Securing Approximate Homomorphic Encryption Using Differential Privacy

Baiyu Li, Daniele Micciancio, Mark Schultz(✉), and Jessica Sorrell

University of California, San Diego, USA
{baiyu,daniele,mdschultz,jlsorrel}@eng.ucsd.edu

Abstract. Recent work of Li and Micciancio (Eurocrypt 2021) has shown that the traditional formulation of *indistinguishability under chosen plaintext attack* (IND-CPA) is not adequate to capture the security of *approximate* homomorphic encryption against passive adversaries, and identified a stronger $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security definition (IND-CPA *with decryption oracles*) as the appropriate security target for approximate encryption schemes. We show how to transform any approximate homomorphic encryption scheme achieving the weak IND-CPA security definition, into one which is provably $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ secure, offering strong guarantees against realistic passive attacks. The method works by postprocessing the output of the decryption function with a mechanism satisfying an appropriate notion of *differential privacy (DP)*, adding an amount of noise tailored to the worst-case error growth of the homomorphic computation.

We apply these results to the approximate homomorphic encryption scheme of Cheon, Kim, Kim, and Song (CKKS, Asiacrypt 2017), proving that adding Gaussian noise to the output of CKKS decryption suffices to achieve $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security. We precisely quantify how much Gaussian noise must be added by proving nearly matching upper and lower bounds, showing that one cannot hope to significantly reduce the amount of noise added in this post-processing step. As an additional contribution, we present and use a finer grained definition of bit security that distinguishes between a computational security parameter (c) and a statistical one (s). Based on our upper and lower bounds, we propose parameters for the counter-measures recently adopted by open-source libraries implementing CKKS.

Lastly, we investigate the plausible claim that smaller DP noise parameters might suffice to achieve $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security for schemes supporting more accurate (dynamic, key dependent) estimates of ciphertext noise during decryption. Perhaps surprisingly, we show that this claim is false, and that DP mechanisms with noise parameters tailored to the error present in a given ciphertext, rather than worst-case error, are vulnerable to $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ attacks.

Research supported by Global Research Cluster program of Samsung Advanced Institute of Technology and NSF Award 1936703.

Y. Dodis and T. Shrimpton (Eds.): CRYPTO 2022, LNCS 13507, pp. 560–589, 2022.
https://doi.org/10.1007/978-3-031-15802-5_20

1 Introduction

Fully homomorphic encryption (FHE) on *approximate* numbers, proposed by Cheon, Kim, Kim and Song in [8], has attracted much attention in the past few years as a method to improve the efficiency of computing on encrypted data in a wide range of applications (like privacy preserving machine learning) where approximate results are acceptable [5–7,9,10,14,25]. The CKKS scheme [8], just like most other (homomorphic) encryption schemes based on lattices, can be proved to satisfy the well established security notion of *indistinguishability under chosen plaintext attack* (IND-CPA) [13] under widely accepted complexity assumptions, like the average-case hardness of the *Learning With Errors (LWE)* problem or the worst-case complexity of computational problems on (algebraic) point lattices [20,26,27,31].

Recently Li and Micciancio [19] have shown that the traditional formulation of IND-CPA security is inadequate to capture security of approximate encryption against passive attacks, and demonstrated that the CKKS scheme is susceptible to a very efficient total key recovery attack, mounted by a passive adversary. The problem highlighted in [19] is not with the IND-CPA security definition per se, which remains a good and well accepted definition for exact FHE schemes, but with the specifics of approximate decryption, which may inadvertently leak information about the secret key even when used by honest parties. The work [19] also proposes a new, enhanced formulation of IND-CPA security (called $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$, or IND-CPA *with decryption oracles*), which properly captures the capabilities of a passive attacker against an *approximate* FHE scheme, and is equivalent to the standard notion of IND-CPA security for encryption schemes with exact decryption. The work [19] also suggested some practical countermeasures to avoid their attack, and all major open source libraries implementing CKKS (e.g., [15,18,24,32]) included similar countermeasures shortly after the results in [19] were made public. However, neither [19] nor any of these libraries present a solution that provably achieves the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security definition proposed in [19], leaving it as an open problem.

1.1 Our Results and Techniques

In this work we show how to achieve $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security in a provable way. More specifically, we present a general technique to transform any approximate FHE scheme satisfying the (weak) IND-CPA security notion into one achieving the strong $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security definition proposed in [19]. We then demonstrate how to apply the technique to the specific case of the CKKS scheme, which is the most prominent example of approximate homomorphic encryption.

Our technique works by combining a given (approximate) FHE scheme with another fundamental tool from the cryptographers' toolbox: differential privacy. The construction is very simple and intuitive: given an approximate FHE scheme (like CKKS), we modify the decryption function by post-processing its output (the decrypted message) with a properly chosen differentially private mechanism. Using differential privacy to limit the key leakage of approximate decryption is a fairly natural idea, and it is essentially the intuition behind the practical countermeasures proposed in [19] and implemented by the libraries. But formally analyzing the method and provably achieving $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security raises a number of technical challenges:

- The Hamming metric, commonly used to define and analyze differentially private mechanisms, is not well suited to the setting of (lattice based) homomorphic encryption.
- Similarly, the Laplace noise commonly used and studied in the standard setting of differential privacy is not a good match for our target application, as it is both associated with the wrong norm (ℓ_1, rather than ℓ_2 or ℓ_∞), and has heavier tails than, e.g., the Gaussian distribution, and so will give worse bounds on the error introduced by post-processing.
- Formally proving the security of our construction requires a careful definition of what it means for an FHE scheme to be *approximate*. Previous works [8,19] simply defined approximate FHE as an encryption scheme which *does not* satisfy the correctness requirement

$$\mathsf{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1), \ldots, \mathsf{Enc}(m_k))) = f(m_1, \ldots, m_k) \tag{1}$$

  without imposing any specific limitation on how a scheme may deviate from it.
- Perturbing the output of the decryption function with a differentially private mechanism comes at the cost of lowering the output quality, making the result of the (already approximate) decryption function even less accurate, highlighting the necessity of carefully tuning the amount of noise added.
- The minimal security level considered acceptable by applications in practice typically depends on whether the cryptographic primitive is statistically secure (against computationally unbounded adversaries) or computationally secure (in which case a higher security margin is advisable to anticipate possible algorithmic or implementation improvements in the attacks.) Our application of statistical security tools (differential privacy) to encryption seems to require the instantiation of statistical security with the high security parameters of a computational encryption scheme.

In order to address the above obstacles, we

- provide a general definition of differential privacy, parameterized by an arbitrary norm, and then instantiate it with the Euclidean norm for the case of lattice-based encryption;
- employ a differentially private mechanism (for the Euclidean norm) based on Gaussian noise, which blends well with the probability distributions used in lattice cryptography;
- give formal definitions of *approximate* FHE, which provide precise guarantees on the output quality of the (approximate) decryption function. In fact, we identify two possible definitions, based on what we call *static* and *dynamic* noise estimates, and show that they result in quite different security properties (more on this below);

- use KL-divergence and other probabilistic tools to carefully calibrate the mechanism noise to the output quality, showing that $\Theta(\kappa)$ bits of noise are required to formally achieve κ-bit $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security;
- present and use a finer grained definition of bit-security that distinguishes between a computational security parameter c and a statistical one s, which can be set to a lower value than c (more on this below).

We first elaborate on our definition of *approximate* FHE. Previous works [8,19] did not include a precise definition of what it means for an encryption scheme (or decryption function) to be approximate, because the quality of the approximation (and more generally, the definition of the decryption function itself) does not impact the IND-CPA security of a scheme. This is contrasted with our work, where bounding the approximation quality of the decryption function plays a critical role in our analysis. Generally speaking, an approximate FHE scheme provides a guarantee (upper bound) on how much the output of the decryption function $\mathsf{Dec}(\mathsf{Eval}(f, \mathsf{Enc}(m_1), \ldots, \mathsf{Enc}(m_k)))$ may deviate from the output of the computation $f(m_1, \ldots, m_k)$. We distinguish two types of approximate FHE:

- Approximate FHE with *static* noise estimates, where this bound can be publicly computed as a function of the homomorphic computation f performed on the input ciphertexts. This is, for example, the type of noise estimates used in the HElib library [15].
- Approximate FHE with *dynamic* noise estimates, where this bound is computed by the decryption function Dec using also the input ciphertext and the secret key. An ingenious method for dynamic noise estimation has been proposed by the PALISADE library [24].

Most of our results, like our general framework based on differential privacy and a provably $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ secure variant of the CKKS approximate FHE scheme, are in the setting of static noise estimates. In this setting, we are able to establish the security of our generic construction (Theorem 2), and provide precise security guarantees for the modified approximate FHE scheme, showing that if the original scheme is κ-bit IND-CPA secure, then combining it with an appropriate differentially private mechanism achieves $\kappa - 8$ bits of security against the stronger $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security definition, losing only 8 bits of security (Theorem 2). The amount of noise required to achieve this result is quantified by the notion of ρ-KLDP (Kullback-Leibler Differential Privacy), for a sufficiently small value of ρ. Our analysis is nearly tight for the CKKS scheme, in the sense that if one uses a substantially smaller amount of noise, we are able to exhibit an attack that breaks $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security (Theorem 4).

When setting the parameters of a cryptosystem (or other computational cryptographic primitive), it is common to use a very conservative security level to anticipate reductions in both the hardware and operational cost of mounting an attack. A common level of security considered adequate for most applications is $c = 128$ bits of security. When applying a statistical technique (like differential privacy) to a computational primitive, this seems to require instantiating the statistical technique with the same (high) level of bit security. We propose a finer grained definition of bit-security (Definition 19) parameterized by both a *computational* parameter c and *statistical* parameter s. Technically, we say that a primitive achieves (c, s)-security if for any adversary A, either A has statistical advantage bounded by 2^{-s} (regardless of A's running time or computational assumptions), or the running time of the attack is at least 2^c times larger than the advantage achieved. Intuitively, this definition captures the notion that if c bits of security are acceptable for a computational cryptographic primitive, and s bits of security are enough for an unconditionally secure cryptographic primitive (independent of any computational assumption), then (c, s)-security is also adequate.

Still, (c, s)-security is technically easier to achieve than both c-bit computational security, and s-bit statistical security, and allows us to decrease the cost of our countermeasure (Theorem 2) by lowering the required amount of DP noise by $(c - s)/2$ bits. The standard notion of bit-security corresponds to setting $s = c$, which gives no improvement. But for typical parameter settings (e.g., $c = 128$ and $s = 64$), the refined definition allows to reduce the required amount of noise from ≈ 75 bits to ≈ 45, a substantial saving of ≈ 30 bits. As even more conservative choices, such as $s = 80$ or $s = 100$, yield savings of ≈ 24 or ≈ 14 bits of noise, we expect this refined notion of security to be concretely useful when securing CKKS against the attacks of [19].

All this is for static noise estimates. Dynamic estimates are interesting because they can provide stronger (probabilistic) guarantees on the output quality of the decryption function. Interestingly, we show that the same intuitive idea of combining approximate FHE with differential privacy, while calibrating the DP noise via dynamic error estimates, does not result in a secure scheme. In particular, we describe attacks to the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security of CKKS using dynamic noise estimates (Theorem 6), and complete key recovery attacks for other (artificially constructed) $\mathsf{IND\text{-}CPA}$-secure FHE schemes (Theorem 7).

1.2 Paper Outline

The rest of the paper is organized as follows. In Sect. 2 we present background definitions and results from cryptography, fully homomorphic encryption, and probability theory. In Sect. 3 we present our general framework to secure approximate FHE using differential privacy, for the setting of *static* error estimation. In Sect. 4 we apply the framework to the CKKS scheme, and develop our relaxed notion of bit security. In Sect. 5 we present our (negative) results for approximate FHE with *dynamic* error estimation. Section 6 concludes with a summary of our results and open problems.

2 Preliminaries

We recall some notions and known results.

2.1 Probability

We abbreviate a list of random variables $(\mathcal{X}_1, \ldots, \mathcal{X}_n)$ as $(\mathcal{X}_i)_i$. For such a list, we write $\mathcal{X}_{<i}$ to denote the prefix $(\mathcal{X}_1, \ldots, \mathcal{X}_{i-1})$. A probability ensemble $(\mathcal{P}_\theta)_\theta$ is a family of probability distributions parameterized by a variable θ, which may be a string or a vector.

Throughout this work we will use *divergences* to measure how far probability distributions are from eachother.

Definition 1. *A $\mathbb{R}$-valued function $\delta(\cdot||\cdot)$ on pairs of discrete distributions is called a* divergence *if it satisfies*

- Non-negativity*: For any discrete distributions $\mathcal{P}, \mathcal{Q}$, $\delta(\mathcal{P}||\mathcal{Q}) \geq 0$.*
- Identity of Discernibles*: If $\delta(\mathcal{P}||\mathcal{Q}) = 0$, then $\mathcal{P} = \mathcal{Q}$.*

Notably, divergences need not be symmetric, nor satisfy triangle inequality, although specific divergences will typically satisfy some additional properties than solely the above two.

Definition 2 (Statistical Distance). *Let $\mathcal{P}, \mathcal{Q}$ be discrete distributions with common support X. The* Statistical Distance *(or* Total Variation Distance*) between $\mathcal{P}$ and $\mathcal{Q}$ is $\Delta(\mathcal{P}, \mathcal{Q}) = \frac{1}{2}\sum_{x\in X} |\mathcal{P}(x) - \mathcal{Q}(x)|$.*

The statistical distance is a divergence that is symmetric and satisfies triangle inequality, *i.e.* is a metric.

Definition 3 (KL Divergence). *Let $\mathcal{P}, \mathcal{Q}$ be discrete distributions with common support X. The* Kullback-Leibler Divergence *between $\mathcal{P}$ and $\mathcal{Q}$ is $D(\mathcal{P}||\mathcal{Q}) := \sum_{x\in\mathcal{X}} \Pr[\mathcal{P} = x] \ln\left(\frac{\Pr[\mathcal{P}=x]}{\Pr[\mathcal{Q}=x]}\right)$.*

Lemma 1 (Properties of the KL Divergence, Theorem 2.2 of [29]). *The KL divergence satisfies*

1. Sub-Additivity for Joint Distributions*: If $(\mathcal{X}_0, \mathcal{X}_1)$ and $(\mathcal{Y}_0, \mathcal{Y}_1)$ are pairs of (possibly dependent) random variables, then*

$$\begin{aligned} D((\mathcal{X}_0, \mathcal{X}_1)||(\mathcal{Y}_0, \mathcal{Y}_1)) &\leq \mathbb{E}_{x\sim\mathcal{X}_0}[D((\mathcal{X}_1 \mid x)||(\mathcal{Y}_1 \mid x))] + D(\mathcal{X}_0||\mathcal{Y}_0) \\ &\leq \max_x D((\mathcal{X}_1 \mid x)||(\mathcal{Y}_1 \mid x)) + D(\mathcal{X}_0||\mathcal{X}_1), \end{aligned}$$

2. Data Processing Inequality*: For any (potentially randomized) function f, for any two distributions $\mathcal{P}, \mathcal{Q}$, $D(f(\mathcal{P})||f(\mathcal{Q})) \leq D(\mathcal{P}||\mathcal{Q})$, and*
3. Pinsker's Inequality*: $\Delta(\mathcal{P}, \mathcal{Q}) \leq \sqrt{D(\mathcal{P}||\mathcal{Q})/2}$.*

We introduce the following notation to more compactly bound the divergence between vectors of random variables.

Definition 4. *Let $\mathcal{X} = (\mathcal{X}_i)_{i=1}^n, \mathcal{Y} = (\mathcal{Y}_i)_{i=1}^n$ be two lists of discrete random variables over the support $\prod_{i=1}^n X_i \subseteq \mathbb{R}^n$, and δ any divergence. We define the* vector divergence *$\hat{\delta}(\mathcal{X}||\mathcal{Y})$ to be the non-negative real vector $(v_1, \ldots, v_n) \in \mathbb{R}^n_{\geq 0}$ with coordinates $v_i = \max_a \delta([\mathcal{X}_i \mid \mathcal{X}_{<i} = a]||[\mathcal{Y}_i \mid \mathcal{Y}_{<i} = a])$.*

In this notation, sub-additivity of the KL divergence (for example) can be written as $D(\mathcal{X}||\mathcal{Y}) \leq \|\widehat{D}(\mathcal{X}||\mathcal{Y})\|_1$. Our lower bound of Sect. 4.3 will require the following bound.

Lemma 2 (Theorem 1.3 [11]). *Let $\sigma_0, \sigma_1 > 0$. Then*

$$\Delta(\mathcal{N}(0,\sigma_0^2), \mathcal{N}(0,\sigma_1^2)) \geq \frac{1}{200} \min\left\{1, \frac{|\sigma_0^2 - \sigma_1^2|}{\sigma_0^2}\right\}. \tag{2}$$

2.2 Bit Security

We use the notion of bit security from [22], which we briefly review below.

Definition 5 (Indistinguishability Game). *Let $\{\mathcal{D}_\theta^0\}_\theta$, $\{\mathcal{D}_\theta^1\}_\theta$ be two distribution 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 θ_i to C, and obtain samples $c_i \leftarrow \mathcal{D}_{\theta_i}^b$. The goal of the adversary is to output $b' = b$.*

Definition 6 (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 $\mathsf{adv}^A = \alpha^A(\delta^A)^2$.*

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

As argued in [22], this is the correct way to define bit security for decision problems. Notice quadratic scaling with δ^A, rather than the linear scaling used for search problems. For additional motivation for the quadratic dependency, we note it matches known sample complexity lower bounds for distinguishing distributions that are close in the total variation distance, see Sect. 5.2 of [3].

Lemma 3 (Lemma 2 of [22]). *Let $\mathcal{H}_i$ be k distributions and $\mathcal{G}_{i,j}$ be the indistinguishability game instantiated with $\mathcal{H}_i$ and $\mathcal{H}_j$. Let C be a fixed constant. Let $\epsilon_{i,j} = \max_A \mathsf{adv}^A$ over all adversaries A against $\mathcal{G}_{i,j}$ with $T(A) \leq C$. Then $\epsilon_{1,k} \leq 3k \sum_{i=1}^{k-1} \epsilon_{i,i+1}$.*

The two distributions to be distinguished in a game $\mathcal{G}$ sometimes both post-process samples from some other probability ensamble $\mathcal{P}_\theta$. The following theorem bounds the loss of bit security of $\mathcal{G}$ if we replace $\mathcal{P}$ with another distribution $\mathcal{Q}$.

Theorem 1 (Theorem 8 of [22]). *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 κ-bit secure, and $\max_\theta D(\mathcal{P}_\theta||\mathcal{Q}_\theta) \leq 2^{-\kappa+1}$, then $\mathcal{G}^{\mathcal{Q}_\theta}$ is $(\kappa - 8)$-bit secure.*

The aforementioned theorem is stated more generally in [22]. Our specialization of it requires that $\delta(\mathcal{P}||\mathcal{Q}) = \sqrt{D(\mathcal{P}||\mathcal{Q})/2}$ is what [22] calls a λ-efficient measure, which is implicit in [2,30].

We will need a few novel bounds on the quantities previously mentioned in this sub-section. These bounds are simplest to describe in terms of the following divergence.

Definition 7 (Bit Security Divergence). *Let $\mathcal{X}, \mathcal{Y}$ be random variables supported on X. The* bit security divergence *between $\mathcal{X}$ and $\mathcal{Y}$ is the quantity*

$$\delta_{\mathsf{BS}}(\mathcal{X}, \mathcal{Y}) = \sup_{S \subseteq X} \frac{\Pr_{\mathcal{X}}[S] + \Pr_{\mathcal{Y}}[S]}{2} \Delta\left(\mathcal{X}|S, \mathcal{Y}|S\right)^2,$$

where $\mathcal{X}|S, \mathcal{Y}|S$ are the conditional distributions of $\mathcal{X}, \mathcal{Y}$, conditioned on the event S.

It is straightforward to verify that this is indeed a divergence, and moreover it is symmetric (which is why we write $\delta_{\mathsf{BS}}(\cdot, \cdot)$ rather than $\delta_{\mathsf{BS}}(\cdot||\cdot)$). It is not a metric, as the $O(k)$ factor in Lemma 3 is known to be tight, which is incompatible with $\delta_{\mathsf{BS}}(\cdot, \cdot)$ satisfying a triangle inequality.

$\delta_{\mathsf{BS}}(\cdot, \cdot)$ captures the advantage of an optimal (potentially computationally unbounded) adversary that aborts on the set S^c, and therefore can be seen as an extension of the standard total variation distance to the framework of [22]. We will need the following novel Pinsker-like bound on this quantity.

Lemma 4. *Let $\mathcal{X}, \mathcal{Y}$ be random variables supported on X. Then $\delta_{\mathsf{BS}}(\mathcal{X}, \mathcal{Y}) \leq D(\mathcal{X}||\mathcal{Y})/2$.*

We delay the proof of this to the full version of the paper. We can use this to bound the advantage of *computationally unbounded* adversaries in the indistinguishability game.

Lemma 5. *Let $\mathcal{G}$ be the indistinguishability game instantiated with distribution ensembles $\{\mathcal{X}_\theta\}_\theta, \{\mathcal{Y}_\theta\}_\theta$, 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*

$$\mathsf{adv}^A \leq \frac{q}{2} \max_{\theta \in \Theta} D(\mathcal{X}_\theta||\mathcal{Y}_\theta). \tag{3}$$

Proof. View an (adaptive) adversary as an arbitrary distribution on query-response pairs $\mathcal{X}_{\hat{\theta}} := ((\hat{\theta}_1, \mathcal{X}_{\hat{\theta}_1}), \ldots, (\hat{\theta}_q, \mathcal{X}_{\hat{\theta}_q}))$ (and similarly for $\mathcal{Y}_{\hat{\theta}}$). We then have that

$$\mathsf{adv}^A \leq \delta_{\mathsf{BS}}(\mathcal{X}_{\hat{\theta}}, \mathcal{Y}_{\hat{\theta}}) \leq \frac{1}{2} D(\mathcal{X}_{\hat{\theta}}, \mathcal{Y}_{\hat{\theta}}) \leq \frac{1}{2} \left\| \widehat{D}(\mathcal{X}_{\hat{\theta}}, \mathcal{Y}_{\hat{\theta}}) \right\|_1 \leq \frac{q}{2} \max_{\theta \in \Theta} D(\mathcal{X}_\theta||\mathcal{Y}_\theta). \tag{4}$$

□

2.3 Fully Homomorphic Encryption

We briefly review definitions related to FHE. For simplicity, we focus on public-key setting. In all our definitions, we denote the security parameter using κ.

Definition 8 (FHE Scheme). *A (public-key) homomorphic encryption scheme with plaintext space $\mathcal{M}$, ciphertext space $\mathcal{C}$, public key space $\mathcal{PK}$, secret-key space $\mathcal{SK}$, and space of evaluatable circuits $\mathcal{L}$ is a tuple of four probabilistic polynomial-time algorithms*

$$\begin{aligned}\mathsf{KeyGen} &: 1^{\mathbb{N}} \to \mathcal{PK} \times \mathcal{SK}\\ \mathsf{Enc} &: \mathcal{PK} \times \mathcal{M} \to \mathcal{C}\\ \mathsf{Dec} &: \mathcal{SK} \times \mathcal{C} \to \mathcal{M}\\ \mathsf{Eval} &: \mathcal{PK} \times \mathcal{L} \times \mathcal{C} \to \mathcal{C}\end{aligned}$$

Typically the public key naturally splits into two components, one used by Enc and one used by Eval. This separation is used to minimize the storage requirements of encryption (as the evaluation key is often quite large), and has no impact on security, so for simplicity we model both Enc and Eval as taking as input the same public key.

Standard FHE schemes are expected to satisfy the following notion of correctness.

Definition 9 (Correctness). *An FHE scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is* correct *for some class of circuits $\mathcal{L}$ if for all $m_1, \ldots, m_k \in \mathcal{M}$, for all $C \in \mathcal{L}$, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, we have that*

$$\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_k))) = C(m_1, \ldots, m_k). \tag{5}$$

One can relax the notion of correctness to *statistical* correctness, where the above identity only holds with high probability (over the random coins of Enc and Eval). We will not make a distinction between these two notions.

The work [8] introduced an "approximate" FHE scheme (CKKS), for which Eq. (5) does not hold. The security implications of this relaxation are investigated in [19], as discussed below. However, neither [8] nor [19] provide a formal definition of an "approximate" FHE scheme, and instead simply drop the correctness requirement (5) without any further restriction. This is despite the CKKS scheme satisfying an approximate version of the correctness property of Eq. (5).

The definition of *approximately correct* FHE scheme plays a fundamental role in our work. Informally, an approximately correct FHE scheme allows for meaningful, but inexact, computation on encrypted messages. To formalize the relaxed correctness requirements of an approximately correct FHE scheme, we first define the *ciphertext error*, which specifies the extent to which a homomorphic computation fails to be exact.

Definition 10 (Ciphertext Error). *Let $\mathit{\Pi} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be an FHE scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, which is a normed space with norm $\|\cdot\| : \widetilde{\mathcal{M}} \to \mathbb{R}_{\geq 0}$. For any ciphertext ct, secret key sk, and message m, the* ciphertext error *of $(\mathsf{ct}, m, \mathsf{sk})$ is defined to be*

$$\mathsf{Error}(\mathsf{ct}, m, \mathsf{sk}) = \|\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) - m\|. \tag{6}$$

Typically, for some circuit $C \in \mathcal{L}$, key pair $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, and input values $m_1, \ldots, m_k \in \mathcal{M}$, one is interested in the quantity $\mathsf{Error}(\mathsf{ct}, m, \mathsf{sk})$ for

$$m = C(m_1, \ldots, m_k), \quad \text{and,} \quad \mathsf{ct} = \mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_k)),$$

i.e. where m and ct correspond to the same computation done on plaintexts and ciphertexts.

In this work we investigate two distinct correctness properties for approximate homomorphic encryption. The first is implicit in the literature on CKKS. We call this notion "static" to contrast with a later notion we investigate in Sect. 5.

Definition 11 (Static Approximate Correctness). *Let $\mathit{\Pi}$ be an FHE scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, which is a normed space with norm $\|\cdot\| : \widetilde{\mathcal{M}} \to \mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ be a space of circuits, $\mathcal{L}_k \subseteq \mathcal{L}$ the subset of parity k circuits, and let $\mathsf{Estimate} : \bigsqcup_{k \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ be an efficiently computable function. We call the tuple $\tilde{\mathit{\Pi}} = (\mathit{\Pi}, \mathsf{Estimate})$ a* statically approximate *FHE scheme if for all $k \in \mathbb{N}$, for all $C \in \mathcal{L}_k$, for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$, if $\mathsf{ct}_1, \ldots, \mathsf{ct}_k$ and $m_1, \ldots, m_k$ are such that $\mathsf{Error}(\mathsf{ct}_i, m_i, \mathsf{sk}) \leq t_i$, then*

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

Note that the type signature $\bigsqcup_{k \in \mathbb{N}} \mathcal{L}_k \times \mathbb{R}^k_{\geq 0} \to \mathbb{R}_{\geq 0}$ encodes that $\mathsf{Estimate}$ takes as input a circuit C, and an error bound t_i for each of the k input wires to the circuit $C \in \mathcal{L}_k$. This correctness notion is "static" in the sense of static typing. In particular, $\mathsf{Estimate}$ only depends on

- the computation C to be done, and
- error bounds t_i for the inputs to the homomorphic computation.

All of these quantities are publicly computable given an abstract description of a computation, and (for non-adaptive computations) can even be precomputed (say by an FHE "compiler").

Generally $\mathsf{Estimate}(\cdot)$ either computes a (provable) worst-case bound on the error, or a (heuristic) average-case bound. Our work assumes worst-case bounds (although we discuss average-case bounds some in Sect. 6). Approximate FHE schemes often require that all $m_1, \ldots, m_k$ are of bounded norm—this can be captured in the above definition by choosing $\mathcal{M}$ to be a set of bounded norm.

Security. We use the following security definition, proposed in [19], which properly captures security of approximate FHE schemes against passive attacks.

Algorithm 1 Oracles for the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ game.

```
initialization
    $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$
global state
    $S \leftarrow \emptyset$
    $i \leftarrow 0$
$\mathsf{E}^b_{\mathsf{pk}}(m_0, m_1) :=$
    $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b)$
    $S[i] \leftarrow (m_0, m_1, \mathsf{ct})$
    $i \leftarrow i + 1$
    return $\mathsf{ct}$
$\mathsf{H}^b_{\mathsf{pk}}(g, \mathbf{J} = (j_1, \ldots, j_k)) :=$
    $\mathsf{ct} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(g, S[j_1].\mathsf{ct}, \ldots, S[j_k].\mathsf{ct})$
    $gm_0 \leftarrow g(S[j_1].m_0, \ldots, S[j_k].m_0)$
    $gm_1 \leftarrow g(S[j_1].m_1, \ldots, S[j_k].m_1)$
    $S[i] \leftarrow (gm_0, gm_1, \mathsf{ct})$
    $i \leftarrow i + 1$
    return $\mathsf{ct}$
$\mathsf{D}^b_{\mathsf{sk}}(i) :=$
    if $S[i].m_0 = S[i].m_1$
        return $\mathsf{Dec}_{\mathsf{sk}}(S[i].\mathsf{ct})$
    else
        return $\perp$
```

Definition 12 ($\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ Security, [19]). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a FHE scheme. We define the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ game to be an indistinguishability game parameterized by distribution ensembles $\{(\mathsf{E}^b_\theta, \mathsf{H}^b_\theta, \mathsf{D}^b_\theta)\}_\theta$ for $b \in \{0, 1\}$, where these oracles are the (stateful[1]) oracles given in Algorithm 1. The scheme Π is κ-bit* $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-secure *if for any A, we have that $\kappa \leq \log_2 \frac{T(A)}{\mathsf{adv}^A}$, where* adv^A *is as in Definition 6.*

In [19] it is also shown that for FHE schemes satisfying the standard correctness requirement (5), $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security is equivalent to the traditional formulation of indistinguishability under chosen plaintext attack ($\mathsf{IND\text{-}CPA}$), defined as follows.

Definition 13 ($\mathsf{IND\text{-}CPA}$ Security). *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a FHE scheme. We define the $\mathsf{IND\text{-}CPA}$ game to be an indistinguishability game parameterized by distribution ensembles $\{\mathsf{E}^b_\theta\}_\theta$ for $b \in \{0, 1\}$ of Algorithm 1. The scheme Π is κ-bit* $\mathsf{IND\text{-}CPA}$-secure *if for any A, we have that $\kappa \leq \log_2 \frac{T(A)}{\mathsf{adv}^A}$, where* adv^A *is as in Definition 6.*

We will additionally use weaker and stronger variants of $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$, informally defined as follows:

- q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security. This is the same as $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security, but restricted to adversaries that make at most $q(\kappa)$ queries to oracle D.

[1] As a standard convention (for this and other games defined in the paper), if at any point in a game the adversary makes an invalid query (*e.g.*, a circuit g not supported by the scheme, or indices out of range), the oracle simply returns an error symbol $\perp$.

- KR^{D} security, or security against key recovery attacks. Here we modify the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ game by restricting[2] the E oracle to queries of the form $\mathsf{E}(m, m)$, and requiring the adversary to output (at the end of the attack) a secret key sk', rather than the bit b'. The attack is successful if $\mathsf{sk} = \mathsf{sk}'$.

KR^{D} security is implied by $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security, but it is much weaker, and it is not generally considered a satisfactory notion of security. Here (as in [19]), KR^{D} security is used exclusively to show that certain schemes are not secure, making the insecurity results stronger. We provide formal definitions of the above notions in the full version of the paper.

3 A Differentially Private Approach to $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ Security

In this section we investigate achieving q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security for statically approximate, $\mathsf{IND\text{-}CPA}$-secure FHE schemes $\tilde{\Pi}$. Our approach is to post-process decryptions of $\tilde{\Pi}$ with an appropriate notion of differential privacy. The noise added by this differentially private mechanism will suffice to information-theoretically hide the ciphertext error, allowing us to reduce our analysis to the case of exact FHE, where $\mathsf{IND\text{-}CPA}$ and q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security are equivalent.

3.1 Our Notion of Differential Privacy

Our notion of differential privacy is a generalization of the notion of Rényi differential privacy [23] to different norms[3]. As the tightest bounds in our setting occur in the simplest[4] case when $\alpha = 1$, we present things solely in terms of this Rényi divergence, *i.e.* the KL divergence.

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

$$D(M_t(x)||M_t(x')) \leq \rho. \tag{7}$$

Note that our mechanism M depends on a bound on the distance $\|x - x'\| \leq t$, which it uses (internally) to set parameters to meet the desired privacy bound.

[2] This is without loss of generality, as the only point of general queries $\mathsf{E}(m, m')$ is to get information correlated with the secret bit b, which the game does not depend on.

[3] In Differential Privacy, "adjacent" values are typically measured in the Hamming norm, while for our purposes the ℓ_2 and ℓ_∞ norms are of primary interest.

[4] There is an alternative simplification of the Rényi divergence when $\alpha = \infty$ known as the *max-log distance* [21] with desirable properties, for example it is a metric, similarly to the statistical distance. As our bounds degrade linearly in α as $\alpha \to \infty$, this notion is unsuitable for our situation.

In the most common case of Gaussian noise, it will use noise of standard deviation $\sigma = \Omega(2^{\kappa/2}t)$ to achieve κ-bit security (Corollary 1).

As $\|x - x'\| = \|x' - x\|$ is itself symmetric, our definition is invariant under replacing $D(\mathcal{D}_0||\mathcal{D}_1)$ with $\max(D(\mathcal{D}_0||\mathcal{D}_1), D(\mathcal{D}_1||\mathcal{D}_0))$, and is therefore implicitly dependent on this larger (symmetric) measure, although we do not make this explicit in our work.

Algorithm 2 The FHE Scheme $M[\tilde{\Pi}]$

```
$\mathsf{Enc}'_{\mathsf{pk}}(m) :=$
    $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$
    return $\mathsf{ct} = (c, t_e)$
$\mathsf{Eval}'_{\mathsf{pk}}(C, \mathsf{ct}'_1, \ldots, \mathsf{ct}'_k) :=$
    $c \leftarrow \mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{ct}_1.c, \ldots, \mathsf{ct}_k.c)$
    $t \leftarrow \mathsf{Estimate}(C, \mathsf{ct}_1.t, \ldots, \mathsf{ct}_k.t)$
    return $\mathsf{ct} = (c, t)$

$\mathsf{Dec}'_{\mathsf{sk}}(\mathsf{ct}) :=$
    return $M_{\mathsf{ct}.t}(\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}.c))$
```

Definition 15. *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be an FHE scheme with plaintext space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, where $\widetilde{\mathcal{M}}$ is a normed space with norm $\|\cdot\|$. Let* $\mathsf{Estimate}$ *be such that $\tilde{\Pi} = (\Pi, \mathsf{Estimate})$ is statically approximate, and let t_e be an upper bound on ciphertext errors of all fresh encryptions $\mathsf{Enc}_{\mathsf{pk}}(m)$ for all $m \in \mathcal{M}$. Let M_t be a ρ-KLDP mechanism on $\widetilde{\mathcal{M}}$. Define the FHE scheme $M[\tilde{\Pi}]$ that has an identical* KeyGen *algorithm to Π, with the modified algorithms $\mathsf{Enc}'_{\mathsf{pk}}, \mathsf{Eval}'_{\mathsf{pk}}$, and $\mathsf{Dec}'_{\mathsf{sk}}$ of Algorithm 2.*

In the above definition of the scheme $M[\tilde{\Pi}]$, we use the "tagged ciphertext" notation $\mathsf{ct} = (c, t)$, where c is an ordinary ciphertext and t is an estimated ciphertext error upper bound. An initial estimation t_e is provided by the encryption algorithm, and the evaluation algorithm updates the error upper bound using $\mathsf{Estimate}(\cdot)$ such that the resulting scheme is a statically approximate FHE scheme.

Algorithm 3 The decryption oracle for the game $\mathcal{G}_1$ of Theorem 2.

```
$\mathsf{D}(i) :=$
    if $S[i].m_0 = S[i].m_1$
        $t_i \leftarrow S[i].\mathsf{ct}.t$
        return $M_{t_i}(S[i].m_0)$
    else
        return $\perp$
```

Theorem 2. *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be an FHE scheme with plaintext space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, where $\widetilde{\mathcal{M}}$ is a normed space with norm $\|\cdot\|$. Let* $\mathsf{Estimate}$ *be such* *that* $\tilde{\Pi} = (\Pi, \mathsf{Estimate})$ *is statically approximate. Let* $\kappa > 0$*, let* M_t *be a* ρ*-KLDP mechanism on* $\widetilde{\mathcal{M}}$ *where* $\rho \leq 2^{-\kappa-7}/q$*, and let* $q \in \mathbb{N}$*. If* Π *is* $(\kappa+8)$*-bit secure in the* $\mathsf{IND\text{-}CPA}$ *game, then* $M[\tilde{\Pi}]$ *is* κ*-bit secure in the* q*-*$\mathsf{IND\text{-}CPA}^D$ *game.*

Proof. We defer details to the full version of the paper, and simply note that this follows from combining Lemma 5 and Theorem 1. □

3.2 Gaussian Mechanism

In this section, we present and analyze a differentially private mechanism M_t which simply adds Gaussian noise to its input.

Definition 16. *Let* $\mu \in \mathbb{Z}$*, and* $\sigma > 0$*. The discrete Gaussian of parameters* μ, σ *(written* $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$*) is the probability distribution supported on* $\mathbb{Z}$ *with p.m.f.* $p(x) \propto \exp(-(x-\mu)^2/2\sigma^2)$*.*

It is known how to (with high probability) exactly sample from this distribution in constant time [4]. We explicitly bound the impact of this on the security of our constructions in the full version of our paper.

Proposition 1 (Prop. 5 of [4]). *Let* $\sigma \in \mathbb{R}_{\geq 0}$*, and let* $\mu, \nu \in \mathbb{Z}$*. Then:*

$$D(\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2) || \mathcal{N}_{\mathbb{Z}}(\nu, \sigma^2)) = \frac{(\nu-\mu)^2}{2\sigma^2}. \tag{8}$$

Definition 17. *Let* $\rho > 0$*, and* $n \in \mathbb{N}$*. Define the* (discrete) Gaussian Mechanism $M_t : \mathbb{Z}^n \to \mathbb{Z}^n$ *be the mechanism that, on input* $x \in \mathbb{Z}^n$*, outputs a sample from* $\mathcal{N}_{\mathbb{Z}^n}(x, \frac{t^2}{2\rho} I_n)$*.*

Lemma 6. *For any* $\rho > 0, n \in \mathbb{N}$*, the Gaussian mechanism is* ρ*-KLDP.*

Proof. Let $\mathcal{X} = \mathcal{N}_{\mathbb{Z}^n}(x, \frac{t^2}{2\rho} I_n)$ and $\mathcal{Y} = \mathcal{N}_{\mathbb{Z}^n}(y, \frac{t^2}{2\rho} I_n)$. By sub-additivity of the KL divergence and Proposition 1, we have that $D(\mathcal{X}||\mathcal{Y}) \leq \|\widehat{D}(\mathcal{X}||\mathcal{Y})\|_1 = \frac{\rho}{t^2} \|x-y\|_2^2 \leq \rho$. □

Corollary 1. *Let* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be an FHE scheme with plaintext space* $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$*, where* $\widetilde{\mathcal{M}} \subseteq \mathbb{Z}^n$ *is a normed space with norm* $\|\cdot\|$*. Let* $\mathsf{Estimate}$ *be such that* $\tilde{\Pi} = (\Pi, \mathsf{Estimate})$ *is a statically approximate FHE scheme. Let* M_t *be the Gaussian mechanism (with* $\rho := 2^{-\kappa-7}/q$*). If* Π *is* $(\kappa+8)$*-bit secure in the* $\mathsf{IND\text{-}CPA}$ *game, then* $M[\tilde{\Pi}]$ *is* κ*-bit secure in the* q*-*$\mathsf{IND\text{-}CPA}^D$ *game.*

For $\rho := 2^{-\kappa-7}/q$, one can check that the Gaussian mechanism adds noise of standard deviation $8\sqrt{q2^{\kappa}}\mathsf{ct}.t$ to each coordinate, so one loses $\kappa/2+3+\log_2\sqrt{q}+\log_2 \mathsf{ct}.t$ bits of precision. As the ciphertext already contains $\log_2 \mathsf{ct}.t$ bits of noise, the *additional* precision lost by $M[\tilde{\Pi}]$ is $\kappa/2 + \log_2\sqrt{q} + 3$ bits.

Proof. This reduces to combining Lemma 6 with Theorem 2. The size of the Gaussian noise comes from $\rho = \mathsf{ct}.t^2/2\sigma^2 \iff \sigma = \frac{1}{\sqrt{2\rho}}\mathsf{ct}.t$. As we need that $\rho \leq 2^{-\kappa-7}/q$, it follows that $\sigma \geq 8\sqrt{q}2^{\kappa/2}\mathsf{ct}.t$. □

This transformation does not explicitly depend on the underlying parameters of the particular implementation of approximate encryption (for example, the size of the LWE moduli one is working over, the dimension of the message space, etc.), and instead only implicitly depends on these quantities via the computation of the static ciphertext error bound. We caution that to apply this result to CKKS one needs to be slightly careful about the underlying norm one is working with, which we do later in Theorem 3.

4 Application to CKKS

Prior work of [19] shows that the approximate FHE scheme of [8] does not satisfy $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security, even though it satisfies $\mathsf{IND\text{-}CPA}$-security. We refer the reader to [19] for additional details, but at a high level they show that publishing the results of an approximate FHE computation under CKKS leaks information about the secret key, enabling a full key recovery attack in the case of trivial computation, and an attack against $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security for more general homomorphic computation. In this section, we apply Theorem 2 and Lemma 6 to give a modification of the CKKS decryption function that allows us to prove $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security of the modified scheme.

We use the results of Sect. 3 to show that post-processing the results of the CKKS decryption function with the Gaussian mechanism is sufficient to achieve $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security for the CKKS scheme, for large enough Gaussian noise (Sect. 4.2). We also prove a nearly matching lower bound on the Gaussian noise necessary to achieve $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security for the CKKS scheme (Sect. 4.3). We then investigate a relaxed notion of security (Sect. 4.4), which may be of independent interest. With these results, we briefly examine the countermeasures adopted by some open-source implementations of CKKS, and we suggest concrete parameters (Sect. 4.5).

4.1 The CKKS Approximate FHE Scheme

We begin with a few mathematical preliminaries necessary to the CKKS scheme. For any positive integer N, let $\Phi_N(X) = \prod_{j \in \mathbb{Z}_N^*}(X - \omega_N^j)$ be the Nth cyclotomic polynomial, where $\omega_N = e^{2\pi \mathsf{i}/N} \in \mathbb{C}$ is the complex Nth principal root of unity, and $\mathbb{Z}_N^*$ is the group of invertible integers modulo N. We recall that $\Phi_N(X) \in \mathbb{Z}[X]$ is a monic polynomial of degree $n = \varphi(N) = |\mathbb{Z}_N^*|$ with integer coefficients. We denote by $\mathcal{R}^N = \mathbb{Z}[X]/(\Phi_N(X))$ the ring of integers of the number field $\mathbb{Q}[X]/(\Phi_N(X))$, omitting the superscript when it is clear from context. We use $\mathcal{R}_Q^N = \mathbb{Z}[X]/(Q, \Phi_N(X))$ to denote the ring of elements of $\mathcal{R}^N$ reduced modulo Q.

An element $a \in \mathbb{R}[X]/(\Phi_N(X))$ may be embedded into $\mathbb{C}^n$ under the *canonical embedding* $\tau(a)$ (typically defined over $\mathbb{Q}[X]/(\Phi_N(X))$, but naturally extending to $\mathbb{R}[X]/(\Phi_N(X))$). The map $\tau(a)$ takes a to the $n = \varphi(N)$ evaluations of a at the n roots of $\Phi_N(X)$. Notice that these n values come in conjugate pairs and can be identified as a vector in $\mathbb{C}^{n/2}$ via a projection $\pi : (z, \bar{z}) \mapsto z$. So, complex vectors in $\mathbb{C}^{n/2}$ are considered as messages in CKKS, and they are encoded to plaintext polynomials in $\mathcal{R}$ by composing π^{-1} and τ^{-1} together with a scaling factor; conversely, plaintexts are decoded using $\tau \circ \pi$. We define the *canonical embedding norm* $\|\cdot\|_\infty^{\mathsf{can}}$ of an element $a \in \mathbb{R}[X]/(\Phi_N(X))$ to be $\|a\|_\infty^{\mathsf{can}} = \|\tau(a)\|_\infty$. We will use this norm to track the ciphertext error of CKKS ciphertexts.

We now present the relevant subroutines of the CKKS FHE scheme. We omit many details of the CKKS scheme, and refer the reader to [8] for a more complete description. The CKKS scheme is parameterized by a plaintext dimension $n/2$ (typically a power-of-two), a ciphertext modulus Q, and a discrete Gaussian error distribution χ with standard deviation σ.

- $\mathsf{CKKS.KeyGen}(1^\kappa)$: Take $w = w(\kappa)$ and $p = p(\kappa, Q)$. To generate the secret key sk, sample $s \leftarrow \{s \in \{-1, 0, 1\}^n : |s|_0 = w\}$ and take $\mathsf{sk} = (1, s)$. To generate the public key pk, sample $a \leftarrow \mathcal{R}_Q$, $e \leftarrow \chi$, and take $\mathsf{pk} = (b = -as + e, a)$. To generate the evaluation key ek, sample $a' \leftarrow \mathcal{R}_{pQ}$, $e' \leftarrow \chi$, and take $\mathsf{ek} = (b', a')$ for $b' = -a's + e' + ps^2 \mod pQ$. Return $(\mathsf{sk}, \mathsf{pk}, \mathsf{ek})$.
- $\mathsf{CKKS.Encode}(\mathbf{x} \in \mathbb{C}^{n/2}; \varDelta)$: Return $\lfloor \varDelta \cdot \varphi^{-1}(\mathbf{x}) \rceil \in \mathcal{R}$.
- $\mathsf{CKKS.Enc}_{\mathsf{pk}}(m)$: Let T denote the distribution over $\{0, \pm 1\}^n$ induced by sampling each coordinate independently, drawing -1 with probability 1/4, 1 with probability 1/4, and 0 with probability 1/2. Sample $r \leftarrow T$, $e_0, e_1 \leftarrow \chi$, and return $r \cdot \mathsf{pk} + (m + e_0, e_1) \mod Q$.
- $\mathsf{CKKS.Add}(\mathbf{c}_0, \mathbf{c}_1 \in \mathcal{R}_Q)$: Return $\mathbf{c}_0 + \mathbf{c}_1 \mod Q$.
- $\mathsf{CKKS.Mult}_{\mathsf{ek}}(\mathbf{c}_0, \mathbf{c}_1 \in \mathcal{R}_Q)$: For $\mathbf{c}_0 = (b_0, a_0)$ and $\mathbf{c}_1 = (b_1, a_1)$, let $(b_2, a_2) = (b_0 b_1, a_0 b_1 + a_1 b_0) + \lfloor p^{-1} \cdot a_0 a_1 \cdot \mathsf{ek} \rceil \mod Q$. Return (b_2, a_2).
- $\mathsf{CKKS.Decode}(a \in \mathcal{R}; \varDelta)$: Return $\varphi(\varDelta^{-1} \cdot a) \in \mathbb{C}^{n/2}$.
- $\mathsf{CKKS.Dec}_{\mathsf{sk}}(\mathbf{c})$: For $\mathbf{c} = (b, a) \in \mathcal{R}_Q^2$, return $b + as \mod Q$.

Note that CKKS supports encryption and decryption of floating-point inputs by pre-processing encryption with CKKS.Encode, and post-processing decryption with CKKS.Decode. All intermediate operations are then done with integer arithmetic. To simplify exposition, we focus on these intermediate operations, and therefore restrict to the case of integer arithmetic.

We will need the following (standard) expressions for how the ciphertext error transforms during addition and multiplication.

Lemma 7 (Error Growth [8]). *Let* $\mathbf{c}_0$ *and* $\mathbf{c}_1$ *denote two* CKKS *ciphertexts, with* $\mathbf{c}_0 = \mathsf{CKKS.Enc}_{\mathsf{pk}}(m_0)$ *and* $\mathbf{c}_1 = \mathsf{CKKS.Enc}_{\mathsf{pk}}(m_1)$ *with errors* e_0 *and* e_1 *respectively. Then the ciphertext* $\mathbf{c}_{\mathsf{Mult}} = \mathsf{CKKS.Mult}(\mathbf{c}_0, \mathbf{c}_1)$ *has error* $m_0 e_1 + m_1 e_0 + e_0 e_1 + e_{\mathsf{Mult}}$ *for a term* e_{Mult} *that depends on the parameters of the* CKKS *instance (and the ciphertexts* $\mathbf{c}_0, \mathbf{c}_1$*). The ciphertext* $\mathbf{c}_{\mathsf{Add}} = \mathsf{CKKS.Add}(\mathbf{c}_0, \mathbf{c}_1)$ *has error* $e_0 + e_1$.

Certain authors have suggested various heuristics for analyzing e_{Mult}. We will find the following one useful for the analysis of the attack of Sect. 4.3.

Heuristic 1 (Appendix A.5 of [12]). *Let* w *be the hamming weight of* sk. *Then* e_{Mult} *may be modeled as a random variable with mean zero and variance* $O(wn)$.

The rest of our work will benefit from the following notation.

Definition 18. *For $\sigma > 0$, let $\mathsf{S\text{-}CKKS}_\sigma$ be the CKKS encryption scheme, where one modifies decryption to compute $\mathsf{S\text{-}CKKS}_\sigma.\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) = \mathsf{CKKS}.\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}.c) + \mathcal{N}_{\mathbb{Z}^n}(0, \sigma^2 \mathsf{ct}.t^2 I_n)$.*

4.2 IND-CPA$^{\mathsf{D}}$-Secure CKKS

It is straightforward to apply Corollary 1 to CKKS to obtain q-IND-CPA$^{\mathsf{D}}$ security.

Theorem 3. *For any $q \in \mathbb{N}$, if CKKS is $(\kappa+8)$-bit IND-CPA-secure, and $\sigma = 8\sqrt{qn}2^{\kappa/2}$, then $\mathsf{S\text{-}CKKS}_\sigma$ is κ-bit q-IND-CPA$^{\mathsf{D}}$-secure, i.e. $\kappa/2 + \tilde{O}(1)$ additional bits of Gaussian noise suffice to achieve q-IND-CPA$^{\mathsf{D}}$ security.*

Proof. This follows immediately from Corollary 1, (using the aforementioned inequality $\|m\|_\infty^{\mathsf{can}} \leq \sqrt{n}\,\|m\|_2$, as our analysis of the Gaussian mechanism uses an ℓ_2 norm bound). □

4.3 Lower Bound for Gaussian Mechanism

Together, Lemma 6 and Theorem 2 give an upper bound on the amount of Gaussian noise required to achieve IND-CPA$^{\mathsf{D}}$-security for an IND-CPA-secure approximate encryption scheme. In this subsection, we show that this upper bound is essentially tight for CKKS by demonstrating an attack against IND-CPA$^{\mathsf{D}}$ security for noticeably smaller Gaussian noise, *i.e.* analyzing $\mathsf{S\text{-}CKKS}_{\sigma_s}$ for sanitization noise $\sigma_s \ll 8\sqrt{qn}2^{\kappa/2}$. In what follows, recall that $n = \varphi(N)$, and w denotes the Hamming weight of the key sk.

Algorithm 4 Adversary $A(1^\kappa, \mathsf{pk}, \mathsf{ek})$

```
for $i \in \{0, \dots, 44\}$ do
    $\mathsf{ct}_i \leftarrow \mathsf{E}_{\mathsf{pk}}(m_i^{(0)} = 0, m_i^{(1)} = B)$;
end for
for $i \in \{45, \dots, 59\}$ do
    $\mathsf{ct}_i \leftarrow \mathsf{E}_{\mathsf{pk}}(m_i^{(0)} = 0, m_i^{(1)} = -B)$;
end for
$\mathsf{ct}_{60} \leftarrow \mathsf{H}_{\mathsf{ek}}(g, \{0, \dots, 59\})$ for $g(x_0, \dots, x_{59}) = \sum_{i=0}^{29}(x_i \cdot x_{30+i})$
$m' \leftarrow \mathsf{D}_{\mathsf{sk}}(60)$
$V_0 = 30\sigma^4 + O(wn) + \sigma_s^2$                    Variance of $\tau(m')_0$ if $b = 0$
$V_1 = 30\sigma^4 + 60B^2\sigma^2 + O(wn) + \sigma_s^2$    Variance of $\tau(m')_0$ if $b = 1$
if $|\tau(m')_0| < \sqrt{\frac{\log(V_1/V_0)V_0V_1}{V_1 - V_0}}$ then
    return 0
else
    return 1
end if
```

At a high level, the adversary A will exploit the message-dependence of the $\mathsf{S\text{-}CKKS}$ error growth (Lemma 7) to design an H query such that the expected magnitude of the ciphertext error of ct_{60} is larger when $b = 1$ than when $b = 0$. The adversary A will then query D on this ciphertext, and choose its bit based on the size of the message m' it receives.

We will next show that the aforementioned adversary will have noticeable advantage unless σ_s is larger than σ (the standard deviation of the underlying RLWE error) by a factor super-polynomial in the security parameter.

Lemma 8. *Let $\sigma_s > 0$. Then there exists an adversary A against $\mathsf{S\text{-}CKKS}_{\sigma_s}$ in the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ such that $\mathsf{adv}^A = \Omega\left(\frac{1}{\sigma_s^4 n^6}\right)$.*

Proof. We first observe that the ciphertext $\mathsf{ct}_{60} = \mathsf{Eval}_{\mathsf{ek}}(g, \{0, \ldots, 59\})$ is an approximate encryption of 0 both when $b = 0$ and $b = 1$ in the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ experiment. Therefore the decryption query made by A returns a value rather than $\perp$.

If $b = 0$, then because all ciphertexts ct_i encrypt messages $m_i = 0$, the message-dependent terms of the error growth from Lemma 7 are also 0, and so the ciphertext error of ct_{60} is $\sum_{i=0}^{29} e_{\mathsf{Mult}} + e_i e_{30+i}$, where e_i denotes the ciphertext error of ct_i. Recall that if error vectors e and e' have entries sampled from a discrete Gaussian with parameter σ, then each of the components of $\tau(ee')$ is distributed with mean 0 and variance σ^4. We can then use the Central Limit Theorem to approximate the distribution of the sum $\sum_{i=0}^{29} e_{\mathsf{Mult}} + e_i e_{30+i}$ as a Gaussian distribution with mean 0 and variance $30\sigma^4 + O(wn)$. Note that this approximation can be improved by increasing the number of terms in the sum to a larger constant. For the sake of concreteness we have designed the adversary such that there are 30 terms, as this is the value at which the Central Limit Theorem is empirically justified.

If $b = 1$, then the message-dependent terms of the error growth are significant, and the error of ct_{60} is

$$\sum_{i=0}^{14} \left(e_{\mathsf{Mult}} + e_i e_{30+i} + Be_i + Be_{30+i}\right) + \sum_{i=15}^{29} \left(e_{\mathsf{Mult}} + e_i e_{30+i} - Be_i + Be_{30+i}\right).$$

As in the case where $b = 0$, we will approximate this distribution as a Gaussian with mean 0. Though the error terms $e_i e_{30+i}$ and $Be_i + Be_{30+i}$ are not independent, they do have covariance 0, as do the terms $e_i e_{30+i}$ and $Be_{30+i} - Be_i$, and so we can approximate the sum of errors as being drawn from a discrete Gaussian distribution with mean 0 and variance $30\sigma^4 + 60B^2\sigma^2 + O(wn)$.

The adversary sees the result of post-processing the error term with the Gaussian mechanism, run with parameter σ_s, and then chooses its bit to return based on the absolute value of the first component $\tau(m')_0$ under the canonical embedding. When $b = 0$, this means the adversary sees a sample drawn from a distribution that is well-approximated by a centered Gaussian with variance $V_0 = 30\sigma^4 + O(wn) + \sigma_s^2 \mathsf{ct}.t^2$. When $b = 1$, however, the adversary sees a sample drawn from a distribution that is well-approximated by a Gaussian with the same mean, but larger variance $V_1 = 30\sigma^4 + 60B^2\sigma^2 + O(wn) + \sigma_s^2 \mathsf{ct}.t^2$. Let

$$x = \sqrt{\frac{\log(V_1/V_0)V_0V_1}{V_1 - V_0}}.$$

A straightforward calculation shows that for $|\tau(m')_0| < x$, m' is a more likely outcome when $b = 0$ than when $b = 1$, and when $|\tau(m')_0| \geq x$, m' is at least as likely when $b = 1$ as it is when $b = 0$. Then we have that the advantage of adversary A is approximately the total variation distance between a Gaussian with variance V_0 and a Gaussian with variance V_1. By Lemma 2, we have that

$$\Delta(\mathcal{N}(0, V_0), \mathcal{N}(0, V_1)) \geq \frac{1}{200}\frac{|V_0 - V_1|}{V_0} \in \Theta\left(\frac{B^2\sigma^2}{\sigma^4 + wn + \sigma_s^2\mathsf{ct}.t^2}\right).$$

Recall that w is the hamming weight of the secret key sk, and so we have $w < n$. For security, we know that $\sqrt{n} < \sigma$, and so it follows that the advantage of our (non-aborting) adversary A against the $\mathsf{IND\text{-}CPA^D}$ security of CKKS is the *square* of the total variation distance, *i.e.* $\Theta\left(\frac{B^4\sigma^4}{(\sigma^4+\sigma_s^2\mathsf{ct}.t^2)^2}\right)$. Finally, note that for $\|e_i\|_\infty^{\mathsf{can}} < \sigma n$ holds with high probability, so $\mathsf{ct}.t \leq O(B\sigma n^{3/2})$ (where we pick up a $\sqrt{n}$ factor to convert to the ℓ_2 norm), and therefore the advantage of our adversary is $\Theta\left(\frac{B^4\sigma^4}{\sigma^8+\sigma_s^4\sigma^4B^4n^6}\right) = \Omega\left(\frac{1}{\sigma_s^4n^6}\right)$. □

Theorem 4. *If $\mathsf{S\text{-}CKKS}_{\sigma_s}$ is κ-bit IND-CPA^D-secure, then $\sigma_s = \Omega(2^{\kappa/4}/n^{3/2})$, i.e. one must add at least $\kappa/4 - \tilde{\Omega}(1)$ bits of additional Gaussian noise.*

Proof. We have that $\kappa \leq \log_2 O\left(\frac{T(A)}{\mathsf{adv}^A}\right) \leq \log_2 O(\sigma_s^4n^6) \implies \sigma_s \geq 2^{\kappa/4}/n^{3/2}$, and therefore $\kappa/4 - \log_2 \Omega(n^{3/2}) \leq \log_2 \sigma_s$. □

We therefore see that while one can potentially improve on the concrete countermeasure of Sect. 4.5, the main (exponential) term is within a constant factor of correct.

4.4 Improved Parameters via a Relaxed Security Notion

The previous sections show that we require between $\kappa/4$ and $\kappa/2$ bits of Gaussian noise to achieve κ-bit q-$\mathsf{IND\text{-}CPA^D}$-security. We next introduce a relaxed notion of security, for which we can justify a reduction in the size of Gaussians one must add to obtain a form of q-$\mathsf{IND\text{-}CPA^D}$ security.

Definition 19. *Let Π be a cryptographic primitive, and $\mathcal{G}$ be an indistinguishability game. Let adv^A be the advantage of an adversary A in breaking the security of Π in the $\mathcal{G}$ game. We say that Π has (c, s)-bits of $\mathcal{G}$-security if, for any adversary A, either*

$$\log_2 \frac{T(A)}{\mathsf{adv}^A} \geq c, \qquad \textit{or} \qquad \log_2 \frac{1}{\mathsf{adv}^A} \geq s. \tag{9}$$

This notion may be equivalently written in a number of ways.

Definition 20. *Let $I \subseteq [0,1]$. A cryptographic primitive Π is said to be $(t(\epsilon), \epsilon)_I$-secure in an indistinguishability game $\mathcal{G}$ if, for any $\epsilon \in I$, any adversary of advantage ϵ has running time at least $t(\epsilon)$.*

Lemma 9. *Let Π be a cryptographic primitive, and $\mathcal{G}$ be an indistinguishability game. Then the following are equivalent*

1. *Π has (c, s)-bits of $\mathcal{G}$-security,*
2. *For any adversary A, $c \leq \log_2 \frac{\max(T(A), 2^{c-s})}{\mathsf{adv}^A}$, and*
3. *Π is $(2^c \epsilon, \epsilon)_{[2^{-s}, 1]}$-secure in $\mathcal{G}$.*

This second condition is a variant of Definition 6 where we implicitly pad all adversaries to have running time at least 2^{c-s}.

Note that when $s \geq c$ the second condition is equivalent to the notion of c-bit security. When $s < c$, the notion of (c, s)-bits of security is strictly weaker than the notion of c-bit security.

Lemma 10. *Let $c > 0$. Let $\mathcal{G}$ be an indistinguishability game, and Π a primitive that has c-bits of $\mathcal{G}$-security. Then for any $s < c$, there exists a primitive Π' and indisinguishability game $\mathcal{H}$ such that Π' has (c, s)-bits of $\mathcal{H}$-security, but not c-bits of $\mathcal{H}$-security.*

We next give an analogue of Theorem 2 in the setting of our relaxed notion of bit security.

Theorem 5. *Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be an FHE scheme with plaintext space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, where $\widetilde{\mathcal{M}}$ is a normed space with norm $\|\cdot\|$. Let $\mathsf{Estimate}$ be such that $\tilde{\Pi} = (\Pi, \mathsf{Estimate})$ is statically approximate. Let M_t be a ρ-KLDP mechanism. If Π is κ-bit $\mathsf{IND\text{-}CPA}$-secure, then $M[\tilde{\Pi}]$ has $(\kappa - \log_2 24, \log_2(1/\rho) - \log_2 q - \log_2 24)$-bits of q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security.*

Proof. We defer detail to the full version of the paper, and simply note that the proof is a (relatively standard) hybrid argument. The main novelty is the relaxed notion of security we target, which allows us to bound the *statistical* advantage term $q\rho$ (corresponding to "breaking" the KLDP mechanism) by a *computational* term (corresponding to breaking $\mathsf{IND\text{-}CPA}$ security of $\tilde{\Pi}$). □

Corollary 2. *Let $\sigma = \sqrt{24qn}2^{s/2}$. If CKKS is $(c + \log_2 24)$-bit $\mathsf{IND\text{-}CPA}$-secure, then $\mathsf{S\text{-}CKKS}_\sigma$ is (c, s)-bit q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-secure.*

Proof. This reduces to combining Lemma 6 with Theorem 5. The expression for ρ comes from the identity $s = \log_2(1/2\rho) - \log_2 q - \log_2 24$. The size of the Gaussian noise comes from $2^{-s}/48q = \rho = \mathsf{ct}.t^2/2\sigma^2$, which can be rewritten as $\sigma = \sqrt{24q}2^{s/2}\mathsf{ct}.t$. Note that the static estimate $\mathsf{ct}.t$ in CKKS is in the norm $\|\cdot\|_\infty^{\mathsf{can}}$, which we upper bound by $\sqrt{n}\,\|\cdot\|_2$ to get that $\sigma = \sqrt{24qn}2^{s/2}\mathsf{ct}.t$. Finally, we can apply Theorem 5 to achieve the result. □

Compare this result to Theorem 3, where the noise scales with the *computational* security parameter rather than the *statistical* security parameter. While choosing $s < c$ leads to a relaxed notion of security, this relaxation is precisely characterized. The non-trivial statistical attacks that we allow are

- simple to analyze (via results such as Lemma 5), and
- independent of any underlying hardware improvements (or other computational improvements, such as parallelization).

One can therefore justify a *much* smaller choice of s than the typical (computational) choice of $c = 128$. We do not suggest a particular choice for s, and instead give a variety of choices in Table 1. Note that the choice of s should be application dependent, as each time the protocol is instantiated[5] the adversary has a fresh chance to mount an attack of advantage up to 2^{-s}. For this reason, one should choose s much larger for protocols that will be instantiated many times. Note that by Lemma 9, for $s = c$ the notion of (c, s)-bits of security reduces to the notion of c-bits of security, so the top row of Table 1 equivalently states parameters to achieve 128-bits of q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security.

A Sample Instantiation. We briefly describe a concrete instantiation of our countermeasure that achieves $(128, 64)$-bits of q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security. Throughout, we let the number of supported decryption queries be $q = 2^{10}$. Note that one can always (later) support more decryption queries, by rekeying when one runs out. Parameterize CKKS to achieve 133-bits of $\mathsf{IND\text{-}CPA}$-security, where $133 > 128 + \log_2 24$. Let n be the resulting dimension of the chosen CKKS instance. We will assume $n \leq 2^{15}$, as every choice of parameters from the Homomorphic Encryption Standard [1] satisfies this bound.

Then, by Corollary 2, if $\sigma = \sqrt{24qn}2^{s/2}$, then $\mathsf{S\text{-}CKKS}_\sigma$ is (c, s)-bit q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-secure. In particular, this loses another $s/2 + \log_2 \sqrt{24qn}$ bits of precision compared to decrypting via returning $\mathsf{CKKS.Dec_{sk}}(\mathsf{ct}.c)$. The particular value of $s/2 + \log_2 \sqrt{24qn}$ can be found in Table 1 as the entry labeled $(s, q) = (64, 2^{10})$, which is 46.79. Therefore, adding an additional 46.79 bits of i.i.d. Gaussian noise suffices to achieve (128, 64)-bits of q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security.

4.5 Parameters for Concrete Countermeasures

As the attack in [19] was made publicly available, the major open-source implementations of the CKKS scheme adopted several different countermeasures. We briefly summarize these countermeasures in this subsection, and we propose concrete parameters for them to achieve the desired $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security.

[5] In our particular application, this includes things like re-keying, which one can do to "refresh" the number of decryptions one may release.

Table 1. Additional size of Gaussian noise (measured in bits) required by the countermeasure of Corollary 2 to achieve (c, s)-bits (Definition 19) of q-IND-$\mathsf{CPA}^{\mathsf{D}}$-security, where q is a bound on the number of decryption queries, and $n \leq 2^{15}$ is a bound on the ring dimension, chosen as it is the highest dimension parameter in the Homomorphic Encryption Standard [1]. This table assumes one samples Gaussians using the sampler of [4], see the full version of the paper for details.

$s\backslash q$	1	2^5	2^{10}	2^{15}
128	73.79	76.29	78.79	81.29
112	65.79	68.29	70.79	73.29
96	57.79	60.29	62.79	65.29
80	49.79	52.29	54.79	57.29
64	41.79	44.29	46.79	49.29
48	33.79	36.29	38.79	41.29
32	25.79	28.29	30.79	33.29

HElib. The decryption API implementation was modified to add pseudorandom Gaussian noise to the raw decryption result. By default, HElib implements S-CKKS_1, *e.g.* the size of the extra noise is equal to the size of the static error bound of the homomorphic computation. HElib also provides an optional precision parameter in its decryption API such that the extra noise is chosen to be the largest within the precision requirement (for example, if the static error bound is not tight). To achieve (c, s)-bit security against at most $q \geq 1$ decryption queries, this precision parameter should be calibrated such that sufficient (as quantified in Theorem 5 and Table 1) noise is added during decryption.

HEAAN, Lattigo. These libraries require the default decryption API to be used only by the secret key holder, and they added a specialized decryption API to share the decryption results publicly. In HEAAN, the new decryption API takes a noise size parameter, which sets the amount of Gaussian noises to be added to the raw decryption result. In Lattigo, the new decryption API takes a rounding parameter, which is used to round the raw decryption result to certain precision. For both of them, one must estimate the ciphertext error $\mathsf{ct}.t$ separately and set the noise parameter as in Theorem 5 and Table 1 to achieve (c, s)-bit security against q decryptions.

PALISADE. The decryption function in PALISADE also adds Gaussian noise to the raw decryption result, but the size of the noise is chosen (dynamically) in a way detailed in Sect. 5.

4.6 The Impact of Our Countermeasure

Evaluating the feasibility of our countermeasure for some application depends on both the required (application) precision, as well as the supported (library) precision. Provided the difference between these is larger than the sum of the DP noise (as measured in Table 1) with the approximation error, our countermeasure should be able to be instantiated.

32-bit applications. Concretely, many applications (say in machine learning) require 32 bits of precision. If a FHE library only supports computations with up to 64 bits of precision, this leaves at most 32 bits available for the sum of the CKKS approximation error and the DP error induced by our countermeasure. This means that at best, one will be able to choose $s \approx 32$, which is likely too low for most applications. Note that if the FHE library supports up to 128-bit precision computations[6], this problem disappears, as there are now ≈ 96 bits available for the sum of the errors, allowing the conservative choice of $s \approx 128$.

Low-precision applications. Some applications may solely require 8 or 16 bits of precision (see for example [16] or [33] for work on training ML models with low-precision computations). This leaves 48–56 bits of precision for the sum of the CKKS approximation error and the DP error. One can then choose $s \approx 64$ (16-bit required precision) or 80 (8-bit), where precise choices of s would depend on the size of the CKKS approximation error. We view either of these choices as much more reasonable than $s \approx 32$, although in all settings the particular choice of s that is appropriate is application-dependent.

5 Dynamic Error Estimation

Yuriy Polyakov [28] has recently suggested a technique to get sharper bounds on the ciphertext error of the CKKS scheme. Briefly, this is done via leveraging a special message encoding which fixes many of the coordinates of the original CKKS message space to be constantly 0. Provided one only evaluates functions which ignore these coordinates, upon decryption these coordinates will only contain the error incurred during the homomorphic computation, and one can attempt to generalize the (exact) error measurements within these coordinates to an estimate of the entirety of the error.

This notion differs from our notion of static approximate correctness in two significant ways, namely

- it depends on the particular ciphertext one is estimating the error of, *e.g.* can only be computed *dynamically* during the program "run-time", and
- it can only be computed during decryption, *e.g.* is not *publicly-computable* information about the ciphertext.

We investigate the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security of applying our transformation of Definition 15 to an approximate encryption scheme that is correct in the "dynamic" sense sketched above. In this slightly modified setting, we get significantly different results. For an $\mathsf{IND\text{-}CPA}$-secure, dynamic approximately correct FHE scheme $\tilde{\Pi}$, we find that $M[\tilde{\Pi}]$ is often insecure. Specifically, assuming a "non-triviality" condition on M that we define in Definition 23, we find that

[6] For example, Lattigo and PALISADE can both support computations of this precision.

1. for a "natural" class of $\mathsf{IND\text{-}CPA}$-secure $\tilde{\Pi}$ (including CKKS), $M[\tilde{\Pi}]$ is not q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ secure when one uses dynamic error estimation, and
2. there exists an $\mathsf{IND\text{-}CPA}$-secure $\tilde{\Pi}$ such that $M[\tilde{\Pi}]$ is not KR^{D}-secure (again, when one uses dynamic error estimation).

5.1 A (Heuristic) Dynamic Estimation Procedure for CKKS

We first provide a detailed description of Yuriy Polyakov's dynamic error estimation procedure for CKKS [28], which has been implemented in PALISADE [24]. We define a variant DE-CKKS of CKKS that is modified to use this dynamic error estimation technique. The message space of DE-CKKS is the set of real vectors $\mathbb{R}^{n/2}$, which is a subset of the message space $\mathbb{C}^{n/2}$ of CKKS. We use $\Re(z)$ and $\Im(z)$ to denote the real and imaginary parts of a complex number $z \in \mathbb{C}$, respectively. We now describe the modified scheme DE-CKKS.

- DE-CKKS.KeyGen: The parameter and key generation algorithms are identical to CKKS, except that the conjugation keys are not generated anymore.
- DE-CKKS.Encode: The encoding algorithm is the same as in CKKS, except that it takes only real vectors $\mathbf{x} \in \mathbb{R}^{n/2}$.
- DE-CKKS.Enc: The encryption algorithm is identical to CKKS.
- DE-CKKS.Eval: The homomorphic evaluation algorithm is also identical to CKKS, except that homomorphic conjugation operation is no longer supported.
- DE-CKKS.Dec: The modified decryption algorithm combines the decryption and decoding algorithms of CKKS, and it works as follows given the secret key sk and a ciphertext ct.
  1. Decrypt ct and then decode the vanilla CKKS decryption result: $\mathbf{z} = \mathsf{CKKS.Decode}(\mathsf{CKKS.Dec}_{\mathsf{sk}}(\mathsf{ct}))$. Note that $\mathbf{z} \in \mathbb{C}^{n/2}$ is a complex vector.
  2. Let $\mathbf{x} = \Re(\mathbf{z})$, and $\mathbf{e} = \Im(\mathbf{z})$. Estimate the standard deviation $\sigma_e = \mathrm{stdev}(\mathbf{e})$.
  3. Return $\mathbf{x} + \mathbf{r}$, where $\mathbf{r} \leftarrow \mathcal{N}(0, \sqrt{q+1} \cdot \sigma_e I_n)$ is a Gaussian noise vector.

In practice, since the canonical embedding is a scaled isometry with respect to the ℓ_2 norm, we can add the same amount of noise without decoding by first decrypting ct to obtain the ring element $m = \mathsf{CKKS.Dec}_{\mathsf{sk}}(\mathsf{ct})$, computing the ℓ_2 norm of $\frac{1}{2}(m(X) - m(1/X))$ to obtain $\sigma'_e = \sqrt{n} \cdot \sigma_e$, adding $n/2$ i.i.d. Gaussians of parameter $\sqrt{q+1} \cdot \sigma'_e$ to m' and then decoding the resulting noisy ring element.

The PALISADE development team has done some experiments to validate this dynamic error estimation method, and they claimed that it provides a good estimation [28]. With optimizations described in [17], they assumed that the rescaling error dominates the ciphertext error after each rescaling operation, and that such error can be reduced in size similar to the ciphertext error in fresh encryptions. Furthermore, they assumed the adversary is non-adaptive, meaning that the input messages do not depend on any decryption result. Their experiments encrypted two random real vectors, homomorphically evaluated their component-wise product followed by a rescaling operation, and then decrypted the resulting ciphertext and compared the estimated error size with the actual ciphertext error. The results showed that the dynamically error estimation is very close to the actual ciphertext error sizes: for example, they differ by at most 2 bits when the lattice dimension is $n = 2^{13}$.

5.2 Dynamic Estimation

We next introduce the notion of a dynamically approximately correct FHE scheme $\tilde{\Pi}$. Our notion of dynamic approximate correctness depends on solely the "run-time" values of the FHE scheme, namely the secret key sk, and the ciphertext ct one wishes to bound. These suffice to instantiate the dynamic estimation scheme described in Sect. 5.1. We omit the other values (such as individual ciphertext error bounds t_i, and the circuit C itself) for simplicity—there clearly cannot be a security benefit to this omission, as an adversary can easily record or compute these values.

Definition 21 (Dynamic Approximate Correctness). *Let Π be a FHE scheme with message space $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$, which is a normed space with norm $\|\cdot\| : \widetilde{\mathcal{M}} \to \mathbb{R}_{\geq 0}$. Let $\mathcal{L}$ be a space of evaluatable functions, and let* $\mathsf{Estimate} : \mathcal{SK} \times \mathcal{C} \to \mathbb{R}_{\geq 0}$ *be an efficiently computable function. We call the tuple of algorithms $\tilde{\Pi} = (\Pi, \mathsf{Estimate})$ a* dynamically approximately correct *FHE scheme if for all $m_1, \ldots, m_k \in \mathcal{M}$, for all $C \in \mathcal{L}$, for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$*, for all* $\mathsf{ct} \leftarrow \mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_k))$*, we have that*

$$\|\mathsf{Dec}_{\mathsf{sk}}(\mathsf{ct}) - C(m_1, \ldots, m_k)\| \leq \mathsf{Estimate}_{\mathsf{sk}}(\mathsf{ct}). \tag{10}$$

The above notion is a "perfect" notion of dynamic approximate correctness—there is an obvious statistical notion as well, where the desired inequality solely has to hold with high probability over all of the various sources of randomness. For simplicity of exposition we will work with the perfect notion.

We will view the notion of dynamic approximate correctness as a refinement of the notion of static approximate correctness. This can be done without loss of generality, as

- every known approximate FHE scheme is statically correct, and
- the minimum of two (correct) estimation functions is correct.

5.3 Attack Against $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-Security of $M[\tilde{\Pi}]$ for "Natural" Π

We next attack the $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security of $M[\tilde{\Pi}]$ for "natural" dynamically correct schemes $\tilde{\Pi}$. We briefly summarize the attack, as it is both "obvious", and establishing it theoretically requires a few new definitions (as it fails for "unnatural" schemes). If

- dynamic error estimation is able to tightly estimate the ciphertext error,
- the growth of ciphertext error during certain operations (such as multiplication) is dependent on the input to the operation, and

- the noise the KLDP mechanism M_t adds is dependent on t in a noticable way, then

an adversary which can distinguish the smaller KLDP noise can immediately break q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security. This is simply because one can use the aforementioned operation to construct two ciphertexts $\mathsf{ct}_0, \mathsf{ct}_1$ that encrypt the same value, but have drastically different ciphertext errors. Then, as the dynamic error estimation can detect this, the KLDP mechanism will add drastically different noise in the left and right worlds of the q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ game, immediately breaking security.

The attack is straightforward to implement, which we demonstrate in Sect. 5.4. We next theoretically establish the validity of the attack, by defining the aforementioned notions of "naturality".

Definition 22 (τ-Separated Noise Estimation). *Let $\tilde{\Pi}$ be a dynamically approximately correct FHE scheme with message space $\mathcal{M}$ and space of evaluatable functions $\mathcal{L}$. Let $\tau \geq 1$, and let $C \in \mathcal{L}$ be a circuit. For $m_0, m_1 \in \mathcal{M}$, let $t(m) = \mathsf{Estimate}_{\mathsf{sk}}(\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(m)))$. We say that C has τ-separated noise under $\tilde{\Pi}$ if there exists $m_0, m_1 \in \mathcal{M}$ such that $\tau t(m_0) = t(m_1)$ with non-negligible probability.*

The seemingly strong condition $t_1 = \tau t_0$ can be replaced by requiring that $|t_0 - \tau t_1|$ is small, and the mechanism M_t produces larger noise as t increases. For example, the Gaussian mechanism adds noise of variance $\sigma^2 = t^2/2\rho$, which increases monotonically with t.

Definition 23 (τ-Sensitivity). *Let M_t be a ρ-KLDP mechanism on a normed space $\mathcal{M}$, and let $\tau : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. We say that M_t is* τ-sensitive *at $m \in \mathcal{M}$ if for any $t \geq 1$, the distributions $M_t(m) \not\approx_c M_{t\tau(\rho)}(m)$ are computationally distinguishable.*

The trivial 0-KLDP mechanism (which ignores its input, and returns a fixed constant) is not τ-sensitive for any τ. Note that this condition is desirable in practice—if M_t is not τ-sensitive, there is no real point in getting sharper noise estimates.

Theorem 6. *Let $\tilde{\Pi}$ be an $\mathsf{IND\text{-}CPA}$-secure, dynamically approximately correct FHE scheme with message space $\mathcal{M}$ and space of evaluatable functions $\mathcal{L}$. Let $\tau : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and assume that M is a ρ-KLDP mechanism which is τ-sensitive at 0. Furthermore, assume there exist $m_0, m_1 \in \mathcal{M}$ and $C \in \mathcal{L}$ such that $C(m_0) = C(m_1) = 0$ and C has τ-separated noise estimation under $\tilde{\Pi}$ with respect to inputs m_0, m_1. Then $M[\tilde{\Pi}]$ is not $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-secure.*

Proof. We defer details (which is simply a formalization of the attack sketched at the beginning of the section) to the full version of the paper. □

While it is not clear how to extend this attack to an attack on KR^{D} security (as was present in [19]), the attack still leaks information correlated with $\|m\|$.

5.4 Breaking q-IND-CPA$^{\mathsf{D}}$-Security of PALISADE's Dynamic Error Estimation Countermeasure

We implemented the attack in Theorem 6 against the PALISADE's implementation of CKKS, which is currently the only known implementation of dynamic noise estimation. Our attack experiments use the exceedingly simple circuit $f(x_1, x_2) = x_1^2 - x_2$, as well as the circuit $g(x_0, \ldots, x_{4k-1}) = \sum_{i=0}^{2k-1}(x_i \cdot x_{2k+i})$ in Algorithm 4. Notice that both f and g evaluate to 0 on input $\mathbf{0}$. On the other hand, we chose several moderate values of $B > 0$ to set the input $\mathbf{m}$ such that $f(\mathbf{m}) = 0$ and $g(\mathbf{m}) = 0$:

- For f, let $m_1 = B$ and $m_2 = B^2$.
- For g, let $m_i = B$ for all $0 \leq i \leq 3k - 1$, and let $m_i = -B$ for all $3k \leq i \leq 4k - 1$.

Our attack homomorphically evaluates f (or g) on encryptions of both $\mathbf{0}$ and $\mathbf{m}$, then it decrypts the final ciphertexts to get z_0 and z_m. As expected, in all our experiments we see that $\|z_0\|_\infty$ and $\|z_m\|_\infty$ can be clearly distinguished. We summarize our experimental results in Table 2 with several parameter sets. We have made the source code of our experimental programs available.[7]

Table 2. The experimental results of applying the attack in Theorem 6 with circuits $f(x_1, x_2) = x_1^2 - x_2$ and $g(x_0, \ldots, x_{4k-1}) = \sum_{i=0}^{2k-1}(x_i \cdot x_{2k+i})$. For both $C \in \{f, g\}$, denote z_0 the decryption result of $\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(\mathbf{0}))$, and z_m the decryption result of $\mathsf{Eval}_{\mathsf{pk}}(C, \mathsf{Enc}_{\mathsf{pk}}(\mathbf{m}))$ for the input $\mathbf{m}$ as defined above with parameters B and k. We set the lattice parameters (n, Q) to achieve at least 128 bit IND-CPA security, and we choose several different values for the scaling factor Δ and the slots number. For each parameter set, we run the attack 100 times and report the average and standard deviation of $\|z_0\|_\infty$ and $\|z_m\|_\infty$. As shown in the last two columns, there are clear distinctions on the estimated noise sizes between ciphertexts evaluated on $\mathbf{0}$ and $\mathbf{m}$.

Circuit	$(n, \log Q)$	$\log \Delta$	B	k	#slots	$\|z_0\|_\infty$	$\|z_m\|_\infty$
f	$(2^{13}, 100)$	40	100	-	1	$2.19 \times 10^{-8} \pm 1.83 \times 10^{-8}$	$2.75 \times 10^{-6} \pm 2.19 \times 10^{-6}$
			100	-	1024	$1.07 \times 10^{-7} \pm 1.42 \times 10^{-8}$	$1.87 \times 10^{-5} \pm 2.54 \times 10^{-6}$
			32	-	1	$1.97 \times 10^{-8} \pm 1.52 \times 10^{-8}$	$1.06 \times 10^{-6} \pm 1.06 \times 10^{-6}$
			32	-	1024	$1.08 \times 10^{-7} \pm 1.54 \times 10^{-8}$	$6.08 \times 10^{-6} \pm 8.85 \times 10^{-7}$
g	$(2^{14}, 150)$	45	32	15	1	$1.08 \times 10^{-8} \pm 4.37 \times 10^{-9}$	$2.27 \times 10^{-7} \pm 1.95 \times 10^{-7}$
			32	15	1024	$1.08 \times 10^{-8} \pm 4.14 \times 10^{-9}$	$1.40 \times 10^{-6} \pm 2.02 \times 10^{-7}$
			16	50	1	$1.07 \times 10^{-8} \pm 4.45 \times 10^{-9}$	$2.00 \times 10^{-7} \pm 1.90 \times 10^{-7}$
			16	50	1024	$1.06 \times 10^{-8} \pm 4.67 \times 10^{-9}$	$1.27 \times 10^{-6} \pm 1.70 \times 10^{-7}$

5.5 Attack Against KR$^{\mathsf{D}}$-Security of $M[\tilde{\Pi}]$ for "Artificial" Π

We construct an (artificial) IND-CPA-secure, dynamically approximately correct FHE scheme $\tilde{\Pi}$ such that $M[\tilde{\Pi}]$ fails to be KR$^{\mathsf{D}}$-secure.

[7] https://github.com/ucsd-crypto/DynamicEstimationAttack.

Theorem 7. *There exists an $\mathsf{IND\text{-}CPA}$-secure, dynamically approximately correct FHE scheme $\tilde{\Pi}$ such that for any linear ρ-KLDP mechanism M that is τ-sensitive at 0, $M[\tilde{\Pi}]$ is not KR^{D}-secure.*

Proof. It suffices to (deterministically) add "noise" of the form $\tau(1, 1, \ldots, 1)$ to decryptions, where τ is (adaptively) chosen to leak bits of the secret key. We defer details to the full version.

6 Conclusion and Open Problems

In this work, we have shown that for CKKS with "static" error estimates, to obtain κ-bit $\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security

- it suffices to add $\kappa/2 + \tilde{O}(1)$ bits of noise (Theorem 3), and
- it is necessary to add $\kappa/4 - \tilde{\Omega}(1)$ bits of noise (Theorem 4).

Our results therefore somewhat tightly characterize the impact on the accuracy of CKKS instantiated with a natural countermeasure to the Li-Micciancio attack [19]—$\Theta(\kappa)$ additional bits of noise are both necessary and sufficient for security. Still, it is natural to wonder if the right scaling for our countermeasures is $\kappa/4$ or $\kappa/2$.

We additionally show how one can concretely obtain smaller noise via a relaxed notion of security (Theorem 5). In particular, we show that $s/2 + \tilde{O}(1)$ bits of additional noise suffice to achieve (c, s)-bits of q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$ security, where s can plausibly be set much less than 128.

We include discussion of the concrete overhead of our countermeasure in Sect. 4.6, where find that our countermeasure is easily implementable (for general purpose computation) provided the FHE library supports 128-bit precision computations, while FHE libraries that support 64-bit precision computations may only be able to instantiate our countermeasure for certain (low-precision) applications, or with aggressive parameterizations.

Both our work and the work of [19] investigate how the *correctness* of encryption can impact the underlying *security* one attains. As correctness analysis typically leverages (unproven) heuristics for tighter noise estimates, we view formally justifying these heuristics to be important going forward, as the false heuristics may lead to security issues.

While our results on "dynamic" error estimation are negative, we have not ruled out achieving some weaker security notion with these techniques (for natural schemes). Our attack of Theorem 6 shows that dynamic error estimation can leak the norm of the input to the computation. Can the leakage be *provably* limited to this information?

Finally, our work examines *black box* modifications one can make to CKKS to attain q-$\mathsf{IND\text{-}CPA}^{\mathsf{D}}$-security. It is plausible that a CKKS-specific construction could attain smaller parameters, say by randomizing homomorphic operations, choosing larger than typical scaling factors Δ, or carefully investigating the ciphertext error after bootstrapping.

References

1. Albrecht, M., et al.: Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, Toronto, Canada, November 2018. https://homomorphicencryption.org/standard/
2. Bai, S., Lepoint, T., Roux-Langlois, A., Sakzad, A., Stehlé, D., Steinfeld, R.: Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. J. Cryptol. **31**(2), 610–640 (2018)
3. Canonne, C.L.: A survey on distribution testing: your data is big. But is it blue? Theory of Computing, pp. 1–100 (2020)
4. Canonne, C.L., Kamath, G., Steinke, T.: The discrete gaussian for differential privacy. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems, vol. 33, pp. 15676–15688. Curran Associates Inc. (2020)
5. Cheon, J.H., et al.: Toward a secure drone system: flying with real-time homomorphic authenticated encryption. IEEE Access **6**, 24325–24339 (2018)
6. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 360–384. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_14
7. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A full RNS variant of approximate homomorphic encryption. In: Cid, C., Jacobson Jr., M. (eds.) SAC 2018. LNCS, vol. 11349, pp. 347–368. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-10970-7_16
8. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 409–437. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_15
9. Cheon, J.H., Kim, A., Yhee, D.: Multi-dimensional packing for HEAAN for approximate matrix arithmetics. IACR Cryptology ePrint Archive, 2018:1245 (2018)
10. Cheon, J.H., Kim, D., Kim, Y., Song, Y.: Ensemble method for privacy-preserving logistic regression based on homomorphic encryption. IEEE Access **6**, 46938–46948 (2018)
11. Devroye, L., Mehrabian, A., Reddad, T.: The total variation distance between high-dimensional Gaussians with the same mean. arXiv preprint arXiv:1810.08693 (2018)
12. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_49
13. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
14. Han, K., Hong, S., Cheon, J.H., Park, D.: Logistic regression on homomorphic encrypted data at scale. In: AAAI 2019, pp. 9466–9471. AAAI Press (2019)
15. HElib (release 2.2.0). https://github.com/homenc/HElib (2021). IBM
16. Kalamkar, D.D., et al.: A study of BFLOAT16 for deep learning training. arXiv preprint arXiv:1905.12322 (2019)
17. Kim, A., Papadimitriou, A., Polyakov, Y.: Approximate homomorphic encryption with reduced approximation error. In: Galbraith, S.D. (ed.) CT-RSA 2022. LNCS, vol. 13161, pp. 120–144. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95312-6_6

18. Lattigo 2.2.0. Online. http://github.com/ldsec/lattigo, July 2021. EPFL-LDS
19. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 648–677. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_23
20. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. J. ACM **60**(6), 43:1–43:35 (2013)
21. Micciancio, D., Walter, M.: Gaussian sampling over the integers: efficient, generic, constant-time. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 455–485. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_16
22. Micciancio, D., Walter, M.: On the bit security of cryptographic primitives. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 3–28. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_1
23. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), pp. 263–275 (2017)
24. PALISADE lattice cryptography library (release 1.11.6). https://gitlab.com/palisade/ (2022). PALISADE Project
25. Park, S., Lee, J., Cheon, J.H., Lee, J., Kim, J., Byun, J.: Security-preserving support vector machine with fully homomorphic encryption. In: SafeAI@AAAI 2019, CEUR Workshop Proceedings, vol. 2301 (2019). CEUR-WS.org
26. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: STOC, pp. 333–342. ACM (2009)
27. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: STOC, pp. 461–473. ACM (2017)
28. Polyakov, Y.: Personal communication, October 2020
29. Polyanskiy, Y., Wu, Y.: Lecture notes on information theory. Lecture Notes for ECE563 (UIUC) and 6(2012–2016):7 (2014)
30. Pöppelmann, T., Ducas, L., Güneysu, T.: Enhanced lattice-based signatures on reconfigurable hardware. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 353–370. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44709-3_20
31. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (2009)
32. Microsoft SEAL (release 3.6). https://github.com/Microsoft/SEAL, November 2020. Microsoft Research, Redmond, WA
33. Wang, N., Choi, J., Brand, D., Chen, C.-Y., Gopalakrishnan, K.: Training deep neural networks with 8-bit floating point numbers. In: Advances in Neural Information Processing Systems, vol. 31 (2018)