

# On Circuit Private, Multikey and Threshold Approximate Homomorphic Encryption

Kamil Kluczniak and Giacomo Santato

CISPA Helmholtz Center for Information Security  
{kamil.kluczniak,giacomo.santato}@cispa.de

**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 hype for CKKS-type encryption stems from its potential to be used in inference or multiparty computation tasks that do not require the exact output, for example, inference and training of machine learning models. 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.

## 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 function  $F$ , returns a ciphertext that decrypts to  $F(m)$ . The concept of FHE was first introduced by Rivest and Dertouzos [RAD78] and later realized by Gentry [Gen09b].

FHE has numerous applications in cryptography. Among others, it is used to build private information retrieval [ABFK16, ALP<sup>+</sup>21, ACLS18, GH19, CHK22, MW22, HHC<sup>+</sup>22], 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 function  $F$ .

Circuit privacy, sometimes called function privacy, is a critical property in FHE, where the ciphertext produced by the server, computing a function  $F$  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<sup>+</sup>16, CdWM<sup>+</sup>17, LJLA17, JKLS18, JVC18, BGGJ18, ABSdV19, CDKS19, RSC<sup>+</sup>19, BGPG20, KS22], analysis of genomic data [KSK<sup>+</sup>18, KSK<sup>+</sup>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<sup>+</sup>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 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 function 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. Then, 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 IND-CPA<sup>D</sup>-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 polynomial 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<sup>+</sup>22]. 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 similar 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. We note that we can easily adapt our lower bounds to the “exact” setting. Our result in this manner is especially relevant due to the following.

- There are several recent proposals [DWF22, CSS<sup>+</sup>22, BS23] to use a noise, bounded by a polynomial in the security parameter, to implement partial decryption. The idea is to make an analysis based on Rényi divergence. Indeed, in some situations, analysis using the Rényi divergence may result in better parameters [BLR<sup>+</sup>18]. Our lower bounds show that there seem to be issues with the security analysis in [DWF22, CSS<sup>+</sup>22]. As [BS23] give the security proof with respect to a new model, we suspect that their security definition may not be suitable for many applications of threshold encryption.
- There is a folklore belief that circuit privacy can be accomplished via multikey (F)HE. The idea is that the server encrypts the circuit with its key, and a client encrypts the query with its key. Then the server computes a universal circuit over both ciphertexts and returns a partial decryption of the evaluated ciphertext back to the client. If the multikey/threshold encryption with partial decryptions gives us a secure MPC protocol, then this approach seems to be correct. Our analysis and lower bounds for the approximate arithmetic setting show that we can indeed use the folklore solution. However, encrypting the circuit does not seem helpful in reducing the flooding noise significantly in comparison to just sanitizing single-key homomorphic encryption.
- Our results lead to tight estimates of the precision when applying the differential privacy mechanism to CKKS and its multikey/threshold versions.

## 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 function 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<sup>+</sup>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 recall some notions and known results.

### 2.1 Notation

We denote an  $n$  dimensional column vector as  $[f(\cdot, i)]_{i=1}^n$ , where  $f(\cdot, i)$  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  $\text{Var}(x)$  the variance of  $x$ , as  $\text{stddev}(x)$  its standard deviation and as  $\mathbf{E}(x)$  its expectation. By  $\text{Ham}(\vec{a})$  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  $\text{poly}(\cdot)$ . We denote as  $\text{negl}(\lambda)$  a negligible function in  $\lambda \in \mathbb{N}$ . That is, for any positive polynomial  $\text{poly}(\cdot)$  there

exists  $c \in \mathbb{N}$  such that for all  $\lambda \geq c$  we have  $\mathbf{negl}(\lambda) \leq \frac{1}{\text{poly}(\lambda)}$ . 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  $\text{Alg}(x)$  chooses its random coins internally. However, sometimes we write  $\text{Alg}(x; r)$  to denote that the random coins  $r \stackrel{\$}{\leftarrow} \mathcal{U}$  are used as a seed for  $\text{Alg}$ , and  $\text{Alg}(x; r)$  is deterministic.

## 2.2 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  $\text{HE}$  for a class of circuits  $\mathcal{L}$  as a tuple of four algorithms  $\text{HE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  with the following syntax.*

$\text{KeyGen}(\lambda) \rightarrow (\text{pk}, \text{sk})$ : *Given a security parameter  $\lambda$ , returns a public key  $\text{pk}$  and a secret key  $\text{sk}$ .*

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

$\text{Eval}(\text{pk}, C, \text{ct}_1, \dots, \text{ct}_k) \rightarrow c$ : *Given a public key  $\text{pk}$ , a circuit  $C \in \mathcal{L}$  and ciphertexts  $\text{ct}_1, \dots, \text{ct}_k$ , returns a ciphertext  $c$ .*

$\text{Dec}(\text{sk}, \text{ct}) \rightarrow m$ : *Given a secret key  $\text{sk}$  and a ciphertext  $\text{ct}$ , 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 an homomorphic encryption scheme  $\text{HE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  is correct if for all  $C \in \mathcal{L}$ , all  $m_1, \dots, m_k \in \mathcal{M}$  and for all  $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(\lambda)$ , we have that*

$$\Pr[\text{Dec}(\text{sk}, \text{Eval}(\text{pk}, C, \text{ct}_1, \dots, \text{ct}_k)) \neq C(m_1, \dots, m_k)] \leq \mathbf{negl}(\lambda),$$

*where  $m_i = \text{Dec}(\text{sk}, \text{ct}_i)$  for  $i \in [k]$ .*

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  $\text{HE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  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*

$(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(\lambda)$ , any ciphertext  $\text{ct} \in \mathcal{C}$  and message  $m \in \mathcal{M}$  the ciphertext error is defined as

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

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

**Definition 4 (Approximate Correctness [LMSS22]).** Let  $\text{HE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  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  $\text{Estimate} : \bigsqcup_{n \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}_{\geq 0}^k \mapsto \mathbb{R}_{\geq 0}$  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  $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(\lambda)$ , if  $\text{ct}_1, \dots, \text{ct}_k$  and  $m_1, \dots, m_k$  are such that  $\text{Error}(\text{sk}, \text{ct}_i, m_i) \leq t_i$ , then

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

To compute  $\text{Estimate}$ , 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 ciphertexts. In particular, we define a *tagged ciphertext*  $\text{ct} = (\dots, t)$  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  $t_{\text{fresh}}$  by  $\text{Enc}$  when a ciphertext  $\text{ct}$  is created. After that, the value of  $\text{ct}.t$  is updated using  $\text{Estimate}$  every time that a circuit is homomorphically evaluated on  $\text{ct}$ .

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

**Definition 5 (IND-CPA-security).** Let  $\text{HE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  be a homomorphic encryption scheme. We define the IND-CPA game as the following indistinguishability game, where  $b \in \{0, 1\}$  and  $\mathcal{A}$  is an adversary.

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

where the adversary has access to an encryption oracle  $\text{E}^b(\cdot, \cdot)$  that takes as input  $m_0, m_1 \in \mathcal{M}$  and returns  $\text{Enc}(\text{pk}, m_b)$ . The scheme HE is said to be  $\lambda$ -bit IND-CPA-secure if, for any adversary  $\mathcal{A}$ , we have that  $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\text{adv}^{\mathcal{A}}}$ , where  $\text{adv}^{\mathcal{A}}$  is defined as in Definition 8.

### 2.3 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].

Given  $N$ , a positive integer, let  $\Phi_N(X) = \prod_{j \in \mathbb{Z}_N^*} (X - \omega^j)$  be the  $N$ -th cyclotomic polynomial, where  $\omega \in \mathbb{C}$  is one of the principal  $N$ -th root of unity and  $\mathbb{Z}_N^*$  is the group of invertible integers modulo  $N$  and has order  $\varphi(N)$ . We denote by  $\mathcal{R}_Q$  the ring  $\mathbb{Z}_Q[X]/(\Phi_N(X))$ , where  $\mathbb{Z}_Q$  is the ring of integers modulo  $Q$ . We will omit  $Q$  when it is clear from the context. The CKKS scheme is able to encrypt complex ciphertext by using the canonical embedding  $\tau : \mathbb{Q}[X]/(\Phi_N(X)) \rightarrow \mathbb{C}^{\varphi(N)}$ ; this embedding is defined by sending the polynomial  $a(X)$  in the tuple of its evaluations in the principal  $N$ -th complex roots of unity, so in the tuple  $(a(\omega^j))_{j \in \mathbb{Z}_N^*}$ . Moreover, the  $n = \varphi(N)$  complex values in each image come in conjugate pairs  $(a(\omega^j), a(\omega^{N-j}))$ , so it is possible to obtain a projection  $\pi$  to  $\mathbb{C}^{n/2}$  by considering only one of the two elements for every complex pair. Using this function, vectors  $z \in \mathbb{C}^{n/2}$  are considered as messages in CKKS. Complex messages are transformed to polynomials in  $\mathcal{R}$  using the inverses of  $\pi$  and  $\varphi$  on a scaled vector  $\delta \cdot z$ , for some scaling factor  $\delta \in \mathbb{R}$  such that  $\|\delta \cdot z\| \ll Q$  and then by rounding the result to a polynomial in  $\mathcal{R}$ . More in detail, the functions that link vectors in  $\mathbb{C}^{n/2}$  to plaintext polynomials in  $\mathcal{R}$  are

$$\begin{aligned} \text{CKKS.Encode}(z \in \mathbb{C}^{n/2}, \delta) &= \lfloor \delta \cdot \varphi^{-1}(\pi^{-1}(z)) \rfloor; \\ \text{CKKS.Decode}(a(X) \in \mathcal{R}, \delta) &= \pi(\varphi(\delta^{-1} \cdot a(X))). \end{aligned}$$

These two functions do not require the knowledge of any secret key nor public key. In the main implementations of CKKS they are, respectively, included in encryption and decryption but for theoretical analysis we will consider them separately. This allow us to study express more clearly the error that arise from the message encoding and to differentiate it from the other errors in this scheme.

Another useful tool to track the ciphertext error in CKKS is the norm induced on  $\mathcal{R}$  by the canonical embedding  $\pi \circ \varphi$ . This norm is defined as  $\|a\|_{\text{can}} = \|\pi \circ \varphi(a)\|_{\infty}$ .

We now give a broad description of the main algorithms in the CKKS scheme that we still have not introduced. The parameters of the scheme are: the plaintext polynomial ring  $\mathcal{R}$  with ring dimension  $N$  typically chosen as a power of two, a ciphertext modulo  $Q$  and a discrete Gaussian error  $\chi$  with standard deviation  $\sigma$ .

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

**CKKS.Enc( $\text{pk}, m \in \mathcal{R}_Q$ ):** 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  $\text{ct}.a = \text{rpk}.a + e_1$ ,  $\text{ct}.b = \text{rpk}.b + e_2 + m$  and return  $\text{ct}$ .

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

**CKKS.Add**( $\text{pk}, \text{ct}_0, \text{ct}_1 \in \mathcal{R}_Q$ ): Set  $\text{ct}.a = \text{ct}_0.a + \text{ct}_1.a$ ,  $\text{ct}.b = \text{ct}_0.b + \text{ct}_1.b$  and return  $\text{ct}$ .

**CKKS.Mul**( $\text{pk}, \text{ct}_0, \text{ct}_1 \in \mathcal{R}_Q$ ): Set  $\text{ct}.b = \text{ct}_0.b \cdot \text{ct}_1.b + \lfloor (\text{ct}_0.a \cdot \text{ct}_1.a \cdot \text{pk}.b')/p \rfloor$ , and  $\text{ct}.a = \text{ct}_0.a \cdot \text{ct}_1.b + \text{ct}_1.a \cdot \text{ct}_0.b + \lfloor (\text{ct}_0.a \cdot \text{ct}_1.a \cdot \text{pk}.a')/p \rfloor$ . Return  $\text{ct}$ .

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

We now give a brief explanation on how this estimate is handled by the algorithms of the CKKS scheme. **CKKS.Enc** assigns to the returned ciphertext an upper bound of the ciphertext error for fresh encryptions. **CKKS.Add** and **CKKS.Mul** follow the noise growth rules of Lemma 2 to assign to the returned ciphertext a noise estimate. More in general, when homomorphically evaluating a circuit  $C$  in CKKS by computing **Eval**( $\text{pk}, C, \text{ct}_1, \dots, \text{ct}_k$ ), 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.

## 2.4 Probability, Bit Security and Differential Privacy

A probability ensemble  $(\mathcal{P}_\theta)_\theta$  is a family of probability distributions parameterized by a variable  $\theta$ . 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 1 (Sub-Additivity of KL divergence for Joint Distributions, Theorem 2.2 of [PW14])**. 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.

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

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

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

**Definition 7 (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  $A$  may send (adaptively chosen) query strings  $\theta_i$  to  $C$  and obtain samples  $c_i \leftarrow \mathcal{D}_{\theta_i}^b$ . The goal of the adversary is to output  $b' = b$ .

**Definition 8 (Bit Security).** For any adversary  $A$  playing an indistinguishability game  $\mathcal{G}$ , we define its

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

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

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

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

In [LMSS22], Li et al. introduce many handy tools to use differential privacy in the approximate FHE setting. For the rest of this subsection we will recall all the ones we need for the proof of Theorem 5.

**Theorem 2.** 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  $\max_\theta D(\mathcal{P}_\theta || \mathcal{Q}_\theta) \leq 2^{-k+1}$ , then  $\mathcal{G}^{\mathcal{Q}_\theta}$  is  $(k - 8)$ -bit secure.

**Theorem 3.** 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  $A$  making at most  $q$  queries to its oracle, we have that

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

**Definition 9 (Norm KL Differential Privacy).** For  $t \in \mathbb{R}_{\geq 0}$  let  $M_t : B \rightarrow C$  be a family of randomized algorithms, where  $B$  is a normed space with norm  $\|\cdot\| : B \rightarrow \mathbb{R}_{\geq 0}$ . Let  $\rho \in \mathbb{R}$  be a privacy bound. We say that the family  $M_t$  is  $\rho$ -KL differentially private ( $\rho$ -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  $M_t : \mathbb{Z}^n \rightarrow \mathbb{Z}^n$  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 4.** For any  $\rho > 0$ ,  $n \in \mathbb{N}$ , the Gaussian mechanism is  $\rho$ -KLDP.

### 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 11 (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 Sim such that*

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

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

Definition 11 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 other words, the evaluation procedure is correct with respect to Definition 2. 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 12 (Indistinguishability Circuit Privacy).** *Let HE = (KeyGen, Enc, Eval, Dec) be a homomorphic encryption scheme for circuits in  $\mathcal{L}$ . We define the experiment  $\text{Exp}_b^{\text{IND-CP}}[\mathcal{A}]$ , where  $b \in \{0, 1\}$  is a bit and  $\mathcal{A}$  is an adversary.*

The experiment is defined as follow:

$$\text{Exp}_b^{\text{IND-CP}}[\mathcal{A}](1^\lambda) :$$

$$\begin{aligned} r, r_1, \dots, r_n &\stackrel{\$}{\leftarrow} \mathcal{U}, \\ (\text{sk}, \text{pk}) &\leftarrow \text{KeyGen}(\lambda; r), \\ m_1, \dots, m_n, C_0, C_1, \text{st} &\leftarrow \mathcal{A}(\lambda, r, r_1, \dots, r_n), \\ [\text{ct}_i &\leftarrow \text{Enc}(\text{pk}, m_i; r_i)]_{i=1}^n, \\ \text{ct} &\leftarrow \text{Eval}(\text{pk}, C_b, \text{ct}_1, \dots, \text{ct}_n), \\ b' &\leftarrow \mathcal{A}(\text{st}, \text{ct}), \\ &\textbf{return } b' \end{aligned}$$

where  $C_0, C_1 \in \mathcal{L}$  and  $C_0(m_1, \dots, m_n) = C_1(m_1, \dots, m_n)$ . The scheme HE is said to be  $\lambda$ -bit IND-CP-secure if, for any adversary  $\mathcal{A}$ , we have that  $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\text{adv}^{\mathcal{A}}}$ , where  $\text{adv}^{\mathcal{A}}$  is defined as in Definition 8.

## 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 12. 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 a superpolynomial 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 superpolynomial in the security parameter, and a significantly lower noise parameter leads to an efficient adversary against IND-CP-security.

### 4.1 IND-CP-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  $\max_{C \in \mathcal{L}} \{\text{Estimate}(C, t_{\text{fresh}}, \dots, t_{\text{fresh}})\}$  for a class of circuits  $\mathcal{L}$ ; we refer to this noise estimate as  $T_{\text{max}}$ .

**Theorem 5.** *Let  $\text{CKKS} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{Dec})$  be the CKKS approximate encryption scheme, with the normed plaintext space  $\mathcal{R}$  and estimate function*

---

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


---

**Data:** A public key  $\text{pk}$ , circuit  $C \in \mathcal{L}$ , a vector of ciphertexts  $\text{ct}_1, \dots, \text{ct}_k$ .

```

begin
   $\text{ct} \leftarrow \text{Eval}(\text{pk}, C, \text{ct}_1, \dots, \text{ct}_k)$  ;
   $\text{ct}.t \leftarrow \max_{D \in \mathcal{L}} \{\text{Estimate}(D, \text{ct}_1.t, \dots, \text{ct}_k.t)\}$  ;
   $\text{ct} \leftarrow \text{ct} + \text{Enc}(\text{pk}, 0)$  ;
   $\text{ct}.b \leftarrow M_{\text{ct}.t}(\text{ct}.b)$  ;
return  $\text{ct}$  ;

```

---

**Estimate.** Let  $M_t$  be a  $\rho$ -KLDP mechanism on  $\mathcal{R}$  where  $\rho \leq 2^{-\lambda-7}$ . Then CKKS with the modified  $\text{Eval}_{\mathcal{L}}$  given by Algorithm 1 is  $\lambda$ -bit secure in the IND-CP game for the circuit space  $\mathcal{L}$ .

*Proof (of Theorem 5).* We give a brief overview of the structure of the proof. First, we construct a new  $\lambda$ -bit secure indistinguishability game. After that, we consider the output to any adversary's query in this game and in the IND-CP 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 sub-additivity (Lemma 1) 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.

1.  $\mathcal{G}_0$ : the CKKS scheme with the evaluation algorithm given by Algorithm 1 in the IND-CP game with circuit space  $\mathcal{L}$ .
2.  $\mathcal{G}_1$ : the original CKKS scheme in a variant of the IND-CP game where the challenger returns a fresh noiseless encryption (that we denote as  $\text{Enc}_n$ ) of the result  $m_{\text{res}} = C_0(m_1, \dots, m_k) = C_1(m_1, \dots, m_k)$ . Furthermore,  $\text{ct}.b$  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:

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

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.

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

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

where  $e_0$  is the real error of the ciphertext  $\text{ct}_0$ . By definition of approximate correctness of CKKS we know that the error  $e_0$  is smaller than the ciphertext noise tag  $\text{ct}_0.t$ . Therefore,

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

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

$$D((M_t(\text{ct}_0.b)|\text{ct}_0.a = a) \parallel (M_t(\text{ct}_1.b)|\text{ct}_1.a = a)) \leq \rho.$$

We now use Lemma 1 to obtain the following inequality.

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

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

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

Then, since the KL-divergence between these two indistinguishability games is smaller than a fixed value  $\rho$  and provided that  $\rho/2 \leq 2^{-\lambda-8}$ , we can use Theorem 3 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  $\lambda$ -bit IND-CP 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  $\rho = 2^{-\lambda-7}$ . In this case, a Gaussian noise of standard deviation  $8\sqrt{2^\lambda T_{\max}}$  is added to each coordinate. We obtain that the bits of precision lost are  $\lambda/2 + 3 + \log_2(T_{\max})$ .

## 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. Remind that the usual noise estimation algorithm estimates the noise based on the circuit, which is enough for IND-CPA<sup>D</sup>-security when post-processing decryption as in [LMSS22]. To obtain circuit privacy, we estimate the noise as the maximum noise over all circuits in a given class of circuits. In particular, we run  $T_{\max} := \max_{D \in \mathcal{L}} \{\text{Estimate}(D, t_{\text{fresh}}, \dots, t_{\text{fresh}})\}$ . 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  $C(x_1, \dots, x_k)$  be a multivariate polynomial of degree smaller or equal to  $d$ . Let  $B \in \mathbb{N}$  such that  $\|m_i\|_{\text{can}} \leq B$  for  $i \in [k]$ , then*

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

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

To prove Theorem 6 we need to recall the basic expressions of noise growth during addition and multiplication in  $\text{CKKS.Eval}$  and also an heuristic on  $e_{\text{mult}}$ .

**Lemma 2 (Lemma 3 of [CKKS17]).** *Let  $\text{ct}_i = \text{CKKS.Enc}(\text{pk}, m_i)$  for  $i \in \{0, 1\}$  and their ciphertext error be, respectively,  $\text{Error}(\text{sk}, \text{ct}_i, m_i) = e_i$ . The ciphertext error of the sum of both ciphertexts is equal to  $e_0 + e_1$  and the ciphertext error of the product of both ciphertexts is equal to  $m_0 e_1 + m_1 e_0 + e_0 e_1 + e_{\text{mult}}$ , where the term  $e_{\text{mult}}$  depends on the parameters of the scheme and on the two ciphertexts  $\text{ct}_0, \text{ct}_1$ .*

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

*Proof (of Theorem 6).* In this proof we denote  $\text{Estimate}(f(x), t_{\text{fresh}})$  as  $\text{Est}(f(x))$ . Also we omit the subscript  $\text{can}$  when using the canonical norm since it is the only norm used in this proof.

First, we want to prove that  $\text{Est}(x^d) = O(dB^{d-1}t_{\text{fresh}})$  by induction. This is trivially true for  $d = 1$ . We now study the statement for  $d > 1$ .  $\text{Est}(x^d) = \text{Estimate}(x^{d-1} \cdot x) = \|m^{d-1}e + me_{d-1} + e_{d-1}e + e_{\text{mult}}\|$  where  $e_{d-1}$  is the resulting error from the evaluation of the polynomial  $x^{d-1}$ . We can bound this quantity from above by using the triangular inequality  $\text{Est}(x^i) \leq B^{d-1}\|e\| + B\|e_{d-1}\| + \|e_{d-1}e + e_{\text{mult}}\|$ . Using the inductive hypothesis  $\|e_{d-1}\| = O((d-1)B^{d-2}t_{\text{fresh}})$ , we can rewrite this quantity as  $\text{Est}(x^i) = O(B^{d-1}t_{\text{fresh}} + (d-1)B^{d-1}t_{\text{fresh}}) + \|e_{d-1}e + e_{\text{mult}}\|$ . Since  $\|e_{d-1}e + e_{\text{mult}}\| \ll B^{d-1}$  we can just conclude that  $\text{Est}(x^i) = O(dB^{d-1}t_{\text{fresh}})$ .

We can now extend our study to monomials  $x_1^{i_1} \dots x_k^{i_k}$ . We prove by induction on  $k$  that  $\mathbf{Est}(x_1^{i_1} \dots x_k^{i_k}) = O(dB^{d-1}t_{\text{fresh}})$ , where  $d = i_1 + \dots + i_k$ . This is trivially true for  $k = 1$ . We now study the statement for  $k > 1$ .  $\mathbf{Est}(x_1^{i_1} \dots x_{k-1}^{i_{k-1}} \cdot x_k^{i_k}) = \|(m_1^{i_1} \dots m_{k-1}^{i_{k-1}})e_{k-1} + m_k^{i_k}e_k + e_{k-1}e_k + e_{\text{mult}}\|$ , 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  $\mathbf{Est}(x_1^{i_1} \dots x_k^{i_k}) \leq B^{i_1 + \dots + i_{k-1}}\|e_k\| + B^{i_k}\|e_{k-1}\| + \|e_{k-1}e_k + e_{\text{mult}}\|$ . Using the inductive hypothesis on  $e_{k-1}$  and  $e_k = O(i_k B^{i_k} t_{\text{fresh}})$  we can rewrite this quantity as  $\mathbf{Est}(x_1^{i_1} \dots x_k^{i_k}) = O(B^{i_k}(i_1 + \dots + i_{k-1}) + B^{i_1 + \dots + i_{k-1}}t_{\text{fresh}}) + \|e_{k-1}e_k + e_{\text{mult}}\|$ . Since  $\|e_{k-1}e_k + e_{\text{mult}}\| \ll B^d$  we can just conclude that  $\mathbf{Est}(x_1^{i_1} \dots x_k^{i_k}) = O(dB^{d-1}t_{\text{fresh}})$  where  $d = i_1 + \dots + i_k$ . Finally, we analyze a generic multivariate polynomial with  $k$  variables and degree smaller or equal to  $d$ .

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

### 4.3 Tightness of the Differential Privacy Parameters

As shown by Theorem 5, the proposed modified version of CKKS achieves  $\lambda$ -bit IND-CP-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  $\sigma_{\text{max}} \leftarrow \frac{T_{\text{max}}^2}{2\rho}$ . Remind that  $\rho \leq 2^{-\lambda-7}$  is the privacy bound for  $\rho$ -KL differential privacy (Definition 9), and  $T_{\text{max}}$  is the noise upper bound for the class of circuits. We show that trying to use an appreciably smaller variance  $\sigma_s \ll \sigma_{\text{max}}$  leads to the existence of an adversary that wins the IND-CP game 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 superpolynomial in the security parameter.

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

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

---

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

---

**Data:** A security parameter  $\lambda$ . The adversary has oracle access to  $\text{Eval}_{\mathcal{L}^d}^{\sigma_s}$ .

**begin**

```

 $r, r_1 \xleftarrow{\$} \mathcal{U};$ 
 $(\text{sk}, \text{pk}) \leftarrow \text{KeyGen}(\lambda; r);$ 
 $m, C_0, C_1 \leftarrow B, x^d, x^d + Bx^{d-1} - B^d;$ 
 $\text{ct} \leftarrow \text{Enc}(\text{pk}, B; r_1);$ 
 $\text{ct}_{\text{res}} \leftarrow \mathcal{O}^{\text{Eval}_{\mathcal{L}^d}^{\sigma_s}(\text{pk}, \cdot, \text{ct})}(C_0, C_1);$ 
 $e_0 \leftarrow \text{Dec}(\text{sk}, \text{Eval}(C_0, \text{ct})) - B^d;$ 
 $e_1 \leftarrow \text{Dec}(\text{sk}, \text{Eval}(C_1, \text{ct})) - B^d;$ 
 $e_{\text{res}} \leftarrow \text{Dec}(\text{sk}, \text{ct}_{\text{res}}) - B^d;$ 
Choose  $i \in \{0, n-1\}$  such that  $|e_{0,i} - e_{1,i}|$  is maximal;
If  $|e_{\text{res},i} - e_{0,i}| \leq |e_{\text{res},i} - e_{1,i}|$  then return 0. Otherwise output 1;
    
```

---

Again, to prove Theorem 7 we need the following lemma that can be easily derived from the proof of Theorem 6.

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

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

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

*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  $e := \text{ct}.e$ , receives the resulting error  $e_{\text{res}}$  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  $\text{ct}_{\text{res}}$ . The oracle computes  $\text{ct}_{\text{res}}$  as  $\text{CKKS.Eval}(C_b, \text{ct}) + e_{\text{sm}}$ , where  $e_{\text{sm}}$  is sampled from  $\mathcal{N}_{\mathbb{Z}^n}(0, \sigma_s^2 T_{\text{max}}^2 I_n)$ . This means that the adversary sees  $e_{\text{res}}$  that is a sample of  $\mathcal{N}_{\mathbb{Z}^n}(e_b, \sigma_s^2 T_{\text{max}}^2 I_n)$ . 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  $e_{\text{res}}$ . This is a sample of  $\mathcal{N}_{\mathbb{Z}}(e_{b,i}, \sigma_s^2 T_{\text{max}}^2)$ . Obtaining that  $|e_{\text{res},i} - e_{0,i}| < |e_{\text{res},i} - e_{1,i}|$  is more likely when  $b = 0$  while if  $|e_{\text{res},i} - e_{0,i}| \geq |e_{\text{res},i} - e_{1,i}|$  it is at least more likely that  $b = 1$  rather than  $b = 0$ . 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

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

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

**Theorem 8.** *If the CKKS scheme with the modified evaluation  $\text{Eval}_{\mathcal{L}_d}^{\sigma_s}$  is  $\lambda$ -bit IND-CP-secure, then  $\sigma_s = \Omega(2^{\lambda/2}/(B^2 t_{\text{fresh}}^2))$ . This implies that one must add at least  $\lambda/2 - \log_2 \tilde{\Omega}(B^2 t_{\text{fresh}}^2)$  bits of additional Gaussian noise.*

*Proof.* By using the definition of bit-security, we know that  $\lambda \leq \log_2 O(\frac{T(A)}{\text{adv}^A}) \leq \log_2 O(\sigma_s^2 B^2 t_{\text{fresh}}^2)$ ; this immediately implies that  $\sigma_s \geq 2^{\lambda/2}/(B^2 t_{\text{fresh}}^2)$  and  $\lambda/2 - \log_2 \Omega(B^2 t_{\text{fresh}}^2) \leq \log_2 \sigma_s$ .

## 5 Threshold FHE and MPC

In Subsections 5.1 and 5.2, we give definitions for threshold and multikey homomorphic encryption over approximate arithmetic. In Subsection 5.3 we present a modification of the MK-CKKS multikey homomorphic encryption scheme that satisfies the indistinguishability security definition as given by Definition 18. In particular, we show that re-randomized MK-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 a superpolynomial Gaussian sample. Finally, in Subsection 5.4, we show that the parameters are tight. Namely, the Gaussian noise must be superpolynomial 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<sup>+</sup>18]. We have the same syntax and we have the same indistinguishability definition as [BGG<sup>+</sup>18], but we redefine the correctness definition for the case of approximate arithmetic. Regarding the indistinguishability, we discuss in Remark 1 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  $\mathcal{S} = (\mathbb{A}_1, \dots, \mathbb{A}_t) \subseteq \mathcal{P}(\mathcal{P}([n]))$  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 13 (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 THE = (KeyGen, Enc, Eval, PDec, Combine) with the following syntax.*

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

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

**Eval**( $\text{pk}, C, \text{ct}_1, \dots, \text{ct}_k$ )  $\rightarrow$   $c$ : *Given a public key  $\text{pk}$ , a circuit  $C \in \mathcal{L}_d$  and ciphertexts  $\text{ct}_1, \dots, \text{ct}_k$ , returns a ciphertext  $c$ .*

**PDec**( $\text{sk}_i, \text{ct}$ )  $\rightarrow$   $\mu$ : *Given a secret key  $\text{sk}_i$  and a ciphertext  $\text{ct}$ , returns a partial decryption  $\mu$ .*

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

**Definition 14 (Ind-secure THE).** *Let  $d \in \mathbb{N}$  and let  $\mathcal{L}_d$  be a class of circuits of multiplicative depth smaller or equal to  $d$ . Let THE = (KeyGen, Enc, Eval, PDec, 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  $\text{Exp}_b^{\text{IND-THE}}[\mathcal{A}]$ , where  $b \in \{0, 1\}$  is a bit and  $\mathcal{A}$  is an adversary. The experiment is defined as follows:*

$\text{Exp}_b^{\text{IND-THE}}[\mathcal{A}](1^\lambda)$  :

- $\mathbb{A} \leftarrow \mathcal{A}(\lambda, d, \mathbb{S}),$
- $(\text{sk}_1, \dots, \text{sk}_n, \text{pk}) \leftarrow \text{KeyGen}(\lambda, \mathbb{A}),$
- $S \leftarrow \mathcal{A}(\text{pk})$  s.t.  $S \notin \mathbb{A}$  and  $S$  is a maximal invalid set,
- $(m_1^{(0)}, \dots, m_k^{(0)}, m_1^{(1)}, \dots, m_k^{(1)}), \text{st} \leftarrow \mathcal{A}([\text{sk}_i]_{i \in S}),$
- $[\text{ct}_i \leftarrow \text{TFHE.Enc}(\text{pk}, m_i^{(b)})]_{i=1}^k,$
- $b' \leftarrow \mathcal{A}^{\text{Eval}(\text{pk}, \dots, \text{ct}_1, \dots, \text{ct}_k)}(\text{st}, \text{ct}_1, \dots, \text{ct}_n),$
- return**  $b'$

The **Eval**( $\text{pk}, \dots, \text{ct}_1, \dots, \text{ct}_k$ ) 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  $\text{ct}_{\text{res}} \leftarrow \text{Eval}(\text{pk}, C_i, \text{ct}_1, \dots, \text{ct}_k)$  and  $\mu_j \leftarrow \text{PDec}(\text{sk}_j, \text{ct}_{\text{res}})$  for all  $j \in [n]$ .

The scheme THE is said to be  $\lambda$ -bit IND-THE-secure if, for any adversary  $\mathcal{A}$ , we have that  $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\text{adv}^{\mathcal{A}}}$ , where  $\text{adv}^{\mathcal{A}}$  is defined as in Definition 8.

## 5.2 Multikey Homomorphic Encryption

There 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 threshold HE the ciphertext is already combined. Finally, (3) the decryption process in MKHE is a special case of threshold HE 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 `PDec`. Furthermore, to the best of our knowledge, all current realizations of these primitives use a flavor of noise flooding to realize `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 15 (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  $\text{MKHE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{PDec}, \text{Combine})$  with the following syntax.*

$\text{KeyGen}(\lambda, d) \rightarrow (\text{pk}, \text{sk})$ : *Given a security parameter  $\lambda$ , the maximal multiplicative depth of evaluable circuits  $d$ , the algorithm returns a public key  $\text{pk}$  and a secret key  $\text{sk}$ .*

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

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

$\text{PDec}(\text{sk}_i, \text{ct}) \rightarrow \mu$ : *Given a secret key  $\text{sk}_i$  and a ciphertext  $\text{ct}$ , returns a partial decryption  $\mu$ .*

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

**Definition 16 (Multikey Ciphertext Error).** *Let  $\text{MKHE} = (\text{KeyGen}, \text{Enc}, \text{Eval}, \text{PDec}, \text{Combine})$  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  $(\text{pk}_i, \text{sk}_i) \leftarrow \text{KeyGen}(\lambda)$  where  $i \in [k]$ , any ciphertexts  $\text{ct}$  in the image of `Eval` and message  $m \in \mathcal{M}$  the ciphertext error is defined as*

$$\text{Error}(\text{sk}_1, \dots, \text{sk}_n, \text{ct}, m) = \|\text{Combine}([\text{PDec}(\text{sk}_i, \text{ct})]_{i \in [k]}) - m\|.$$

Below we give our definition of approximate correctness for multikey homomorphic encryption. Definition 18 gives our definition for indistinguishability

security of multikey homomorphic encryption. Remind 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 17 (Approximate Correctness).** *Let us define MKHE = (KeyGen, Enc, Eval, PDec, 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  $\text{Estimate} : \bigsqcup_{n \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}_{\geq 0}^k \mapsto \mathbb{R}_{\geq 0}$  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  $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}(\lambda)$ , if  $\text{ct}_1, \dots, \text{ct}_k$  and  $m_1, \dots, m_k$  are such that  $\text{Error}_{\text{sk}_i}(\text{ct}_i, m_i) \leq t_i$ ,  $\text{ct} \leftarrow \text{Eval}(\text{pk}_1, \dots, \text{pk}_n, C, \text{ct}_1, \dots, \text{ct}_k)$ , then*

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

**Definition 18 (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 MKHE = (KeyGen, Enc, Eval, PDec, Combine) be a multikey homomorphic encryption scheme for a class circuits in  $\mathcal{L}_d$ . We define the experiment  $\text{Exp}_b^{\text{IND-MKHE}}[\mathcal{A}]$ , where  $b \in \{0, 1\}$  is a bit and  $\mathcal{A}$  is an adversary. The experiment is defined as follows:*

$\text{Exp}_b^{\text{IND-MKHE}}[\mathcal{A}](\lambda) :$

$$\begin{aligned} & [r'_i \xleftarrow{\$} \mathcal{U}]_{i \in [n]}, \\ & [(\text{sk}_i, \text{pk}_i) \leftarrow \text{KeyGen}(\lambda, d, r'_i)]_{i \in [n]}, \\ & i^* \leftarrow \mathcal{A}(\text{pk}_1, \dots, \text{pk}_n), \\ & [r_i \xleftarrow{\$} \mathcal{U}]_{i \in [n]}, \\ & (m_1^{(0)}, \dots, m_n^{(0)}, m_1^{(1)}, \dots, m_n^{(1)}), \text{st} \leftarrow \mathcal{A}([r_i, r'_i]_{i \in [n] \setminus \{i^*\}}), \\ & [\text{ct}_i \leftarrow \text{MKHE.Enc}(\text{pk}_i, m_i^{(b)}, r_i)]_{i \in [n]}, \\ & b' \leftarrow \mathcal{A}^{\text{Eval}(\{\text{pk}_i\}_{i \in [n]}, \cdot, \text{ct}_1, \dots, \text{ct}_n)}(\text{st}, [r'_i]_{i \in [n] \setminus \{i^*\}}, [r_i]_{i \in [n] \setminus \{i^*\}}, \text{ct}_{i^*}), \\ & \text{return } b' \end{aligned}$$

The  $\text{Eval}(\{\text{pk}_i\}_{i \in [n]}, \cdot, \text{ct}_1, \dots, \text{ct}_n)$  oracle takes as input a circuit  $C_i \in \mathcal{L}_d$  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  $\text{ct}_{\text{res}} \leftarrow \text{Eval}(\{\text{pk}_i\}_{i \in [n]}, C_i, \text{ct}_1, \dots, \text{ct}_n)$  and  $\mu_j \leftarrow \text{PDec}(\text{sk}_j, \text{ct}_{\text{res}})$  for all  $j \in [n]$ .

The scheme MKHE is said to be  $\lambda$ -bit IND-MKHE-secure if, for any adversary  $\mathcal{A}$ , we have that  $\lambda \leq \log_2 \frac{T(\mathcal{A})}{\text{adv}^{\mathcal{A}}}$ , where  $\text{adv}^{\mathcal{A}}$  is defined as in Definition 8.

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 a multikey homomorphic encryption scheme for two keys gives us a homomorphic encryption scheme with circuit privacy.

**Theorem 9.** *Let MKHE be a IND-MKHE-secure multikey homomorphic encryption scheme for  $n = 2$  parties. We can build a homomorphic encryption scheme HE that is IND-CP-secure.*

*Proof.* Let MKHE be a multikey homomorphic encryption for  $n = 2$  keys. We build the HE encryption as follows. The **KeyGen** and **Enc** algorithms are the same as in MKHE. We denote the keys output by the **KeyGen** algorithm as  $(\text{sk}_1, \text{pk}_1)$ . The evaluation algorithm **HE.Eval** on input  $\text{ct}_1 \leftarrow \text{MKHE.Enc}(\text{pk}_1, m)$  first samples  $(\text{pk}_2, \text{sk}_2) \leftarrow \text{KeyGen}(\lambda, d)$ , encrypts the circuit  $C$  as  $\text{ct}_2 \leftarrow \text{Enc}(\text{pk}_2, C)$ , and evaluates  $\text{ct} \leftarrow \text{MKHE.Eval}((\text{pk}_1, \text{pk}_2), U, \text{ct}_1, \text{ct}_2)$ , where  $U$  is a circuit that takes as input a message  $x$  and another circuit  $F$  and outputs  $F(x)$ . Finally, the eval algorithm outputs  $\text{ct}$  and  $\mu_2 \leftarrow \text{PDec}(\text{sk}_2, \text{ct})$ .

The decryption algorithm **HE.Dec** runs  $\text{ct} \leftarrow \text{MKHE.Eval}((\text{pk}_1, \text{pk}_2), U, \text{ct}_1, \text{ct}_2)$ ,  $\mu_1 \leftarrow \text{PDec}(\text{sk}_1, \text{ct})$ , and  $m' \leftarrow \text{Combine}(\{\mu_i\}_{i \in [n]}, \text{ct})$ . 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 IND-CP of HE to break IND-MKHE. The solver  $\mathcal{S}$  obtains  $\text{pk}_1, \text{pk}_2$  from the IND-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  $(m_1, \dots, m_k)$  and  $C_0$  and  $C_1$ , and sends  $(m_1, C_0)$  and  $(m_1, \dots, m_{k-1}, C_1)$  the MKHE challenger. Consequently,  $\mathcal{S}$  obtains  $\text{ct}_1$  and  $\text{ct}_2$ , and queries the **Eval** oracle on the  $U$  circuit and both ciphertexts. Denote the response of the oracle as  $\mu_2$ . The solver returns  $\mu_2$  and  $\text{ct} \leftarrow \text{Eval}(\text{pk}_1, \text{pk}_2, U, \text{ct}_1, \dots, \text{ct}_n)$  to  $\mathcal{A}$ . If  $\mathcal{A}$  returns a bit  $b'$  the solver outputs it as its solution to the IND-MKHE experiment.

Note that  $\mathcal{S}$  perfectly follows the IND-MKHE experiment. In particular, we set  $(m_1^{(b)}, m_2^{(b)}) = (m_1, \dots, m_k, C_b)$ . Note that we set  $m_1^{(b)} = (m_1, \dots, m_k)$  and  $m_2^{(b)} = C_b$ . From the requirement on  $C_0$  and  $C_1$  imposed by the IND-CP definition we have that  $C_0(m_1, \dots, m_k) = C_1(m_1, \dots, m_k)$ , and what follows  $U(C_0, m_1, \dots, m_k) = U(C_1, m_1, \dots, m_k)$  as required by the IND-MKHE experiment. To summarize, we have that the simulator  $\mathcal{S}$  has advantage  $\text{adv}_{\text{IND-CP}}[\mathcal{A}](\lambda)$  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 1 (On threshold homomorphic encryption and circuit privacy).* Remind 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 doesn't let itself use to build circuit privacy so easily. The reasons for this are that the common key generation algorithm in Definition 13 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 IND-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 IND-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**( $\lambda$ ): Given the security parameter  $\lambda$ , set  $n \in \mathbb{N}$  and  $Q \in \mathbb{N}$ , the ring  $\mathcal{R} := \mathcal{R}_Q^n$ , the key distribution  $\chi$  and the noise distribution  $\psi$ . Sample

$a \xleftarrow{\$} \mathcal{R}_Q^n$  uniformly. Return  $\text{pp} = (n, Q, \chi, \psi, a)$ .

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

**MK-CKKS.Enc**( $\text{pk}, m \in \mathcal{R}_Q$ ): Sample  $v \leftarrow \chi$  and  $e_0, e_1 \leftarrow \psi$ . Denoting  $\text{pk} = (b, a)$ , 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**( $\{\text{pk}_i\}_{i \in [k]}, C, \overline{\text{ct}}_1, \dots, \overline{\text{ct}}_k$ ): For given ciphertexts  $\overline{\text{ct}}_i \in \mathcal{R}^{k_i+1}$ , we denote  $k \geq \max_{i \in [k]} \{k_i\}$  the number of parties involved in at least one of the  $\overline{\text{ct}}_i$ . Rearrange the entries of each  $\overline{\text{ct}}_i$  and pad zeroes in empty entries to generate some ciphertexts  $\overline{\text{ct}}_i^*$  sharing the same secret key  $\overline{\text{sk}} = (1, \text{sk}_1, \dots, \text{sk}_k)$ . Then, the algorithm evaluates the arithmetic circuit  $C$  by means of addition and multiplication:

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

**CKKS.Mul**( $\{\text{pk}_i\}_{i \in [k]}, \overline{\text{ct}}_0, \overline{\text{ct}}_1 \in \mathcal{R}^{k+1}$ ): Compute  $\overline{\text{ct}} = \overline{\text{ct}}_1 \otimes \overline{\text{ct}}_2$  and return the ciphertext  $\overline{\text{ct}}' \leftarrow \text{Relin}(\overline{\text{ct}}, \{\text{pk}_i\}_{i \in [k]})$ . The `Relin` algorithm returns a ciphertext  $\overline{\text{ct}} \in \mathcal{R}^{k+1}$  encrypting  $m_0 m_1$  with an error that follows the noise growth law of Lemma 5.

**MK-CKKS.PDec**( $\text{sk}, \overline{\text{ct}} \in \mathcal{R}^{k+1}$ ): Call  $\overline{\text{ct}}.a_i$  the component of  $\overline{\text{ct}}$  associated to the secret key  $\text{sk}$ . Return  $\mu = \text{sk} \cdot \overline{\text{ct}}.a_i$ .<sup>1</sup>

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

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

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

To simplify the notation, from now on, we are going to refer to the entries of a ciphertext  $\text{ct} \in \mathcal{R}^{k+1}$  as  $(\text{ct}.b, \text{ct}.a_1, \dots, \text{ct}.a_k)$ . Also, when writing  $\text{ct}.a$ , we will be referring to  $(\text{ct}.a_1, \dots, \text{ct}.a_k)$ . 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  $\text{pk}$  associated to  $\text{ct}$  and then to post-process the component  $\text{ct}.b$  using an appropriate differential privacy mechanism  $M_T$ .

---

**Algorithm 3:** The modified evaluation MK-CKKS.`Eval'`

---

**Data:** A set of public keys  $\{\text{pk}_i\}_{i \in [k]}$ , circuit  $C \in \mathcal{L}$ , a vector of ciphertexts  $\text{ct}_1 \in \mathcal{R}^{k+1}, \dots, \text{ct}_N \in \mathcal{R}^{k+1}$ .

```

begin
   $\text{ct}_{\text{res}} \leftarrow \text{Eval}(\{\text{pk}_i\}_{i \in [k]}, C, \text{ct}_1, \dots, \text{ct}_k)$ ;
  For  $i = 1$  to  $k$ :  $\text{ct}_{\text{res}} \leftarrow \text{ct}_{\text{res}} + \text{Enc}(\text{pk}_i, 0)$ ;
   $T \leftarrow \text{ct}_{\text{res}}.t + t_{\text{fresh}}$ ;
   $\text{ct}_{\text{res}}.b \leftarrow M_T(\text{ct}_{\text{res}}.b)$ ;
  return  $\text{ct}_{\text{res}}$ ;

```

---



---

**Algorithm 4:** The modified partial decryption MK-CKKS.`PDec'`

---

**Data:** A secret key  $\text{sk}$ , a ciphertext  $\text{ct} \in \mathcal{R}^{k+1}$ .

```

begin
   $\mu \leftarrow M_{\text{ct}.t}(\text{PDec}(\text{sk}, \text{ct}))$ ;
  return  $\mu$ ;

```

---

**Theorem 10.** *Let  $\text{MK-CKKS} = (\text{Setup}, \text{KeyGen}, \text{Enc}, \text{Eval}, \text{PDec}, \text{Combine})$  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  $\rho$ -KLDP mechanism on  $\mathcal{R}$  where  $\rho \leq 2^{-\lambda-8}/q$ . If  $\text{MK-CKKS.Enc}$  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  $\lambda$ -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  $\text{sk}_{i^*}$ . 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 a little bit harder.

We start by describing the two indistinguishability games.

1.  $\mathcal{G}_0$ : 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.
2.  $\mathcal{G}_1$ : 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  $\text{Eval}'$ . The oracle  $\text{Eval}'(\{\text{pk}_i\}_{i=1}^n, \cdot, \text{ct}_1, \dots, \text{ct}_n)$  takes as input a circuit  $C_i \in \mathcal{L}_d$  such that  $C_i(m_1^{(0)}, \dots, m_n^{(0)}) = C_i(m_1^{(1)}, \dots, m_n^{(1)})$ , and behaves in the following way. When writing  $\text{Enc}_n(\text{pk}, m)$  we denote a noiseless encryption of  $m$ .

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

In  $\mathcal{G}_0$ , the ciphertext  $\text{ct}_{\text{res}}$  and the decryption shares  $\mu_i$  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  $\text{ct}_{\text{res}}$  and with a differential privacy mechanism on both. In  $\mathcal{G}_1$ , the ciphertext  $\text{ct}_{\text{res}}$  and the decryption shares  $\mu_i$  are simulated, and they do not depend from the input ciphertexts, from  $b$  or from the secret key of the non-corrupted party  $i^*$ .  $\text{ct}_{\text{res}}$  is a fresh, random encryption of  $m_{\text{res}}$ , and the share  $\mu_{i^*}$  is obtained without using  $\text{sk}_{i^*}$ .

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

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

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

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

divergence after post-processing.

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

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  $\text{ct}_0.b = \text{ct}_1.b = b$  and  $\text{ct}_0.a = \text{ct}_1.a = a$  are chosen, we compute the norm of the difference between  $\mu_{i^*,0}$  and  $\mu_{i^*,1}$ , which are the decryption shares before applying the differential privacy mechanism.

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

This implies, thanks to Definition 9, that

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

From this point forward, we often use the notation  $D_a(\mathcal{X}||\mathcal{Y})$  when referring to  $D(\mathcal{X}|\text{ct}.a = a)||\mathcal{Y}|\text{ct}.a = a)$ . We now use Lemma 1 to obtain the following inequality.

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

It is easy to show that  $\text{ct}_0.a_i$  are uniform random in  $\mathcal{R}$  for each  $i \in [k]$  because we re-randomized each entry by adding  $\text{Enc}(\text{pk}_i, 0)$  to  $\text{ct}_0$ . This is also true for  $\text{ct}_1.a_i$  for each  $i \neq i^*$ . We can also say that  $\text{ct}_1.a_{i^*}$  is uniform random in  $\mathcal{R}$  because it is obtained as a fresh encryption of 0. This implies that the KL divergence  $D(\text{ct}_0.a||\text{ct}_1.a) = 0$ . We can now apply Lemma 1 and obtain the following inequality.

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

We have already shown that  $\rho$  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.

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

Then, we use Lemma 3 with  $\mathcal{X}_\theta$  defined as a query to the oracle  $\text{Eval}$  of  $\mathcal{G}_0$  and  $\mathcal{Y}_\theta$  as a query to the oracle  $\text{Eval}'$ .

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

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  $\text{sk}_{i^*}$  and then receives a fresh encryption of zero under  $\text{sk}_{i^*}$  for a polynomial number of times  $q$ . This implies that, if  $\text{MK-CKKS.Enc}$  is  $(\lambda+8)$ -bit secure, 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 2 and obtain that  $\mathcal{G}_0$  is  $\lambda$ -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 `Combine` algorithm, a Gaussian noise of standard deviation  $2^{7/2}\sqrt{q}2^\lambda(\text{ct}.t + kt_{\text{fresh}})$  and  $(k - 1)$  Gaussian noises of standard deviation  $2^{7/2}\sqrt{q}2^\lambda\text{ct}.t$  are added to each coordinate. The additional bits of precision lost are approximately  $\lambda/2 + \log_2 \sqrt{q} + 7/2 + \log_2 k + \log_2 t_{\text{fresh}}$ .

#### 5.4 Tightness of the Differential Privacy Parameters

By Theorem 10, it is possible to achieve  $\lambda$  bits of IND-MKHE-security by post-processing the outputs from `Eval` and `PDec` with a differentially private algorithm. Concretely we choose the Gaussian mechanism with Gaussian noise of variance  $\sigma_{\max} \leftarrow \frac{\text{ct}.t^2}{2\rho}$ , where  $\rho \leq 2^{-\lambda-8}/q$  is the privacy bound for  $\rho$ -KL differential privacy (Definition 9). We show that, using an appreciably smaller variance  $\sigma_s \ll \sigma_{\max}$ , 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 superpolynomial in the security parameter.

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

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

*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.

**Algorithm 5:** Adversary  $\mathcal{A}(\lambda)$ .**Data:** A security parameter  $\lambda$ . The adversary has oracle access to  $\text{Eval}_{\sigma_s}$ .

---

```

begin
  pp  $\leftarrow$  Setup( $\lambda, d$ );
   $[r'_i \xleftarrow{\$} \mathcal{U}]$ ;
   $[(\text{sk}_i, \text{pk}_i) \leftarrow \text{KeyGen}(\text{pp}, r'_i)]_{i \in [2]}$ ;
   $i^* \leftarrow 1$ ;
   $[r_i \xleftarrow{\$} \mathcal{U}]_{i \in [2]}$ ;
   $(m_1^{(0)}, m_2^{(0)}), (m_1^{(1)}, m_2^{(1)}) \leftarrow (0, B), (B, B)$ ;
   $C \leftarrow x_1 \cdot x_2 - B \cdot x_1$ ;
  ct  $\leftarrow$  Enc( $\text{pk}_1, m_1^{(b)}, r_1$ );
   $\tilde{\text{ct}} \leftarrow$  Enc( $\text{pk}_2, m_2^{(b)}, r_2$ );
   $\tilde{e} \leftarrow$  Dec( $\text{sk}_2, \text{ct}_2$ )  $- B$ ;
   $\text{ct}_{\text{res}}, \mu_1, \mu_2 \leftarrow \mathcal{O}^{\text{Eval}_{\sigma_s}}(\{\text{pk}_i\}_{i \in [2]}, C, \text{ct}_1, \text{ct}_2)$ ;
   $e_{\text{res}} \leftarrow$  Combine( $\mu_1, \text{PDec}(\text{sk}_2, \text{ct}_{\text{res}}), \text{ct}_{\text{res}}$ );
  Choose  $I \in \{0, n-1\}$  such that  $|\tilde{e}_I|$  is maximal;
  If  $|e_{\text{res}, I} - B\tilde{e}_I| \geq |e_{\text{res}, I}|$  then return 0. Otherwise output 1;

```

---

The adversary knows the exact error  $\tilde{e} := \tilde{\text{ct}}.e$  and obtains the resulting error  $e_{\text{res}}$  after post-processing. We denote as  $e \leftarrow \mathcal{N}_{\mathbb{Z}^n}(0, \sigma^2 I_n)$  the exact error of  $\text{ct}$ . Recalling the error growth rule of MK-CKKS from Lemma 5, we can estimate the two possible outputs for  $b \in \{0, 1\}$ . The resulting error after computing  $x \cdot y$  is equal to  $e\tilde{e} + m_b\tilde{e} + Be + e_{\text{mult}}$ . 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  $\text{ct}_{\text{res}}$  is  $e\tilde{e} + m_b\tilde{e} + e_{\text{mult}} + e_{\text{sm}}^{(1)}$  where the  $e_{\text{sm}}^{(1)}$  is the post-processing noise of  $\text{Eval}_{\sigma_s}$ . When we compute the decryption of  $\text{ct}_{\text{res}}$  using the Combine algorithm, we obtain that the result is

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

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

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

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  $e_{\text{res}}$  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  $e_{\text{mult}}$  and  $e_{\text{lin}}$  are significantly smaller than the other terms (Heuristic 1) and that we can omit them; this approximation allows us to express  $e_{\text{res},I}$  as a sample from the following Gaussian distribution.

$$\mathcal{N}_{\mathbb{Z}}(m_b \tilde{e}_I, \sum_{i=0}^{n-1} \tilde{e}_i^2 \sigma^2 + 2\sigma_s^2 \text{ct}.t^2).$$

Obtaining that  $|e_{\text{res},I} - B\tilde{e}_I| < |e_{\text{res},I}|$  is more likely when  $b = 1$  while, if  $|e_{\text{res},I} - B\tilde{e}_I| \geq |e_{\text{res},I}|$ , 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  $V := \sqrt{\|\tilde{e}\|_2^2 \sigma^2 + 2\sigma_s^2 \text{ct}.t^2}$  and use Lemma 3 to obtain the following lower bound.

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

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

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

**Theorem 12.** *If the scheme MK-CKKS with the modified evaluation  $\text{Eval}_{\sigma_s}$  and the modified partial decryption  $\text{PDec}_{\sigma_s}$  is  $\lambda$ -bit IND-MKHE-secure, then  $\sigma_s = \Omega(2^{\lambda/2}/\sigma n^{3/2})$ , i.e. one must add at least  $\lambda/2 - \tilde{\Omega}(\sigma n^{3/2})$  bits of additional Gaussian noise.*

*Proof.* By using the definition of bit-security, we know that  $\lambda \leq \log_2 O\left(\frac{T(A)}{\text{adv}^A}\right) \leq \log_2 O(\sigma_s^2 \sigma^2 n^3)$ . 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  $\lambda$ -bit IND-CP-security by post-processing the ciphertext with

$\lambda/2 + \tilde{O}(1)$  bits of noise. Additionally, we modified the MK-CKKS scheme (Theorem 10) to achieve  $\lambda$ -bit IND-MKHE-security. We did this by post-processing the ciphertext and the decryption shares with  $\lambda/2 + \tilde{O}(1)$  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.

*Acknowledgments.* This work has been partially funded/supported by the German Ministry for Education and Research through funding for the project CISPASTanford Center for Cybersecurity (Funding number: 16KIS0927).

## References

- ABFK16. Carlos Aguilar Melchor, Joris Barrier, Laurent Fousse, and Marc-Olivier Killijian. XPIR: Private information retrieval for everyone. *PoPETs*, 2016(2):155–174, April 2016.
- 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, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 453–472. Springer, Heidelberg, March 2019.
- 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. IEEE Computer Society Press, May 2018.
- 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, Part II*, volume 13748 of *LNCS*, pages 70–99. Springer, Heidelberg, November 2022.
- 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, Part I*, volume 12550 of *LNCS*, pages 28–57. Springer, Heidelberg, November 2020.
- ALP<sup>+</sup>21. Asra Ali, Tancrede Lepoint, Sarvar Patel, Mariana Raykova, Phillipp Schoppmann, Karn Seth, and Kevin Yeo. Communication-computation trade-offs in PIR. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 1811–1828. USENIX Association, August 2021.
- BDGM20. Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate iO from homomorphic encryption schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 79–109. Springer, Heidelberg, May 2020.
- BDPMW16. Florian Bourse, Rafaël Del Pino, Michele Minelli, and Hoeteck Wee. Fhe circuit privacy almost for free. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 62–89, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

- BGG<sup>+</sup>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. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 565–596, Cham, 2018. Springer International Publishing.
- BGGJ18. Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev. CHIMERA: Combining ring-LWE-based fully homomorphic encryption schemes. Cryptology ePrint Archive, Report 2018/758, 2018. <https://eprint.iacr.org/2018/758>.
- 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, 2020.
- BLR<sup>+</sup>18. Shi Bai, Tancrede 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.
- BP16. Zvika Brakerski and Renen Perlman. Lattice-based fully dynamic multikey FHE with short ciphertexts. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 190–213. Springer, Heidelberg, August 2016.
- BS23. Katharina Boudgoust and Peter Scholl. Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. Cryptology ePrint Archive, Report 2023/016, 2023. <https://eprint.iacr.org/2023/016>.
- CCS19. Hao Chen, Ilaria Chillotti, and Yongsoo Song. Multi-key homomorphic encryption from TFHE. In Steven D. Galbraith and Shihō Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 446–472. Springer, Heidelberg, December 2019.
- CDKS19. Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. Efficient multikey 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*, pages 395–412. ACM Press, November 2019.
- CdWM<sup>+</sup>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. <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, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 3–33. Springer, Heidelberg, May / June 2022.
- 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, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 409–437. Springer, Heidelberg, December 2017.
- 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*, pages 1243–1255. ACM Press, October / November 2017.

- 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, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 630–656. Springer, Heidelberg, August 2015.
- CO17. Wutichai Chongchitmate and Rafail Ostrovsky. Circuit-private multi-key FHE. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 241–270. Springer, Heidelberg, March 2017.
- CSS<sup>+</sup>22. Siddhartha Chowdhury, Sayani Sinha, Animesh Singh, Shubham Mishra, Chandan Chaudhary, Sikhar Patranabis, Pratyay Mukherjee, Ayantika Chatterjee, and Debdeep Mukhopadhyay. Efficient threshold FHE with application to real-time systems. Cryptology ePrint Archive, Report 2022/1625, 2022. <https://eprint.iacr.org/2022/1625>.
- 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, Part II*, volume 10678 of *LNCS*, pages 597–627. Springer, Heidelberg, November 2017.
- DGBL<sup>+</sup>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.
- DS16. Léo Ducas and Damien Stehlé. Sanitization of FHE ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 294–310. Springer, Heidelberg, May 2016.
- DWF22. Xiaokang Dai, Wenyuan Wu, and Yong Feng. Summation rather than concatenation: a more efficient MKFHE scheme in the plain model. Cryptology ePrint Archive, Report 2022/055, 2022. <https://eprint.iacr.org/2022/055>.
- Gen09a. Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009.
- Gen09b. Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.
- GH19. Craig Gentry and Shai Halevi. Compressible FHE with applications to PIR. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 438–464. Springer, Heidelberg, December 2019.
- GHS12. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 850–867. Springer, Heidelberg, August 2012.
- GHV10. Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. i-Hop homomorphic encryption and rerandomizable Yao circuits. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 155–172. Springer, Heidelberg, August 2010.
- GP21. Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 736–749. ACM Press, June 2021.

- 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, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.
- 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.
- HHC<sup>+</sup>22. 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. *Cryptology ePrint Archive*, Report 2022/949, 2022. <https://eprint.iacr.org/2022/949>.
- IP07. Yuval Ishai and Anat Paskin. Evaluating branching programs on encrypted data. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 575–594. Springer, Heidelberg, February 2007.
- 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*, pages 1209–1222. ACM Press, October 2018.
- 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*, pages 1651–1669. USENIX Association, August 2018.
- KKL<sup>+</sup>22. Taechan Kim, Hyesun Kwak, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Asymptotically faster multi-key homomorphic encryption from homomorphic gadget decomposition. *Cryptology ePrint Archive*, Report 2022/347, 2022. <https://eprint.iacr.org/2022/347>.
- Klu22. Kamil Kluczniak. NTRU- $\nu$ -um: Secure fully homomorphic encryption from NTRU with small modulus. *Cryptology ePrint Archive*, Report 2022/089, 2022. <https://eprint.iacr.org/2022/089>.
- 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.
- KSK<sup>+</sup>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(4):23–31, 2018.
- KSK<sup>+</sup>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(7):1–12, 2020.
- 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*, pages 619–631. ACM Press, October / November 2017.
- LM21. Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 648–677. Springer, Heidelberg, October 2021.

- LMSS22. Baiyu Li, Daniele Micciancio, Mark Schultz, and Jessica Sorrell. Securing approximate homomorphic encryption using differential privacy. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 560–589. Springer, Heidelberg, August 2022.
- 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 ACM STOC*, pages 1219–1234. ACM Press, May 2012.
- 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.
- MW16. Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 735–763. Springer, Heidelberg, May 2016.
- MW18. Daniele Micciancio and Michael Walter. On the bit security of cryptographic primitives. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 3–28. Springer, Heidelberg, April / May 2018.
- 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. IEEE Computer Society Press, May 2022.
- PW14. Yury Polyanskiy and Yihong Wu. *Lecture notes on information theory*. 2014.
- QWW18. Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. Cryptology ePrint Archive, Report 2018/409, 2018. <https://eprint.iacr.org/2018/409>.
- RAD78. RL Rivest, L Adleman, and ML Dertouzos. On data banks and privacy homomorphisms. foundations of secure computation (1978), 169–180. *Search in*, 1978.
- RSC<sup>+</sup>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*, pages 1501–1518. USENIX Association, August 2019.