate arithmetic. Regarding the indistinguishability, we discuss in Remark 3 a slight strengthening of the definition that lets us construct a meaningful circuit private homomorphic encryption scheme.

Recall that a monotone access structure $\mathbb{A}$ on $[n]$ is a collection $\mathbb{A} \subseteq \mathcal{P}([n])$, where $\mathcal{P}([n])$ contains all subsets of $[n]$, such that whenever we have sets B, C satisfying $B \in \mathbb{A}$ and $B \subseteq C \subseteq [n]$ then $C \in \mathbb{A}$. The sets in $\mathbb{A}$ are called the valid sets and the sets in $\mathcal{P}([n]) \setminus \mathbb{A}$ are called invalid sets.A class of monotone access structures is a collection $\mathcal{S} = (\mathbb{A}_1, \ldots, \mathbb{A}_t) \subseteq \mathcal{P}(\mathcal{P}([n]))$ of monotone access structures on $[n]$. A set $S \subseteq [n]$ is a maximal invalid share set if $S \notin \mathbb{A}$ and for every $i \in [n] \setminus S$ we have that $S \cup \{i\} \in \mathbb{A}$.

Definition 16 (Threshold Homomorphic Encryption)**.** Let $d \in \mathbb{N}$ and let $\mathcal{L}_d$ be a class of circuits of multiplicative depth smaller or equal to d. A threshold homomorphic encryption scheme THE on $\mathcal{L}_d$ is a tuple of five algorithms $\mathsf{THE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ with the following syntax.

$\mathtt{KeyGen}(\lambda, d, n, \mathbb{A}) \to (\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$**:** Given a security parameter λ, the maximal multiplicative depth of evaluatable circuits d, the number of parties n, and access structure $\mathbb{A}$, returns a public key pk and n secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$.

$\mathtt{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$**:** Given a public key pk and a message m, returns a ciphertext ct.

$\mathtt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k) \to \mathsf{ct}$**:** Given a public key pk, a circuit $C \in \mathcal{L}_d$ and ciphertexts $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$, returns a ciphertext ct.

$\mathtt{PDec}(\mathsf{sk}_i, \mathsf{ct}) \to \mu$**:** Given a secret key sk_i and a ciphertext ct, returns a partial decryption μ.

$\mathtt{Combine}(\{\mu_i\}_{i \in S}, \mathsf{ct}) \to m$**:** Given a set of partial decryptions $\{\mu_i\}_{i \in S}$ where $S \in \mathbb{A}$, and a ciphertext ct, returns a message m.

Definition 17 (Threshold Ciphertext Error)**.** Let $\mathsf{THE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ be a threshold homomorphic encryption scheme with message space $\mathcal{M}$. Furthermore, let $\mathcal{M}$ be a normed space with norm $|| \cdot || : \mathcal{M} \mapsto \mathbb{R}_{\geq 0}$. For all public/secret key tuples $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathtt{KeyGen}(\lambda, d, n, \mathbb{A})$, for any ciphertext ct in the image of $\mathtt{Eval}$ and any message $m \in \mathcal{M}$ the ciphertext error of ct w.r.t. m is defined as

$$\mathtt{Error}(\mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{ct}, m) = ||\mathtt{Combine}([\mathtt{PDec}(\mathsf{sk}_i, \mathsf{ct})]_{i \in S}) - m||.$$

Definition 18 (Approximate Correctness)**.** Let us define $\mathsf{THE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ to be a threshold homomorphic encryption scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$ that is a normed space with norm $|| \cdot || : \widetilde{\mathcal{M}} \mapsto \mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ be the class of circuits, $\mathcal{L}_k \subseteq \mathcal{L}$ be the subset of circuits with k input wires, and let $\mathtt{Estimate} : \bigsqcup_{k \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}^k_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ be an efficiently computable function. We call HE an approximate homomorphic encryption scheme if for all $k \in \mathbb{N}$, for all $C \in \mathcal{L}_k$, for all $(\mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathtt{KeyGen}(\lambda, d, n, \mathbb{A})$, if $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ and $m_1, \ldots, m_k$ are such that $\mathtt{Error}(\mathsf{sk}_i, \mathsf{ct}_i, m_i) \leq t_i$, for some $t_1, \ldots, t_k \in \mathbb{R}_{\geq 0}$, and $\mathsf{ct} \leftarrow \mathtt{Eval}(\mathsf{pk}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$, then

$$\mathtt{Error}(\mathsf{sk}_1, \ldots, \mathsf{sk}_k, \mathsf{ct}, C(m_1, \ldots, m_k)) \leq \mathtt{Estimate}(C, t_1, \ldots, t_k).$$

Definition 19 (Ind-secure THE)**.** Let $d, n \in \mathbb{N}$ and let $\mathcal{L}_d$ be a class of circuits of multiplicative depth smaller or equal to d. Let $\mathsf{THE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ be a threshold fully homomorphic encryption scheme for a class of access structures $\mathbb{S}$ and circuits in $\mathcal{L}_d$. We define the experiment $\mathtt{Exp}_b^{\mathsf{IND\text{-}THE}}[\mathcal{A}]$, where $b \in \{0,1\}$ is a bit and $\mathcal{A}$ is an adversary. The experiment is defined as follows:

$$
\begin{aligned}
\mathtt{Exp}_b^{\mathsf{IND\text{-}THE}}[\mathcal{A}](\lambda):\ & \mathbb{A} \leftarrow \mathcal{A}(\lambda, d, n, \mathbb{S}),\\
& (\mathsf{sk}_1, \dots, \mathsf{sk}_n, \mathsf{pk}) \leftarrow \mathtt{KeyGen}(\lambda, \mathbb{A}),\\
& S \leftarrow \mathcal{A}(\mathsf{pk}) \text{ s.t. } S \notin \mathbb{A} \text{ and S is a maximal invalid set},\\
& (m_1^{(0)}, \dots, m_k^{(0)}, m_1^{(1)}, \dots, m_k^{(1)}), \mathsf{st} \leftarrow \mathcal{A}([\mathsf{sk}_i]_{i \in S}),\\
& [\mathsf{ct}_i \leftarrow \mathsf{THE}.\mathtt{Enc}(\mathsf{pk}, m_i^{(b)})]_{i=1}^k,\\
& b' \leftarrow \mathcal{A}^{\mathtt{Eval}(\mathsf{pk}, \cdot, \mathsf{ct}_1, \dots, \mathsf{ct}_k)}(\mathsf{st}, \mathsf{ct}_1, \dots, \mathsf{ct}_n),\\
& \textbf{return } b'.
\end{aligned}
$$

The $\mathtt{Eval}(\mathsf{pk}, \cdot, \mathsf{ct}_1, \dots, \mathsf{ct}_k)$ oracle takes as input circuit in $C_i \in \mathcal{L}_d$ is such that $C_i(m_1^{(0)}, \dots, m_k^{(0)}) = C_i(m_1^{(1)}, \dots, m_k^{(1)})$. The oracle computes and outputs $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathtt{Eval}(\mathsf{pk}, C_i, \mathsf{ct}_1, \dots, \mathsf{ct}_k)$ and $\mu_j \leftarrow \mathtt{PDec}(\mathsf{sk}_j, \mathsf{ct}_{\mathsf{res}})$ for all $j \in [n]$.

The scheme THE is said to be λ-bit $\mathsf{IND\text{-}THE}$-secure if, for any adversary $\mathcal{A}$, we have that $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}}$, where $\mathsf{adv}^{\mathcal{A}}$ is defined as in Definition 12.

5.2 Multikey Homomorphic Encryption

There are many flavors of multikey homomorphic encryption in the literature. Most of the definitions differ in syntax, but the overall concept is same. The main differences between a multikey homomorphic encryption scheme and threshold homomorphic encryption schemes are (1) in MKHE the secret keys are generated by each user separately instead of by a single setup, (2) messages are encrypted with public keys of each user instead of a master public key. Consequently, the evaluation algorithm in MKHE "combines" ciphertexts with respect to different public keys into one ciphertext, whereas in THE the ciphertext is already combined. Finally, (3) the decryption process in MKHE is a special case of THE where all secret keys are needed to decrypt the message.

Both primitives, however, share the same interface for decryption. In particular, both primitives define a partial decryption algorithm $\mathtt{PDec}$. Furthermore, to the best of our knowledge, all current realizations of these primitives use a flavor of noise flooding to realize $\mathtt{PDec}$. Hence it makes sense in our paper to investigate multikey homomorphic encryption together with threshold homomorphic encryption.

Below we give the syntax for multikey homomorphic encryption.

Definition 20 (Multikey Homomorphic Encryption)**.** Let $d \in \mathbb{N}$ and let $\mathcal{L}_d$ be a class of circuits of multiplicative depth smaller or equal to d. A multikey homomorphic encryption scheme MKHE on $\mathcal{L}_d$ is a tuple of five algorithms $\mathsf{MKHE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ with the following syntax.

$\mathtt{KeyGen}(\lambda, d) \to (\mathsf{pk}, \mathsf{sk})$**:** Given a security parameter λ, the maximal multiplicative depth of evaluatable circuits d, the algorithm returns a public key pk and s secret key sk.

$\mathtt{Enc}(\mathsf{pk}, m) \to \mathsf{ct}$**:** Given a public key pk and a message m, the algorithm returns a ciphertext ct.

$\mathtt{Eval}(\mathsf{pk}_1, \dots, \mathsf{pk}_n, C, \mathsf{ct}_1, \dots, \mathsf{ct}_n) \to \mathsf{ct}$**:** Given a list of public keys $\mathsf{pk}_1, \dots, \mathsf{pk}_n$, a circuit $C \in \mathcal{L}_d$ and ciphertexts $\mathsf{ct}_1, \dots, \mathsf{ct}_n$, returns a ciphertext ct.

$\texttt{PDec}(\mathsf{sk}_i, \mathsf{ct}) \to \mu$**:** Given a secret key sk_i and a ciphertext ct, returns a partial decryption μ.

$\texttt{Combine}(\{\mu_i\}_{i\in[n]}, \mathsf{ct}) \to m$**:** Given a set of partial decryptions $\{\mu_i\}_{i\in[n]}$ and a ciphertext ct, returns a message m.

Definition 21 (Multikey Ciphertext Error)**.** Let $\mathsf{MKHE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{PDec}, \texttt{Combine})$ be a multikey homomorphic encryption scheme with message space $\mathcal{M}$. Furthermore, let $\mathcal{M}$ be a normed space with norm $||\cdot|| : \mathcal{M} \mapsto \mathbb{R}_{\geq 0}$. For all public/secret key pairs $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \texttt{KeyGen}(\lambda)$ where $i \in [n]$, any ciphertext ct in the image of $\texttt{Eval}$ and message $m \in \mathcal{M}$ the ciphertext error is defined as

$$\texttt{Error}(\mathsf{sk}_1, \ldots, \mathsf{sk}_n, \mathsf{ct}, m) = ||\texttt{Combine}([\texttt{PDec}(\mathsf{sk}_i, \mathsf{ct})]_{i\in[n]}) - m||.$$

Below we give our definition of approximate correctness for multikey homomorphic encryption. Definition 23 gives our definition for indistinguishability security of multikey homomorphic encryption. Recall that this is the first security definition for multikey approximate homomorphic encryption that gives the adversary access to partial decryptions. Previously [CDKS19], only standard semantic security was considered, and security in the presence of partial decryptions were omitted.

Definition 22 (Approximate Correctness)**.** Let us define $\mathsf{MKHE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{PDec}, \texttt{Combine})$ to be a multikey homomorphic encryption scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$ that is a normed space with norm $||\cdot|| : \widetilde{\mathcal{M}} \mapsto \mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ be the class of circuits, $\mathcal{L}_k \subseteq \mathcal{L}$ be the subset of circuits with k input wires, and let $\texttt{Estimate} : \bigsqcup_{k\in\mathbb{N}} \mathcal{L}_k \times \mathbb{R}^k_{\geq 0} \mapsto \mathbb{R}_{\geq 0}$ be an efficiently computable function. We call HE an approximate homomorphic encryption scheme if for all $k \in \mathbb{N}$, for all $C \in \mathcal{L}_k$, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \texttt{KeyGen}(\lambda)$, if $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ and $m_1, \ldots, m_k$ are such that $\texttt{Error}(\mathsf{sk}_i, \mathsf{ct}_i, m_i) \leq t_i$, for some $t_1, \ldots, t_k \in \mathbb{R}_{\geq 0}$, and $\mathsf{ct} \leftarrow \texttt{Eval}(\mathsf{pk}_1, \ldots, \mathsf{pk}_k, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$, then

$$\texttt{Error}(\mathsf{sk}_1, \ldots, \mathsf{sk}_k, \mathsf{ct}, C(m_1, \ldots, m_k)) \leq \texttt{Estimate}(C, t_1, \ldots, t_k).$$

Definition 23 (Ind-secure MKHE)**.** Let $d \in \mathbb{N}$ and let $\mathcal{L}_d$ be a class of circuits of multiplicative depth smaller or equal to d. Let $\mathsf{MKHE} = (\texttt{KeyGen}, \texttt{Enc}, \texttt{Eval}, \texttt{PDec}, \texttt{Combine})$ be a multikey homomorphic encryption scheme for a class circuits in $\mathcal{L}_d$. We define the experiment $\texttt{Exp}_b^{\mathsf{IND\text{-}MKHE}}[\mathcal{A}]$, where $b \in \{0, 1\}$ is a bit and $\mathcal{A}$ is an adversary. The experiment is defined as follows:

```
$\texttt{Exp}_b^{\mathsf{IND\text{-}MKHE}}[\mathcal{A}](\lambda)$ :
        $[r'_i \xleftarrow{\$} \mathcal{U}]_{i\in[n]}$,
        $[(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \texttt{KeyGen}(\lambda, d, r'_i)]_{i\in[n]}$,
        $i^*, \mathsf{st}_1 \leftarrow \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$,
        $[r_i \xleftarrow{\$} \mathcal{U}]_{i\in[n]}$,
        $(m_1^{(0)}, \ldots, m_n^{(0)}, m_1^{(1)}, \ldots, m_n^{(1)}), \mathsf{st}_2 \leftarrow \mathcal{A}(\mathsf{st}_1, [r_i, r'_i]_{i\in[n]\setminus\{i^*\}})$,
        $[\mathsf{ct}_i \leftarrow \mathsf{MKHE}.\mathsf{Enc}(\mathsf{pk}_i, m_i^{(b)}, r_i)]_{i\in[n]}$,
        $b' \leftarrow \mathcal{A}^{\texttt{Eval}(\{\mathsf{pk}_i\}_{i\in[n]}, \cdot, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)}(\mathsf{st}_2, \mathsf{ct}_{i^*})$,
        **return** $b'$.
```

The $\texttt{Eval}(\{\mathsf{pk}_i\}_{i\in[n]}, \cdot, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ oracle takes as input a circuit $C_i \in \mathcal{L}_d$ such that $C_i(m_1^{(0)}, \ldots, m_n^{(0)}) = C_i(m_1^{(1)}, \ldots, m_n^{(1)})$. The oracle computes and outputs $\mathsf{ct}_{\mathsf{res}} \leftarrow \texttt{Eval}(\{\mathsf{pk}_i\}_{i\in[n]}, C_i, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ and $\mu_j \leftarrow \texttt{PDec}(\mathsf{sk}_j, \mathsf{ct}_{\mathsf{res}})$ for all $j \in [n]$.

The scheme MKHE is said to be λ-bit $\mathsf{IND\text{-}MKHE}$-secure if, for any adversary $\mathcal{A}$, we have that $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}}$, where $\mathsf{adv}^{\mathcal{A}}$ is defined as in Definition 12.

An important question when stating a new security definition is whether the definition is meaningful in any way. Intuitively it seems that our definition captures what we would expect from the multikey HE. In particular, the adversary should not be able to distinguish encryptions even when given all secret keys except one, and given partial decryptions on evaluated ciphertexts. To give a more formal argument, we show how to use a multikey homomorphic encryption scheme for two keys to build a homomorphic encryption scheme with circuit privacy. To do this, we need to recall the definition of universal circuit.

Definition 24 (Universal circuit [Val76])**.** A universal circuit U takes as inputs a circuit C (of bounded depth d and width k) and a vector of messages $m_1, \ldots, m_k$, and outputs $C(m_1, \ldots, m_k)$.

Theorem 9. *Let* MKHE *be a* $\mathsf{IND\text{-}MKHE}$*-secure multikey homomorphic encryption scheme for* $n = 2$ *parties with message space* $\mathcal{M} \subseteq \mathcal{L}_d$*. We can build a homomorphic encryption scheme* HE *on* $\mathcal{L}_d$ *that is* $\mathsf{IND\text{-}CFA}$*-secure.*

Proof. Let MKHE be a multikey homomorphic encryption for $n = 2$ keys. We build the HE encryption as follows. The $\mathtt{KeyGen}$ and $\mathtt{Enc}$ algorithms are the same as in MKHE. We denote the keys output by the $\mathtt{KeyGen}$ algorithm as $(\mathsf{sk}_1, \mathsf{pk}_1)$. The evaluation algorithm $\mathsf{HE}.\mathtt{Eval}$ on input $\mathsf{ct}_1 \leftarrow \mathsf{MKHE}.\mathtt{Enc}(pk_1, m)$ first samples $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathtt{KeyGen}(\lambda, d)$, encrypts the circuit C as $\mathsf{ct}_2 \leftarrow \mathtt{Enc}(\mathsf{pk}_2, C)$, and evaluates $\mathsf{ct} \leftarrow \mathsf{MKHE}.\mathtt{Eval}((\mathsf{pk}_1, \mathsf{pk}_2), U, \mathsf{ct}_1, \mathsf{ct}_2)$, where U is a universal circuit that supports the evaluation of circuits in $\mathcal{L}_d$. Finally, the eval algorithm outputs ct and $\mu_2 \leftarrow \mathtt{PDec}(\mathsf{sk}_2, \mathsf{ct})$.

The decryption algorithm $\mathsf{HE}.\mathtt{Dec}$ runs $\mathsf{ct} \leftarrow \mathsf{MKHE}.\mathtt{Eval}((\mathsf{pk}_1,\ \mathsf{pk}_2),\ U,\ \mathsf{ct}_1,\ \mathsf{ct}_2)$, $\mu_1 \leftarrow \mathtt{PDec}(\mathsf{sk}_1, \mathsf{ct})$, and $m' \leftarrow \mathtt{Combine}(\{\mu_i\}_{i \in [n]}, \mathsf{ct})$. Note that from approximate correctness of MKHE we have that m' is close to $C(m)$, what implies that the HE is approximately correct.

Now we proceed to show circuit privacy. We construct a solver $\mathcal{S}$ that uses an adversary $\mathcal{A}$ against $\mathsf{IND\text{-}CFA}$ of HE to break $\mathsf{IND\text{-}MKHE}$. The solver $\mathcal{S}$ obtains $\mathsf{pk}_1, \mathsf{pk}_2$ from the $\mathsf{IND\text{-}MKHE}$ challenger, and sends $i^* = 2$ back. The solver $\mathcal{S}$ obtains r_1 and r_1' and passes both to the adversary. $\mathcal{A}$ responds with $(m_1, \ldots, m_k)$ and C_0 and C_1, and sends $(m_1, \ldots, m_k, C_0)$ and $(m_1, \ldots, m_k, C_1)$ to the MKHE challenger. Consequently, $\mathcal{S}$ obtains ct_1 and ct_2, and queries the $\mathtt{Eval}$ oracle on the U circuit and both ciphertexts. Denote the response of the oracle as μ_2. The solver returns μ_2 and $\mathsf{ct} \leftarrow \mathtt{Eval}(\mathsf{pk}_1, \mathsf{pk}_2, U, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ to $\mathcal{A}$. If $\mathcal{A}$ returns a bit b' the solver outputs it as its solution to the $\mathsf{IND\text{-}MKHE}$ experiment.

Note that $\mathcal{S}$ perfectly follows the $\mathsf{IND\text{-}MKHE}$ experiment. In particular, we set $(m_1^{(b)}, m_2^{(b)}) = (m_1, \ldots, m_k, C_b)$. Note that we set $m_1^{(b)} = (m_1, \ldots, m_k)$ and $m_2^{(b)} = C_b$. From the requirement on C_0 and C_1 imposed by the $\mathsf{IND\text{-}CFA}$ definition we have that $C_0(m_1, \ldots, m_k) = C_1(m_1, \ldots, m_k)$, and $U(C_0, m_1, \ldots, m_k) = U(C_1, m_1, \ldots, m_k)$ as required by the $\mathsf{IND\text{-}MKHE}$ experiment. To summarize, we have that the simulator $\mathcal{S}$ has advantage $\mathsf{adv}_{\mathsf{IND\text{-}CFA}}[\mathcal{A}](\lambda)$ in returning the b' such that $b' = b$ and also has a running time that is similar to the running time of $\mathcal{A}$. □

Remark 3 (On threshold homomorphic encryption and circuit privacy). Recall that we proved that multikey homomorphic encryption for two keys already gives us homomorphic encryption with indistinguishability circuit privacy. Note that the definition of threshold homomorphic encryption does not let itself use to build circuit privacy so easily. The reasons for this are that the common key generation algorithm in Definition 16 returns just one public key and all secret keys, and we cannot give the random seed to the adversary to generate its own keys honestly. Similarly, we would need to redefine the $\mathsf{IND\text{-}THE}$ experiment and encrypt part of the messages using honestly sampled seeds that are then passed to the adversary. Note that this modification strengthens the security notion. However, we are still unable to provide a seed for the key generation algorithm since $\mathsf{IND\text{-}THE}$ would be trivially broken. In this case, we would need to introduce a relaxation of our indistinguishability circuit privacy definition such that the adversary is given a secret key instead of a seed.

5.3 Achieving IND-MKHE-security for MK-CKKS

In this subsection we analyze the scheme MK-CKKS from [CDKS19] and show how to modify it to achieve IND-MKHE-security. We stress that this construction can also be adapted to other MKHE schemes that share similarities with MK-CKKS. In particular, the relevant properties we use are: the linearity of the `Combine` algorithm and the structure of extended ciphertext in $\mathcal{R}^k$, where all elements except one are uniform random in fresh encryptions. We present the algorithms of MK-CKKS, but we refer the reader to the original paper [CDKS19] for a complete description.

MK-CKKS.`Setup`(λ): Given the security parameter λ, set $n \in \mathbb{N}$ and $Q \in \mathbb{N}$, the ring $\mathcal{R} := \mathcal{R}_Q^n$, the key distribution χ and the noise distribution ψ. Sample $a \xleftarrow{\$} \mathcal{R}_Q^n$ uniformly. Return $\mathsf{pp} = (n, Q, \chi, \psi, a)$.

MK-CKKS.`KeyGen`(pp): Sample $s \leftarrow \chi$. Sample an error $e \leftarrow \psi$ and compute $b = -sa + e$. Return $((b, a), s)$ as $(\mathsf{pk}, \mathsf{sk})$.

MK-CKKS.`Enc`($\mathsf{pk}, m \in \mathcal{R}_Q$): Sample $v \leftarrow \chi$ and $e_0, e_1 \leftarrow \psi$. Denoting $\mathsf{pk} = (b, a)$, then compute $c_0 = vb_0 + m + e_0$ and $c_1 = va_0 + e_1$. Return $(c_0, c_1) \in \mathcal{R}^2$.

MK-CKKS.`Eval`($\{\mathsf{pk}_i\}_{i\in[k]}, C, \overline{\mathsf{ct}}_1, \ldots, \overline{\mathsf{ct}}_k$) : For given ciphertexts $\overline{\mathsf{ct}}_i \in \mathcal{R}^{k_i+1}$, we denote $k \geq \max_{i\in[k]}\{k_i\}$ the number of parties involved in at least one of the $\overline{\mathsf{ct}}_i$. Rearrange the entries of each $\overline{\mathsf{ct}}_i$ and pad zeroes in empty entries to generate some ciphertexts $\overline{\mathsf{ct}}_i^*$ sharing the same secret key $\overline{\mathsf{sk}} = (1, \mathsf{sk}_1, \ldots, \mathsf{sk}_k)$. Then, the algorithm evaluates the arithmetic circuit C by means of addition and multiplication:

CKKS.`Add`($\overline{\mathsf{ct}}_0, \overline{\mathsf{ct}}_1 \in \mathcal{R}^{k+1}$): Return the entry-by-entry addition $\overline{\mathsf{ct}_0} + \overline{\mathsf{ct}_1}$.

CKKS.`Mul`($\{\mathsf{pk}_i\}_{i\in[k]}, \overline{\mathsf{ct}}_0, \overline{\mathsf{ct}}_1 \in \mathcal{R}^{k+1}$): Compute $\overline{\mathsf{ct}} = \overline{\mathsf{ct}}_1 \otimes \overline{\mathsf{ct}}_2$ and return the ciphertext $\overline{\mathsf{ct}}' \leftarrow \mathtt{Relin}(\overline{\mathsf{ct}}, \{\mathsf{pk}_i\}_{i\in[k]})$. The `Relin` algorithm returns a ciphertext $\overline{\mathsf{ct}} \in \mathcal{R}^{k+1}$ encrypting m_0m_1 with an error that follows the noise growth law of Lemma 5.

MK-CKKS.`PDec`($\mathsf{sk}, \overline{\mathsf{ct}} \in \mathcal{R}^{k+1}$): Call $\overline{\mathsf{ct}}.a_i$ the component of $\overline{\mathsf{ct}}$ associated to the secret key sk. Return $\mu = \mathsf{sk} \cdot \overline{\mathsf{ct}}.a_i$. [1]

MK-CKKS.`Combine`($\{\mu_i\}_{i\in[k]}, \overline{\mathsf{ct}} \in \mathcal{R}^{k+1}$): Return $m = \overline{\mathsf{ct}}.b + \sum_{i=1}^{k} \mu_i$.

The estimate function of MK-CKKS is handled similarly to CKKS but with the noise growth rule of Lemma 5.

While this presentation of MK-CKKS is helpful for describing how the scheme works, it might not explain in a clear way why MK-CKKS is a MKHE scheme. To see this, first notice that the `KeyGen` can be made independently by each party. In fact, we can consider the output of MK-CKKS.`Setup` as a common reference string that is used as input for MK-CKKS.`KeyGen`. Second, consider the ciphertext space to be the disjoint union $\bigcup_{i\geq 2} \mathcal{R}^i$. By doing this, all the ciphertexts in the input/output of MK-CKKS.`Eval` live in the same space, so the algorithm satisfies the definition from MKHE.

To simplify the notation, from now on, we are going to refer to the entries of a ciphertext $\mathsf{ct} \in \mathcal{R}^{k+1}$ as $(\mathsf{ct}.b, \mathsf{ct}.a_1, \ldots, \mathsf{ct}.a_k)$. Also, when writing $\mathsf{ct}.a$, we will be referring

[1] In the original scheme, the partial decryption algorithm already added a smudging noise $e_{\mathsf{sm}} \leftarrow \phi$. Since ϕ is not described in detail, we decided not to include it here so as to simplify the exposition of `PDec` in Algorithm 4.

to $(\mathsf{ct}.a_1, \ldots, \mathsf{ct}.a_k)$. We now show how to modify the Eval and the PDec algorithm in MK-CKKS to achieve IND-MKHE-security. The main idea behind Eval′ is to re-randomize the ciphertext by adding a fresh encryption of zero for each public key pk associated to ct and then to post-process the component $\mathsf{ct}.b$ using an appropriate differential privacy mechanism M_T.

Algorithm 3: The modified evaluation MK-CKKS.Eval′

```
Data: A set of public keys $\{\mathsf{pk}_i\}_{i\in[k]}$, circuit $C \in \mathcal{L}$, a vector of ciphertexts
      $\mathsf{ct}_1 \in \mathcal{R}^{k+1}, \ldots, \mathsf{ct}_N \in \mathcal{R}^{k+1}$.
begin
    $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathtt{Eval}(\{\mathsf{pk}_i\}_{i\in[k]}, C, \mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ ;
    For $i = 1$ to $k$: $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathsf{CKKS.Add}(\mathsf{pk}, \mathsf{ct}_{\mathsf{res}}, \mathtt{Enc}(\mathsf{pk}_i, 0))$ ;
    $T \leftarrow \mathsf{ct}_{\mathsf{res}}.t + t_{\mathsf{fresh}}$ ;
    $\mathsf{ct}_{\mathsf{res}}.b \leftarrow M_T(\mathsf{ct}_{\mathsf{res}}.b)$;
    return $\mathsf{ct}_{\mathsf{res}}$ ;
```

Algorithm 4: The modified partial decryption MK-CKKS.PDec′

```
Data: A secret key sk, a ciphertext $\mathsf{ct} \in \mathcal{R}^{k+1}$.
begin
    $\mu \leftarrow M_{\mathsf{ct}.t}(\mathtt{PDec}(\mathsf{sk}, \mathsf{ct}))$ ;
    return $\mu$ ;
```

Theorem 10. *Let* MK-CKKS = (Setup*,* KeyGen*,* Enc*,* Eval*,* PDec*,* Combine) *be the* MK-CKKS *multikey homomorphic encryption scheme, with plaintext space $\mathcal{R}$ and estimate function* Estimate*. Let $q \in \mathbb{N}$. Let M_t be a ρ-KLDP mechanism on $\mathcal{R}$ where $\rho \leq 2^{-\lambda-8}/q$. If* MK-CKKS.Enc *is $(\lambda + 8)$-bit secure in the* IND-CPA *game, then* MK-CKKS *with the modified* MK-CKKS.Eval′ *given by Algorithm 3 and with the modified* MK-CKKS.PDec′ *given by Algorithm 4 is λ-bit secure in the* IND-MKHE *game where q is the maximum amount of oracle queries by the adversary.*

Proof. The high-level idea is as in Theorem 5. The main difference between the two proofs is the structure of the game $\mathcal{G}_1$ that has not only to protect the message choice b but also to guarantee the protection of sk_{i^*}. Also, the output of the adversary's queries is not a rLWE ciphertext anymore but it is a couple made by an extended rLWE ciphertext and a partial decryption share. This makes the tasks of upper-bounding the KL-divergence somewhat harder.

We start by describing the two indistinguishability games.

- $\mathcal{G}_0$: the MK-CKKS scheme with the modified algorithms given by Algorithm 3 and Algorithm 4 in the IND-MKHE-security game with a bound of maximum q queries.
- $\mathcal{G}_1$: the original MK-CKKS scheme in a variant of the IND-MKHE-security game with a bound of maximum q queries and the modified oracle Eval′. The oracle $\mathtt{Eval}'(\{\mathsf{pk}_i\}_{i=1}^n, \cdot, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ takes as input a circuit $C_i \in \mathcal{L}_d$ that satisfies the equality $C_i(m_1^{(0)}, \ldots, m_n^{(0)}) = C_i(m_1^{(1)}, \ldots, m_n^{(1)})$, and behaves in the following way. When writing $\mathtt{Enc}_{\mathsf{n}}(\mathsf{pk}, m)$ we denote a noiseless encryption of m. Also, the sums

among ciphertexts are a shorter notation for $\mathsf{CKKS.Add}$.

```
$\mathtt{Eval}'(\{\mathsf{pk}_i\}_{i\in[n]}, \cdot, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ :
    $m_{\mathsf{res}} \leftarrow C(m_1^{(0)}, \ldots, m_n^{(0)})$,
    $\mathsf{ct}_{\mathsf{res}} \leftarrow \mathtt{Enc}(\mathsf{pk}_{i^*}, 0) + \sum_{j\in[n]\setminus\{i^*\}} \mathtt{Enc}_{\mathsf{n}}(\mathsf{pk}_j, 0)$,
    $\mathsf{ct}_{\mathsf{res}}.t \leftarrow \mathtt{Estimate}(C, \mathsf{ct}_1.t, \ldots, \mathsf{ct}_n.t) + (k+1)t_{\mathsf{fresh}}$,
    $\mu_{i^*} \leftarrow M_{\mathsf{ct}_{\mathsf{res}}.t}(\mathsf{ct}_{\mathsf{res}}.b - \sum_{j\neq i^*} \mathsf{sk}_j \cdot \mathsf{ct}_{\mathsf{res}}.a_j)$,
    $[\mu_i \leftarrow \mathsf{sk}_i \cdot \mathsf{ct}_{\mathsf{res}}.a_i]_{i\neq i^*}$,
    $\mathsf{ct}_{\mathsf{res}}.b \leftarrow M_{\mathsf{ct}_{\mathsf{res}}.t}(\mathsf{ct}_{\mathsf{res}}.b + m_{\mathsf{res}})$,
    $\mathbf{return}(\mathsf{ct}_{\mathsf{res}}, [\mu_i]_{i\in[n]})$.
```

In $\mathcal{G}_0$, the ciphertext $\mathsf{ct}_{\mathsf{res}}$ and the decryption shares μ_i are obtained by homomorphically evaluating the circuit C on the input ciphertexts and partially decrypting the resulting ciphertext. After computing them, we perform some post-processing with a re-randomization on $\mathsf{ct}_{\mathsf{res}}$ and with a differential privacy mechanism on both. In $\mathcal{G}_1$, the ciphertext $\mathsf{ct}_{\mathsf{res}}$ and the decryption shares μ_i are simulated, and they do not depend from the input ciphertexts, from b or from the secret key of the non-corrupted party i^*. $\mathsf{ct}_{\mathsf{res}}$, in the final output of $\mathcal{G}_1$, after modifications, is a fresh, random encryption of m_{res}, and the share μ_{i^*} is obtained without using sk_{i^*}.

To simplify the notation in this proof, we will denote $\mathsf{ct}_{\mathsf{res}}^{\mathcal{G}_0}$ as ct_0, $\mathsf{ct}_{\mathsf{res}}^{\mathcal{G}_1}$ as ct_1 and $\mathsf{ct}_{\mathsf{res}}^{\mathcal{G}_0}.t$ as t.

While assuming that $\mathsf{ct}_0.a = \mathsf{ct}_1.a = a$, we compute the norm of the difference between $\mathsf{ct}_0.b$ and $\mathsf{ct}_1.b$, which are the first components of the ciphertexts before applying the differential privacy mechanism.

$$\begin{aligned}\|\mathsf{ct}_0.b - \mathsf{ct}_1.b\| &= \|(\mathsf{ct}_0.b + a\cdot(\mathsf{sk}_1, \ldots, \mathsf{sk}_k)) - (\mathsf{ct}_1.b + a\cdot(\mathsf{sk}_1, \ldots, \mathsf{sk}_k))\| \\ &= \|(m+e_0) - (m+e_1)\| = \|e_0 - e_1\| \le t + t_{\mathsf{fresh}},\end{aligned}$$

We will denote $t + t_{\mathsf{fresh}}$ as T for the rest of the proof. Since we were able to bound $\|\mathsf{ct}_0.b - \mathsf{ct}_1.b\|$ with T, we can now use Definition 9 to bound their KL divergence after post-processing.

$$D(M_T(\mathsf{ct}_0.b)|\mathsf{ct}_0.a = a||M_T(\mathsf{ct}_1.b)|\mathsf{ct}_1.a = a) \le \rho.$$

We repeat the same reasoning with decryption shares. To simplify the notation in this proof, we will denote $\mu_j^{\mathcal{G}_b}$ with $\mu_{j,b}$. While assuming that $\mathsf{ct}_0.b = \mathsf{ct}_1.b = b$ and $\mathsf{ct}_0.a = \mathsf{ct}_1.a = a$ are chosen, we compute the norm of the difference between μ_{0,i^*} and μ_{1,i^*}, which are the decryption shares before applying the differential privacy mechanism.

$$\|\mu_{0,i^*} - \mu_{1,i^*}\| = \|(a_{i^*}\cdot\mathsf{sk}_{i^*}) - (b - \sum_{j\neq i^*} a_j\cdot\mathsf{sk}_j)\| = \|e_0\| \le t.$$

This implies, thanks to Definition 9, that

$$D(M_t(\mu_{0,i^*})|(\mathsf{ct}_0.b = b, \mathsf{ct}_0.a = a)||M_t(\mu_{1,i^*})|(\mathsf{ct}_1.b = b, \mathsf{ct}_1.a = a)) \le \rho$$

From this point forward, we often use the notation $D_a(\mathcal{X}||\mathcal{Y})$ when referring to $D(\mathcal{X}|(\mathsf{ct}.a = a)||\mathcal{Y}|(\mathsf{ct}.a = a))$. We now use Lemma 2 to obtain the following inequality.

$$\begin{aligned}&D(M_t(\mu_{0,i^*}), M_T(\mathsf{ct}_0.b), \mathsf{ct}_0.a||M_t(\mu_{1,i^*}), M_T(\mathsf{ct}_1.b), \mathsf{ct}_1.a) \\ &\le \max_a D_a(M_t(\mu_{0,i^*}), M_T(\mathsf{ct}_0.b)||M_t(\mu_{1,i^*}), M_T(\mathsf{ct}_1.b)) + D(\mathsf{ct}_0.a||\mathsf{ct}_1.a)\end{aligned}$$

It is easy to show that $\mathsf{ct}_0.a_i$ are uniform random in $\mathcal{R}$ for each $i \in [k]$ because we re-randomized each entry by adding $\mathtt{Enc}(\mathsf{pk}_i, 0)$ to ct_0. This is also true for $\mathsf{ct}_1.a_i$ for each $i \neq i^*$. We can also say that $\mathsf{ct}_1.a_{i^*}$ is uniform random in $\mathcal{R}$ because it is obtained as a fresh encryption of 0. This implies that the KL divergence $D(\mathsf{ct}_0.a||\mathsf{ct}_1.a) = 0$. We can now apply Lemma 2 and obtain the following inequality.

$$\begin{aligned}&D(M_t(\mu_{0,i^*}), M_T(\mathsf{ct}_0.b), \mathsf{ct}_0.a||M_t(\mu_{1,i^*}), M_T(\mathsf{ct}_1.b), \mathsf{ct}_1.a)\\ &\leq \max_{b,a} D_{b,a}(M_t(\mu_{0,i^*})||M_t(\mu_{1,i^*})) + \max_a D_a(M_T(\mathsf{ct}_0.b)||M_T(\mathsf{ct}_1.b))\end{aligned}$$

We have already shown that ρ is an upper bound for each of these two terms, for every a and b. This means that the upper bound can be rewritten as follows.

$$D(M_t(\mu_{0,i^*}), M_T(\mathsf{ct}_0.b), \mathsf{ct}_0.a||M_t(\mu_{1,i^*}), M_T(\mathsf{ct}_1.b), \mathsf{ct}_1.a) \leq 2\rho$$

Then, we use Theorem 4 with $\mathcal{X}_\theta$ defined as a query to the oracle $\mathtt{Eval}$ of $\mathcal{G}_0$ and $\mathcal{Y}_\theta$ as a query to the oracle $\mathtt{Eval}'$.

$$\mathsf{adv}^{\mathcal{A}} \leq \frac{q}{2} \max_{\theta \in [q]} D(\mathcal{X}_\theta||\mathcal{Y}_\theta) \leq \frac{q}{2}(2\rho) = q\rho.$$

We conclude the proof by studying the bit security of $\mathcal{G}_1$. In the first phase of the game the adversary receives a rLWE encryption of $m_{i^*}^{(b)}$ under sk_{i^*} and then receives a fresh encryption of zero under sk_{i^*} for a polynomial number of times q. This implies that, if MK-CKKS is $(\lambda + 8)$-bit secure in the IND-CPA game, then $\mathcal{G}_1$ is also $(\lambda + 8)$-bit secure. Provided that $q\rho \leq 2^{-(\lambda+8)}$, we can finally relate the bit security of $\mathcal{G}_0$ with the bit security of $\mathcal{G}_1$, using Lemma 3 and obtain that $\mathcal{G}_0$ is λ-bit secure in the IND-MKHE security game with maximum q oracle queries.

□

Analysis of the post-processing noise. We give an analysis of the lost precision when modifying the MK-CKKS scheme as in Theorem 10. We instantiate the differential privacy mechanism from Definition 10 and $\rho = 2^{-\lambda-8}/q$. Considering the output of the $\mathtt{Combine}$ algorithm and that $\mathsf{ct}.t$ is expressed in the canonical infinity norm and not in the euclidean norm, we obtain that a Gaussian noise of standard deviation $2^{7/2}\sqrt{qn2^\lambda}(\mathsf{ct}.t + kt_{\mathsf{fresh}})$ and $(k-1)$ Gaussian noises of standard deviation $2^{7/2}\sqrt{qn2^\lambda}\mathsf{ct}.t$ are added to each coordinate. The additional bits of precision lost are approximately $\lambda/2 + \log_2\sqrt{q} + \log_2\sqrt{n} + 7/2 + \log_2 k + \log_2 t_{\mathsf{fresh}}$.

Parameters for MK-CKKS. Table 2 gives parameters for instantiating MK-CKKS with k parties and with a bound on the maximum number of queries of q. For the base CKKS scheme, we consider parameters such as ring dimension and ciphertext modulus from [ACC+18]. In particular, we set the ring dimension to be smaller or equal to 2^{15} and the standard deviation for fresh encryption σ_{fresh} to be 3.2.

5.4 Tightness of the Differential Privacy Parameters

By Theorem 10, it is possible to achieve λ bits of IND-MKHE-security by post-processing the outputs from $\mathtt{Eval}$ and $\mathtt{PDec}$ with a differentially private algorithm. Concretely we choose the Gaussian mechanism with Gaussian noise of variance $\sigma_{\mathsf{max}} \leftarrow \frac{\mathsf{ct}.t^2}{2\rho}$, where $\rho \leq 2^{-\lambda-8}/q$ is the privacy bound for ρ-KL differential privacy (Definition 9). We show that, using an appreciably smaller variance $\sigma_{\mathsf{s}} \ll \sigma_{\mathsf{max}}$, leads to the existence of an adversary that wins the IND-MKHE schemes with a non-negligible probability. In other words, we show that the noise parameters are tight when using the Gaussian mechanism, and the added Gaussian noise must be exponential in the security parameter.

Table 2: Bits of additional Gaussian noise added in the modified MK-CKKS of Theorem 10 to achieve 128-bits of IND-MKHE-security.

		Number of Parties			
		$k=2$	$k=2^2$	$k=2^3$	$k=2^5$
Max Queries	$q=1$	81.13	82.13	83.13	85.13
	$q=2^5$	83.64	84.64	85.64	87.64
	$q=2^{10}$	86.14	87.14	88.14	90.14

The adversary that we construct exploits the noise growth in the `Eval` algorithm. This noise growth follows the rules of the following lemma.

Lemma 5 (Appendix C.3 of [CDKS19])**.** *Let* $\mathsf{ct}_i = \mathsf{MK\text{-}CKKS}.\mathtt{Enc}(\mathsf{pk}, m_i)$ *for* $i \in \{0,1\}$ *and their ciphertext error be, respectively,* $\mathtt{Error}(\mathsf{sk}, ct_i, m_i) = e_i$*. The ciphertext error of the sum of both ciphertexts is equal to* $e_0 + e_1$ *and the ciphertext error of their product is equal to* $m_0e_1 + m_1e_0 + e_0e_1 + e_{\mathsf{mult}} + e_{\mathsf{lin}}$*, where the term* e_{mult} *depends on the parameters of the scheme and on the two ciphertexts.*

Algorithm 5: Adversary $\mathcal{A}(\lambda)$.

```
Data: A security parameter $\lambda$. The adversary has oracle access to $\mathtt{Eval}_{\sigma_\mathsf{s}}$.
begin
    $\mathsf{pp} \leftarrow \mathtt{Setup}(\lambda, d)$;
    $[r'_i \overset{\$}{\leftarrow} \mathcal{U}]$ ;
    $[(\mathsf{sk}_i, \mathsf{pk}_i) \leftarrow \mathtt{KeyGen}(\mathsf{pp}, r'_i)]_{i\in[2]}$;
    $i^* \leftarrow 1$;
    $[r_i \overset{\$}{\leftarrow} \mathcal{U}]_{i\in[2]}$;
    $(m_1^{(0)}, m_2^{(0)}), (m_1^{(1)}, m_2^{(1)}) \leftarrow (0, B), (B, B)$ ;
    $C \leftarrow x_1 \cdot x_2 - B \cdot x_1$ ;
    $\mathsf{ct} \leftarrow \mathtt{Enc}(\mathsf{pk}_1, m_1^{(b)}, r_1)$;
    $\tilde{\mathsf{ct}} \leftarrow \mathtt{Enc}(\mathsf{pk}_2, m_2^{(b)}, r_2)$;
    $\tilde{e} \leftarrow \mathtt{Dec}(\mathsf{sk}_2, \mathsf{ct}_2) - B$ ;
    $\mathsf{ct}_{\mathsf{res}}, \mu_1, \mu_2 \leftarrow \mathcal{O}^{\mathtt{Eval}_{\sigma_\mathsf{s}}}(\{\mathsf{pk}_i\}_{i\in[2]}, C, \mathsf{ct}, \tilde{\mathsf{ct}})$ ;
    $e_{\mathsf{res}} \leftarrow \mathtt{Combine}(\mu_1, \mathtt{PDec}(\mathsf{sk}_2, \mathsf{ct}_{\mathsf{res}}), \mathsf{ct}_{\mathsf{res}})$ ;
    Choose $I \in \{0, \ldots, n-1\}$ such that $|\tilde{e}_I|$ is maximal ;
    If $|e_{\mathsf{res},I} - B\tilde{e}_I| \geq |e_{\mathsf{res},I}|$ then return 0. Otherwise output 1 ;
```

Theorem 11. *Let* $\sigma_\mathsf{s} > 0$*. Let* $\mathtt{Eval}_{\sigma_\mathsf{s}}$ *and* $\mathtt{PDec}_{\sigma_\mathsf{s}}$ *be the modified* MK-CKKS *algorithms we presented as Algorithm 3 and as Algorithm 4 but where the post-processing noise are sampled from* $\mathcal{N}_{\mathbb{Z}^n}(0, \sigma_\mathsf{s}^2 \mathsf{ct}.t^2 I_n)$*. Let* σ *be the standard deviation of the underlying rLWE error. Then there exists an adversary* $\mathcal{A}$ *(Algorithm 5)against* $\mathsf{MK\text{-}CKKS}_{\sigma_\mathsf{s}}$ *in the* IND-MKHE*-security game such that* $\mathsf{adv}^{\mathcal{A}} = \Omega\left(\frac{1}{\sigma_\mathsf{s}^2\sigma^2 n^3}\right)$*.*

Proof. The high-level idea is as in the proof of Theorem 7. The main difference between the two proofs is that the adversary cannot compute the error after the homomorphic evaluation of the circuit because it depends from the encrypted message of the non-corrupted party. Nonetheless, using the ring structure of $\mathcal{R}$ and the circuit $x_1x_2 - Bx_2$, we are still able to rewrite the error as a sample of a Gaussian distribution where mean and variance only depend from the encrypted message and variables known by the challenger. Finally, we compute the statistical distance between the two Gaussian distributions linked to the two possible messages and use this distance to obtain a lower bound on the adversary's advantage.

The adversary knows the exact error $\tilde{e} := \tilde{\mathsf{ct}}.e$ and obtains the resulting error e_{res} after post-processing. We denote as $e \leftarrow \mathcal{N}_{\mathbb{Z}^n}(0, \sigma^2 I_n)$ the exact error of ct. Recalling the error growth rule of $\mathsf{MK\text{-}CKKS}$, we can estimate the two possible outputs for $b \in \{0,1\}$. The resulting error after computing $x \cdot y$ is equal to $e\tilde{e} + m_b\tilde{e} + Be + e_{\mathsf{mult}}$. When subtracting $B \cdot x$ in the evaluation, we also subtract Be from the error and we obtain that the error in the output of the oracle $\mathsf{ct}_{\mathsf{res}}$ is $e\tilde{e} + m_b\tilde{e} + e_{\mathsf{mult}} + e^{(1)}_{\mathsf{sm}}$ where the $e^{(1)}_{\mathsf{sm}}$ is the post-processing noise of $\mathtt{Eval}_{\sigma_{\mathsf{s}}}$. When we compute the decryption of $\mathsf{ct}_{\mathsf{res}}$ using the $\mathtt{Combine}$ algorithm, we obtain that the result is

$$e_{\mathsf{res}} = e\tilde{e} + m_b\tilde{e} + e_{\mathsf{mult}} + e^{(1)}_{\mathsf{sm}} + e^{(2)}_{\mathsf{sm}},$$

where $e^{(2)}_{\mathsf{sm}}$ is the post-processing noise of $\mathtt{PDec}_{\sigma_{\mathsf{s}}}$. Referring to the i-th coefficient of e and $\tilde{e}$ as e_i and as $\tilde{e}_i$, we can rewrite e_{res} as follows.

$$\begin{aligned} e_{\mathsf{res}} &= \sum_{i=0}^{n-1}\left(\sum_{j=0}^{i}\tilde{e_j}e_{i-j} - \sum_{j=i}^{n-1}\tilde{e}_j e_{n+i-j} + m_b\tilde{e}_i\right)x^i + e_{\mathsf{mult}} + e^{(1)}_{\mathsf{sm}} + e^{(2)}_{\mathsf{sm}} \\ &:= \sum_{i=0}^{n-1} E_i x^i + e_{\mathsf{mult}} + e^{(1)}_{\mathsf{sm}} + e^{(2)}_{\mathsf{sm}} \end{aligned}$$

The adversary analyzes the polynomial $\tilde{e}$ and chooses I as the component where the absolute value $|\tilde{e}_I|$ is maximal. We now focus on the I-th coefficient of e_{res} and, in particular, on E_I. The term E_I is an affine combination of $\{e_i\}_{i=0}^{n-1}$ that are independently sampled from $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$ with coefficients that are known to the adversary. This implies that E_I is a sample from the Gaussian $\mathcal{N}_{\mathbb{Z}}(m_b\tilde{e_I},\ \sum_{i=0}^{n-1}\tilde{e_i}^2\sigma^2)$. To estimate the total variation distance, we assume that e_{mult} and e_{lin} are significantly smaller than the other terms (Lemma 5) and that, considering the scope of this paper and the asymptotical nature of our results, we can omit them; this approximation allows us to express $e_{\mathsf{res},I}$ as a sample from the following Gaussian distribution.

$$\mathcal{N}_{\mathbb{Z}}(m_b\tilde{e_I}, \sum_{i=0}^{n-1}\tilde{e_i}^2\sigma^2 + 2\sigma_{\mathsf{s}}^2\mathsf{ct}.t^2).$$

Obtaining that $|e_{\mathsf{res},I} - B\tilde{e_I}| < |e_{\mathsf{res},I}|$ is more likely when $b = 1$ while, if $|e_{\mathsf{res},I} - B\tilde{e_I}| \geq |e_{\mathsf{res},I}|$, it is at least more likely that $b = 0$ rather than $b = 1$. To compute the advantage of this adversary in distinguishing these distributions, we need to compute the total variation distance between them. Computing this quantity for discrete Gaussian is not easy; therefore, we will approximate it by considering their counterparts on the real numbers. We define $V := \sqrt{\|\tilde{e}\|_2^2\sigma^2 + 2\sigma_{\mathsf{s}}^2\mathsf{ct}.t^2}$ and use Lemma 3 to obtain the following lower bound.

$$\Delta(\mathcal{N}(0,V), \mathcal{N}(B\tilde{e_I}, V)) \geq \frac{1}{50}\frac{B|\tilde{e_I}|}{\sqrt{V}} = \Theta\left(\frac{B|\tilde{e_I}|}{\sqrt{\|\tilde{e}\|_2^2 + 2\sigma_{\mathsf{s}}^2\mathsf{ct}.t^2}}\right)$$

The advantage of the adversary in the $\mathsf{IND\text{-}MKHE}$ game is the square of the total variation distance we just estimated which is $\Theta\left(\frac{B^2|\tilde{e_I}|^2}{\|\tilde{e}\|_2^2 + 2\sigma_{\mathsf{s}}^2\mathsf{ct}.t^2}\right)$.

With high probability $|\tilde{e}_I| \geq 1$ and $\|\tilde{e}\|_{\mathsf{can}} \leq \sigma n$. This implies that $\|\tilde{e}\|_2^2 \leq \sigma^2 n^3$ and also that $\mathsf{ct}.t \leq O(B\sigma n^{3/2})$. Putting together all these bounds, we obtain that the advantage of the adversary is $\Omega\left(\frac{B^2}{\sigma^4 n^3 + 2\sigma_{\mathsf{s}}^2 B^2 \sigma^2 n^3}\right) = \Omega\left(\frac{1}{\sigma_{\mathsf{s}}^2 \sigma^2 n^3}\right)$.

□

Theorem 12. *If the scheme* MK-CKKS *with the modified evaluation* $\mathtt{Eval}_{\sigma_{\mathsf{s}}}$ *and the modified partial decryption* $\mathtt{PDec}_{\sigma_{\mathsf{s}}}$ *is* λ*-bit* IND-MKHE*-secure, then* $\sigma_{\mathsf{s}} = \Omega(2^{\lambda/2}/\sigma n^{3/2})$*, i.e. one must add at least* $\lambda/2 - \tilde{\Omega}(\sigma n^{3/2})$ *bits of additional Gaussian noise to the standard* MK-CKKS *operations in order to achieve* IND-MKHE *security.*

Proof. By using the definition of bit security, we know that

$$\lambda \leq \log_2 O(\frac{T(A)}{\mathsf{adv}^A}) \leq \log_2 O(\sigma_s^2 \sigma^2 n^3).$$

This means that $\sigma_s \geq 2^{\lambda/2}/(\sigma n^{3/2})$ and $\lambda/2 - \log_2 \Omega(\sigma n^{3/2}) \leq \log_2 \sigma_s$. □

6 Conclusion and Open Problems

In this paper, we introduced formal models for the study of circuit privacy in the FHE approximate setting. We included the first security analysis for approximate multikey homomorphic encryption and approximate threshold homomorphic encryption that considers the knowledge of partial decryptions.

We presented a modified version of the CKKS scheme (Theorem 5) that is able to achieve λ-bit IND-CFA-security by post-processing the ciphertext with $\lambda/2 + \tilde{\mathcal{O}}(1)$ bits of noise. Additionally, we modified the MK-CKKS scheme (Theorem 10) to achieve λ-bit IND-MKHE-security. We did this by post-processing the ciphertext and the decryption shares with $\lambda/2 + \tilde{\mathcal{O}}(1)$ bits of noise. We proved that these bounds are essentially tight by providing adversaries for when only $\lambda/2 - \tilde{\Omega}(1)$ bits of noise are added.

Our work investigates Circuit Privacy for HE schemes in the approximate setting and *sanitizes* ciphertexts by applying KL differential privacy mechanisms. It would be interesting to investigate possible relations between the recent *funcCPA*-security definition [AGHV22] and the approximate setting. Another possible direction is to study the impacts of the new security definitions we introduced on exact schemes, like [CCP+24] did with IND-CPA$^{\mathsf{D}}$-security.

References

[ABSdV19] Mark Abspoel, Niek J. Bouman, Berry Schoenmakers, and Niels de Vreede. Fast secure comparison for medium-sized integers and its application in binarized neural networks. In Mitsuru Matsui, editor, *Topics in Cryptology – CT-RSA 2019*, volume 11405 of *Lecture Notes in Computer Science*, pages 453–472, San Francisco, CA, USA, March 4–8, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-12612-4_23.

[ACC+18] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018.

[ACLS18] Sebastian Angel, Hao Chen, Kim Laine, and Srinath T. V. Setty. PIR with compressed queries and amortized query processing. In *2018 IEEE Symposium on Security and Privacy*, pages 962–979, San Francisco, CA, USA, May 21–23, 2018. IEEE Computer Society Press. doi:10.1109/SP.2018.00062.

[AGHV22] Adi Akavia, Craig Gentry, Shai Halevi, and Margarita Vald. Achievable CCA2 relaxation for homomorphic encryption. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022: 20th Theory of Cryptography Conference, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 70–99, Chicago, IL, USA, November 7–10, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-22365-5_3.

[AJJM20] Prabhanjan Ananth, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Multi-key fully-homomorphic encryption in the plain model. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020: 18th Theory of Cryptography Conference, Part I*, volume 12550 of *Lecture Notes in Computer Science*, pages 28–57, Durham, NC, USA, November 16–19, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-64375-1_2.

[ALP+21] Asra Ali, Tancrède Lepoint, Sarvar Patel, Mariana Raykova, Phillipp Schoppmann, Karn Seth, and Kevin Yeo. Communication–Computation trade-offs in PIR. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1811–1828. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/ali.

[AMBFK15] Carlos Aguilar-Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. Xpir: Private information retrieval for everyone. *Proceedings on Privacy Enhancing Technologies*, 2016(2):155–174, December 2015. URL: http://dx.doi.org/10.1515/popets-2016-0010, doi:10.1515/popets-2016-0010.

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part I*, volume 12105 of *Lecture Notes in Computer Science*, pages 79–109, Zagreb, Croatia, May 10–14, 2020. Springer, Cham, Switzerland. doi:10.1007/978-3-030-45721-1_4.

[BDPMW16] Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. *FHE Circuit Privacy Almost for Free*, page 62–89. Springer Berlin Heidelberg, 2016. URL: http://dx.doi.org/10.1007/978-3-662-53008-5_3, doi:10.1007/978-3-662-53008-5_3.

[BGG+18] Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. *Threshold Cryptosystems from Threshold Fully Homomorphic Encryption*, page 565–596. Springer International Publishing, 2018. URL: http://dx.doi.org/10.1007/978-3-319-96884-1_19, doi:10.1007/978-3-319-96884-1_19.

[BGGJ20] Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. Chimera: Combining ring-lwe-based fully homomorphic encryption schemes. *Journal of Mathematical Cryptology*, 14(1):316–338, August 2020. URL: http://dx.doi.org/10.1515/jmc-2019-0026, doi:10.1515/jmc-2019-0026.

[BGPG20] Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser. Secure large-scale genome-wide association studies using homomorphic encryption. *Proceedings of the National Academy of Sciences*, 117(21):11608–11613, May 2020. URL: http://dx.doi.org/10.1073/pnas.1918257117, doi:10.1073/pnas.1918257117.

[BLR+18] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, April 2018. doi:10.1007/s00145-017-9265-9.

[BP16] Zvika Brakerski and Renen Perlman. Lattice-based fully dynamic multi-key FHE with short ciphertexts. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 190–213, Santa Barbara, CA, USA, August 14–18, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-53018-4_8.

[CCH+24] Anamaria Costache, Benjamin R. Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, and Rachel Player. On the precision loss in approximate homomorphic encryption. In Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen, editors, *SAC 2023: 30th Annual International Workshop on Selected Areas in Cryptography*, volume 14201 of *Lecture Notes in Computer Science*, pages 325–345, Fredericton, Canada, August 14-18, 2024. Springer, Cham, Switzerland. doi:10.1007/978-3-031-53368-6_16.

[CCP+24] Jung Hee Cheon, Hyeongmin Choe, Alain Passelègue, Damien Stehlé, and Elias Suvanto. Attacks against the IND-CPA$^{\mathrm{D}}$ security of exact FHE schemes. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024: 31st Conference on Computer and Communications Security*, pages 2505–2519, Salt Lake City, UT, USA, October 14–18, 2024. ACM Press. doi:10.1145/3658644.3690341.

[CCS19] Hao Chen, Ilaria Chillotti, and Yongsoo Song. Multi-key homomorphic encryption from TFHE. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part II*, volume 11922 of *Lecture Notes in Computer Science*, pages 446–472, Kobe, Japan, December 8–12, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-34621-8_16.

[CDKS19] Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019: 26th Conference on Computer and Communications Security*, pages 395–412, London, UK, November 11–15, 2019. ACM Press. doi:10.1145/3319535.3363207.

[CdWM+17] Hervé Chabanne, Amaury de Wargny, Jonathan Milgram, Constance Morel, and Emmanuel Prouff. Privacy-preserving classification on deep neural network. Cryptology ePrint Archive, Report 2017/035, 2017. URL: https://eprint.iacr.org/2017/035.

[CHK22] Henry Corrigan-Gibbs, Alexandra Henzinger, and Dmitry Kogan. Single-server private information retrieval with sublinear amortized time. In Orr

Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology – EUROCRYPT 2022, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 3–33, Trondheim, Norway, May 30 – June 3, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-07085-3_1.

[CKKS17] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437, Hong Kong, China, December 3–7, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-70694-8_15.

[CLR17] Hao Chen, Kim Laine, and Peter Rindal. Fast private set intersection from homomorphic encryption. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 1243–1255, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134061.

[CM15] Michael Clear and Ciaran McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *Advances in Cryptology – CRYPTO 2015, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 630–656, Santa Barbara, CA, USA, August 16–20, 2015. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-48000-7_31.

[CO17] Wutichai Chongchitmate and Rafail Ostrovsky. Circuit-private multi-key FHE. In Serge Fehr, editor, *PKC 2017: 20th International Conference on Theory and Practice of Public Key Cryptography, Part II*, volume 10175 of *Lecture Notes in Computer Science*, pages 241–270, Amsterdam, The Netherlands, March 28–31, 2017. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-54388-7_9.

[CZW17] Long Chen, Zhenfeng Zhang, and Xueqing Wang. Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension. In Yael Kalai and Leonid Reyzin, editors, *TCC 2017: 15th Theory of Cryptography Conference, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 597–627, Baltimore, MD, USA, November 12–15, 2017. Springer, Cham, Switzerland. doi:10.1007/978-3-319-70503-3_20.

[DGBL$^+$16] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. ICML'16, page 201–210. JMLR.org, 2016.

[DMR18] Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional gaussians with the same mean, 2018. URL: https://arxiv.org/abs/1810.08693, doi:10.48550/ARXIV.1810.08693.

[DS16] Léo Ducas and Damien Stehlé. Sanitization of FHE ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 294–310, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-49890-3_12.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, MD, USA, May 31 – June 2, 2009. ACM Press. doi:10.1145/1536414.1536440.

[GH19] Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 438–464, Nuremberg, Germany, December 1–5, 2019. Springer, Cham, Switzerland. doi:10.1007/978-3-030-36033-7_17.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867, Santa Barbara, CA, USA, August 19–23, 2012. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-32009-5_49.

[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*, volume 6223 of *Lecture Notes in Computer Science*, pages 155–172, Santa Barbara, CA, USA, August 15–19, 2010. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-14623-7_9.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd Annual ACM Symposium on Theory of Computing*, pages 736–749, Virtual Event, Italy, June 21–25, 2021. ACM Press. doi:10.1145/3406325.3451070.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92, Santa Barbara, CA, USA, August 18–22, 2013. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-642-40041-4_5.

[HFH99] Bernardo A. Huberman, Matt Franklin, and Tad Hogg. Enhancing privacy and trust in electronic communities. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC ’99, page 78–86, New York, NY, USA, 1999. Association for Computing Machinery. doi:10.1145/336992.337012.

[HHCG^{+}23] Alexandra Henzinger, Matthew M Hong, Henry Corrigan-Gibbs, Sarah Meiklejohn, and Vinod Vaikuntanathan. One server for the price of two: Simple and fast {Single-Server} private information retrieval. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 3889–3905, 2023.

[IP07] Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 575–594, Amsterdam, The Netherlands, February 21–24, 2007. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-540-70936-7_31.

[JKLS18] Xiaoqian Jiang, Miran Kim, Kristin E. Lauter, and Yongsoo Song. Secure outsourced matrix computation and application to neural networks. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018: 25th Conference on Computer and Communications Security*, pages 1209–1222, Toronto, ON, Canada, October 15–19, 2018. ACM Press. doi:10.1145/3243734.3243837.

[JVC18] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. GAZELLE: A low latency framework for secure neural network inference. In William Enck and Adrienne Porter Felt, editors, *USENIX Security 2018: 27th USENIX Security Symposium*, pages 1651–1669, Baltimore, MD, USA, August 15–17, 2018. USENIX Association.

[KKL+23] Taechan Kim, Hyesun Kwak, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Asymptotically faster multi-key homomorphic encryption from homomorphic gadget decomposition. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023: 30th Conference on Computer and Communications Security*, pages 726–740, Copenhagen, Denmark, November 26–30, 2023. ACM Press. doi:10.1145/3576915.3623176.

[Klu22] Kamil Kluczniak. NTRU-v-um: Secure fully homomorphic encryption from NTRU with small modulus. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022: 29th Conference on Computer and Communications Security*, pages 1783–1797, Los Angeles, CA, USA, November 7–11, 2022. ACM Press. doi:10.1145/3548606.3560700.

[KS22] Kamil Kluczniak and Leonard Schild. Fdfb: Full domain functional bootstrapping towards practical fully homomorphic encryption. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2023(1):501–537, Nov. 2022. URL: https://tches.iacr.org/index.php/TCHES/article/view/9960, doi:10.46586/tches.v2023.i1.501-537.

[KSK+18] Andrey Kim, Yongsoo Song, Miran Kim, Keewoo Lee, and Jung Hee Cheon. Logistic regression model training based on the approximate homomorphic encryption. *BMC Medical Genomics*, 11(S4), October 2018. URL: http://dx.doi.org/10.1186/s12920-018-0401-7, doi:10.1186/s12920-018-0401-7.

[KSK+20] Duhyeong Kim, Yongha Son, Dongwoo Kim, Andrey Kim, Seungwan Hong, and Jung Hee Cheon. Privacy-preserving approximate gwas computation based on homomorphic encryption. *BMC Medical Genomics*, 13(S7), July 2020. URL: http://dx.doi.org/10.1186/s12920-020-0722-1, doi:10.1186/s12920-020-0722-1.

[LJLA17] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via MiniONN transformations. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017: 24th Conference on Computer and Communications Security*, pages 619–631, Dallas, TX, USA, October 31 – November 2, 2017. ACM Press. doi:10.1145/3133956.3134056.

[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, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 648–677, Zagreb, Croatia, October 17–21, 2021. Springer, Cham, Switzerland. doi:10.1007/978-3-030-77870-5_23.

[LMSS22] Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part I*, volume 13507 of *Lecture Notes in Computer Science*, pages 560–589, Santa Barbara, CA, USA, August 15–18, 2022. Springer, Cham, Switzerland. doi:10.1007/978-3-031-15802-5_20.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *44th Annual ACM Symposium on Theory of Computing*, pages 1219–1234, New York, NY, USA, May 19–22, 2012. ACM Press. doi:10.1145/2213977.2214086.

[Mea86] Catherine Meadows. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In *1986 IEEE Symposium on Security and Privacy*, pages 134–134, 1986. doi:10.1109/SP.1986.10022.

[Mir17] Ilya Mironov. Rényi differential privacy. In Boris Köpf and Steve Chong, editors, *CSF 2017: IEEE 30th Computer Security Foundations Symposium*, pages 263–275, Santa Barbara, CA, USA, August 21–25, 2017. IEEE Computer Society Press. doi:10.1109/CSF.2017.11.

[MW16] Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 735–763, Vienna, Austria, May 8–12, 2016. Springer Berlin Heidelberg, Germany. doi:10.1007/978-3-662-49896-5_26.

[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, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 3–28, Tel Aviv, Israel, April 29 – May 3, 2018. Springer, Cham, Switzerland. doi:10.1007/978-3-319-78381-9_1.

[MW22] Samir Jordan Menon and David J. Wu. SPIRAL: Fast, high-rate single-server PIR via FHE composition. In *2022 IEEE Symposium on Security and Privacy*, pages 930–947, San Francisco, CA, USA, May 22–26, 2022. IEEE Computer Society Press. doi:10.1109/SP46214.2022.9833700.

[PW25] Yury Polyanskiy and Yihong Wu. *Information theory: From coding to learning*. Cambridge university press, 2025.

[QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In Mikkel Thorup, editor, *59th Annual Symposium on Foundations of Computer Science*, pages 859–870, Paris, France, October 7–9, 2018. IEEE Computer Society Press. doi:10.1109/FOCS.2018.00086.

[RSC$^+$19] M. Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin E. Lauter, and Farinaz Koushanfar. XONN: XNOR-based oblivious deep neural network inference. In Nadia Heninger and Patrick Traynor, editors, *USENIX Security 2019: 28th USENIX Security Symposium*, pages 1501–1518, Santa Clara, CA, USA, August 14–16, 2019. USENIX Association.

[Sma22] Nigel Smart. Description of the folklore construction. https://www.reddit.com/r/privacy/comments/yp5enz/comment/ivhkxcx/?rdt=46430, 2022.

[Val76] Leslie G. Valiant. Universal circuits (preliminary report). In *Proceedings of the eighth annual ACM symposium on Theory of computing - STOC '76*, STOC '76, page 196–203. ACM Press, 1976. URL: http://dx.doi.org/10.1145/800113.803649, doi:10.1145/800113.803649.

A Improving parameters with Relaxed Bit Security

In [LMSS22], Li et al. introduced a relaxation of the bit security definition and showed how relaxed $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ can be achieved in approximate HE schemes with less demanding amounts of noise.

Informally, a primitive is (c, s)-bit secure if, for any adversary $\mathcal{A}$, either $\mathcal{A}$ has less than 2^{-s} statistical advantage, or the running time of the attack is at least 2^c times greater than the advantage achieved.

We recall the formal definition of Relaxed Bit Security.

Definition 25 (Relaxed Bit Security, Definition 19 of [LMSS22])**.** Let $\mathcal{G}$ be an indistinguishability game. Let $\mathsf{adv}^{\mathcal{A}}$ be the advantage of an adversary $\mathcal{A}$ against $\mathcal{G}$, as in Definition 12. We say that the indistinguishability game $\mathcal{G}$ is (c, s)-bit secure if, for any adversary $\mathcal{A}$, either

$$\log_2 \frac{T(\mathcal{A})}{\mathsf{adv}^{\mathcal{A}}} \geq c \quad \text{or} \quad \log_2 \frac{1}{\mathsf{adv}^{\mathcal{A}}} \geq s.$$

This definition expresses two different security parameters: a computational one (c) and a statistical one (s). When choosing $s < c$, the notion of security becomes more permissive than standard bit security (Definition 12); however, this relaxation and the additional allowed statistical attacks can be accurately described and analyzed.

When using statistical techniques on a computational primitive, this finer grained definition allows to tailor the desired achieved security depending on the application. In our case, to achieve (c, s)-bits of $\mathsf{IND\text{-}MKHE}$-security, the amount of added noise depends on the statistical parameter s and not on the computational parameter c. This allows us to decrease the cost of our post-processing phase in Algorithm 3 and 4, saving around $(c - s)/2$ bits of Gaussian noise.

Theorem 13. *Let* $\mathsf{MK\text{-}CKKS} = (\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{PDec}, \mathtt{Combine})$ *be the* $\mathsf{MK\text{-}CKKS}$ *multikey homomorphic encryption scheme, with plaintext space* $\mathcal{R}$ *and estimate function* $\mathtt{Estimate}$*. Let* $q \in \mathbb{N}$*. Let* M_t *be a* ρ*-KLDP mechanism on* $\mathcal{R}$*. If* $\mathsf{MK\text{-}CKKS}.\mathtt{Enc}$ *is* λ*-bit secure in the* $\mathsf{IND\text{-}CPA}$ *game, then* $\mathsf{MK\text{-}CKKS}$ *with the modified* $\mathsf{MK\text{-}CKKS}.\mathtt{Eval}'$ *given by Algorithm 3 and with the modified* $\mathsf{MK\text{-}CKKS}.\mathtt{PDec}'$ *given by Algorithm 4 is* $(\lambda - \log_2 24, \log_2(1/\rho) - \log_2 q - \log_2 24)$*-bit secure in the* $\mathsf{IND\text{-}MKHE}$ *game where* q *is the maximum amount of oracle queries by the adversary.*

Proof. The proof in ([LMSS22], Appendix F) can be easily adapted to this theorem just by considering, as games $\mathcal{G}_0$ and $\mathcal{G}_1$, the games that we used in the proof of Theorem 10. □

Parameters for $\mathsf{MK\text{-}CKKS}$ with relaxed bit security. We provide concrete parameters for instantiating $\mathsf{MK\text{-}CKKS}$ with k parties and a statistical security parameter λ_{s}. For the base CKKS scheme, we consider parameters such as ring dimension and ciphertext modulus from [ACC⁺18]. In particular, we set the ring dimension to be smaller or equal to 2^{15} and the standard deviation for fresh encryption σ_{fresh} to be 3.2.

The choice of the appropriate statistical security parameter strongly depends from the desired application and we refer to ([LMSS22], Subsections 4.4 and 4.5) for a more in-depth discussion on parameters choice and on Definition 25.

Table 3: Bits of additional Gaussian noise added in the modified MK-CKKS of Theorem 10 to achieve (128,λ_{s})-bits of IND-MKHE-security, with a bound on the maximum number of queries of 2^{10}.

λ_{s} \ k	2	2^2	2^3	2^5
128	86.14	87.14	88.14	90.14
112	78.14	79.14	80.14	82.14
96	70.14	71.14	72.14	74.14
80	62.14	63.14	64.14	66.14