On the precision loss in approximate homomorphic encryption

Anamaria Costache[1], Benjamin R. Curtis[2], Erin Hales[3], Sean Murphy[3], Tabitha Ogilvie[3], and Rachel Player[3]

[1] Norwegian University of Science and Technology (NTNU), Norway
anamaria.costache@ntnu.no
[2] Zama, Paris, France
ben.curtis@zama.ai
[3] Royal Holloway, University of London, UK
{erin.hales.2018}, {tabitha.ogilvie.2019} @live.rhul.ac.uk
{s.murphy}, {rachel.player} @rhul.ac.uk

Abstract. Since its introduction at Asiacrypt 2017, the CKKS approximate homomorphic encryption scheme has become one of the most widely used and implemented homomorphic encryption schemes. Due to the approximate nature of the scheme, application developers using CKKS must ensure that the evaluation output is within a tolerable error of the corresponding cleartext computation. This is achieved by scaling the underlying raw data by an appropriate amount, known as the *scale parameter*, in order to preserve a certain amount of significant figures. Unfortunately, there is no clear guidance available for choosing an appropriate scale parameter, with a trial-and-error approach typically advised. In this work, we significantly improve the state-of-affairs and present the following main contributions. We give a comprehensive theoretical and experimental analysis of CKKS noise, that considers noise coming from the encoding and homomorphic evaluation operations separately. This enables us to give the first explicit definition for precision in the CKKS context. Additionally, we demonstrate the applicability of our analysis to determine convergence properties of iterative algorithms that are commonly used in applications.

1 Introduction

Homomorphic Encryption (HE) enables computation on ciphertexts without revealing any information about the underlying plaintexts. The problem of constructing a fully homomorphic encryption scheme was first posed in 1978 [42] and was not resolved until 2009, when Gentry [25] constructed the first such scheme based on ideal lattice problems. Since then, many homomorphic encryption schemes have been proposed [4, 5, 6, 14, 16, 22, 23], based on the security of the Learning with Errors (LWE) problem [41] and its variants.

One of the most popular schemes is the approximate homomorphic encryption scheme CKKS [14], which we describe in Section A of the Supplementary

Material. Ciphertexts in all homomorphic encryption schemes based on the LWE problem contain noise, which grows with each evaluation operation, and must be carefully controlled to ensure correct decryption. The main insight of [14] is that it may be tolerable for decryption to be approximate, for example in applications where we expect small errors to occur. This enables the CKKS scheme to natively support real-valued plaintexts, making it attractive for application settings such as privacy-preserving machine learning [3, 32, 39]. In contrast, other schemes based on Ring-LWE [37, 45], such as BGV [4] or BFV [23], are exact and thus have a finite plaintext space that data must be encoded into. CKKS has been extensively optimised [11, 12, 33] and is implemented in many prominent open-source homomorphic encryption libraries [28, 30, 40, 44].

Homomorphic encryption schemes involve many different parameters, and it can be a challenge to choose appropriate parameters that balance efficiency, security and noise growth. This is particularly true for the CKKS scheme, for two main reasons. Firstly, encoding and encryption noises must be considered together. Most prior works do not attempt to explicitly detangle encoding and encryption noise, or do so only heurisitically (e.g. as in HElib [30]).

Secondly, in CKKS, we have to track not only the level of ciphertexts (as in BFV and BGV), but we must also track the scaling factor. Works building on CKKS often rely on the precision of the scheme without clearly defining what precision means in this context, how it changes through homomorphic operations, or indeed how it may affect the underlying data. Instead, the scale parameter Δ is typically understood as the means through which the user can control the desired precision[4]. Unfortunately, there is no clear guidance for choosing Δ and a trial-and-error approach is usually advised[5].

Contributions: In this work, we significantly improve this state of affairs, by giving a comprehensive theoretical and experimental analysis of both encoding and encryption noises, as they grow through homomorphic evaluation operations. Our study of encoding provides theoretical support for the heuristics in [30]. Moreover, we present the first definition of precision loss in the context of CKKS homomorphic evaluation computation. In particular, we give theoretical bounds on the precision loss from encoding and decoding. Further, we confirm these with extensive experiments in the HEAAN library [29].

In more detail, CKKS encoding maps an element from the (complex) message space into an element in the (polynomial ring) plaintext space via a scaled restriction of the inverse canonical embedding. We investigate the relationship of distances between these two spaces. Our first main result is Theorem 1, which relates the size of an error in the plaintext space to the size of the induced error in the message space. We provide a proof that this bound is the best possible. Next, perhaps surprisingly, we show that bounding a decrypted and decoded message over the only the real part, rather than the whole embedding, provides no benefit for worst case analyses. Indeed, the worst case expansion factor in

[4] See e.g. `https://github.com/microsoft/SEAL/issues/251`

[5] See e.g. `https://ibm.github.io/fhe-toolkit-linux/html/helib/md__opt__i_b_m__f_h_e-distro__h_elib__c_k_k_s-security.html`

either the real or complex part of our message equals the worst case expansion factor of the entire embedding.

Next, we significantly improve the state of affairs in analysing the noise growth coming from encryption and homomorphic evaluation operations in CKKS. The previous approach for this is the worst-case canonical embedding norm; we use this as our baseline, refining it following [31]. Then, we introduce two new approaches for analysing noise growth and compare them against the refined canonical embedding norm approach. In particular, we first present a novel worst-case analysis in the ring, and then additionally present the first average-case noise analysis for CKKS, using a Central Limit Theorem (CLT) approach. We further refine all of our analyses by parameterising our worst-case bounds in terms of a failure probability, α, rather than a-priori fixing a one dimensional failure probability as in prior work [17, 18, 26].

Our main experimental results can be found in Tables 1 and 2, and then we present further experiments in Supplementary Material F. In the tables in the main body, we can see that in the plaintext space, the CLT method very tightly models the noise growth. The Worst-Case Ring (WCR) method follows closely, and finally the Canonical Embedding (CE) method results in the largest practical-to-heuristic gap, as identified in [18]. In the message space, we see that the CLT method models the noise almost perfectly, with the CE following closesly, and finally the WCR performs very badly in this setting, perhaps unsurprisingly. Our conclusions are therefore as follows. If we are in a setting where we can accept for our bounds to fail with a certain probability α, then our work shows that we should follow the CLT method. If we are looking in the plaintext space, and wish to be more conservative, we should follow the WCR method. Finally, if we interested in the precision loss in the message space and wish to be conservative, we may follow the CE method. Alternatively, one could always follow the CLT method and adjust the value of the failure probability α according to what is deemed acceptable. The tables below show that this work effectively closes the practical-to-heuristic gap in both the plaintext and the message spaces.

All of our experiments were run in HEAAN v1.0 [29]. The reason for this choice is that our work presents a methodology for the "basic" operations of CKKS. As such, we choose not to look at the most recent improvements of CKKS, although we note that our methods can be applied to any such improvement in a straightforward manner.

Our next contribution is to combine the noise analyses from encoding and homomorphic evaluation, developed in this work, to precisely define and quantify precision loss due to homomorphic evaluation over CKKS. We present the results of experiments in Table 2, which shows the CLT approach is able to accurately upper bound the precision loss for both complex and real computations. We extend our work on precision and define a precision tracker tool which allows users to track precision through various operations in a circuit. The tracker can also be converted to a *precision budget*, in analogue to noise budgets for BGV or

BFV [18, 44]. To the best of our knowledge, this is the first time that precision has been precisely defined, and theoretically justified, in the CKKS context.

Our final main contribution is to show how our analyses can be applied in the context of iterative algorithms. These are widely used in applications of homomorphic encryption, a prominent example being in CKKS bootstrapping [10, 11, 27]. We show how to use our noise analyses to determine the point at which the inherent CKKS precision loss begins to interfere with the accurate bits of the desired solution, and thus at which it is no longer useful to continue the iteration process. Our discussion focuses on the Newton-Raphson algorithm for division, and can serve as a template for determining convergence properties of other iterative algorithms.

Related work: Prior noise analyses for CKKS were presented in [11, 12, 14, 33, 35, 30]. Most of these works employ a worst-case analysis in the canonical embedding, in analogue to the line of work [17, 18, 26] for analysing noise growth in BGV [4] and BFV [23]. In particular, a bound on the noise of each ciphertext (in the canonical embedding) can be tracked and dynamically updated as the homomorphic evaluation is performed. This leads to a bound on the noise in the output ciphertext, which can be used to set parameters for correctness. These worst-case bounds are developed assuming that the random variable falls within a certain multiple of standard deviations (e.g. six [26] or ten [30]) from its mean. In contrast, our worst-case analyses are refined through the use of a tunable failure probability.

Lee *et al.* [35] use the signal-to-noise ratio, and proposes to track the variance of the errors, rather than an upper bound. We provide a theoretical justification for such an 'average-case' approach. Average-case noise analyses for other homomorphic encryption schemes were presented in [16, 38].

Our work provides a methodology for noise analysis of CKKS, and we focus on the original scheme [14] for concreteness and for wide applicability. However, our methods could be applied to other variants and optimisations of CKKS, such as those presented in [12, 33, 35]. In more detail, Cheon *et al.* [12] give an RNS variant of CKKS. Kim *et al.* [33] propose new variants that encode with Δ^2 and reverse the order of rescale and multiplication, to enable automation of the rescaling procedure and to reduce noise growth. Lee *et al.* [35] propose to reorder operations in CKKS which admits a reduction in the signal-to-noise ratio.

Recently, Li and Micciancio [36] showed a key recovery attack for CKKS that exploits its approximate nature. In particular, the noise of an output from decryption e depends on the secret key. Various countermeasures have since been deployed in commonly used homomorphic encryption libraries [13]. The Li-Micciancio attacks highlight the importance of a thorough noise analysis not simply from the point of view of accuracy, but more importantly, from the point of view of security, and is a major motivator for our work. We discuss this further in Section 8 and describe how, under certain conditions, an exact variant of CKKS can be developed, that could be deployed as a mitigation.

The CKKS precision trackers that we develop in this work could form part of an automatic parameter selection tool or FHE compiler, and thus feeds into this line of work [2, 20, 21].

Structure: In Section 2 we introduce relevant background material and notation. In Section 3 we study the precision loss coming from encoding and decoding in CKKS. In Section 4 we study the noise coming from the homomorphic evaluation operations in CKKS. In Section 5 we combine these analyses to give an analysis of precision and precision trackers for operations with respect to the three different noise analysis methods. In Section 6 we report on experimental results. In Section 7 we consider the application of our results to the Newton-Raphson method.

2 Preliminaries

Vectors are denoted in small bold font $\mathbf{z}$, and z_j refers to a vector's j^{th} element, indexing from zero. The notation $\lfloor \cdot \rceil$ is used for rounding to the nearest integer and $[\cdot]_q$ represents reduction modulo q. For $z = x + iy \in \mathbb{C}$, we denote by $\lceil z \rfloor := \lceil x \rfloor + i \lceil y \rfloor$ the rounding of both its real and imaginary components, and extend this componentwise to define the rounding $\lceil \mathbf{z} \rfloor$ of a complex vector $\mathbf{z} \in \mathbb{C}^{N/2}$. Unless otherwise stated, log will always mean $\log_2$.

In this work, we will consider several different norms. We denote the p-norm by $\|\cdot\|_p$ and the infinity norm by $\|\mathbf{z}\|_\infty$. We consider norms on a polynomial m both as a vector of its coefficients and under the canonical embedding, and denote these norms by $\|m\|$ and $\|m\|^{\mathsf{can}}$ respectively. We use $s \leftarrow D$ to denote sampling s according to the distribution D.

We use the notation $\mathrm{N}(\mu, \sigma^2)$ to refer to a univariate Normal distribution with mean μ and variance σ^2, and $\mathrm{N}(\boldsymbol{\mu}; \Sigma)$ to refer to an N-dimensional multivariate Normal distribution with N-dimensional mean vector $\boldsymbol{\mu}$ and $N \times N$ covariance matrix Σ. For a polynomial $Z(X) \in \mathbb{R}[X]/(X^N + 1)$, we will write $Z \sim \mathrm{N}(\boldsymbol{\mu}, \rho^2 I_N)$ to indicate that each coefficient of Z is independently and identically normally distributed, i.e., $Z_i \sim \mathrm{N}(\mu_i, \rho^2)$.

The CKKS scheme uses the canonical embedding to define an encoding from the message space $\mathbb{C}^{N/2}$ to the plaintext space $\mathbb{Z}[X]/(X^N + 1)$ in the following way: an isomorphism $\tau : \mathbb{R}[X]/(X^N + 1) \to \mathbb{C}^{N/2}$ can be defined via considering the canonical embedding restricted to $N/2$ of the $2N^{\text{th}}$ primitive roots and discarding conjugates. Encoding and decoding then use this map τ, as well as a precision parameter Δ, as follows:

$$\mathrm{Encode}(\mathbf{z}, \Delta) = \lceil \Delta \tau^{-1}(\mathbf{z}) \rfloor, \qquad \mathrm{Decode}(m, \Delta) = \frac{1}{\Delta} \tau(m),$$

where $\mathbf{z} \in \mathbb{C}^{N/2}$, $m \in \mathbb{Z}[X]/(X^N + 1)$ and $\lceil \cdot \rfloor$ is taken coefficient-wise.

3 Encoding

In this section, we give theoretical bounds on the precision loss from encoding and decoding. To understand precision loss due to encoding, as well as translate

noise bounds derived in the plaintext space to noise bounds in the message space, we investigate how distance measured in $\mathbb{R}[X]/(X^N+1)$ corresponds to distance measured in $\mathbb{C}^{N/2}$, when we move between the two via τ. If we measure using the 2-norm in both spaces, these two distances correspond exactly as here τ gives a scaled isometry with $\|\tau(m)\|_2 = \sqrt{\frac{N}{2}}\|m\|_2$. However, we will use the infinity norm in both spaces. We do this to support our Worst Case in the Ring analysis (see Section 4), which uses the infinity norm, as well as to more accurately characterise error due to encoding which, as a rounding loss, is best characterised in the infinity norm on polynomial coefficients. In addition, we believe the infinity norm in $\mathbb{C}^{N/2}$ is most appropriate as it corresponds to bounding slot-wise error. Moreover, essentially all prior work on bounding noise in homomorphic encryption schemes uses the infinity norm in the canonical embedding [14, 17, 18, 26]. We find that, in the worst case, there is an $O(N)$ expansion in the infinity norm under the map τ and unlike the 2-norm, there is no contraction under the map τ^{-1}.

3.1 Mapping Theory

Lemma 1 ([19]). *Let $m \in \mathbb{R}^N$. Then $\|m\|_\infty \leqslant \|m\|_\infty^{\mathsf{can}}$.*

This inequality is best possible in the sense that it is achieved: if $m = \tau^{-1}(\mathbf{z})$ and we let $z_k = B\zeta_k^j$ for $0 \leqslant k \leqslant \frac{N}{2}-1$, then we find $\|m\|_\infty^{\mathsf{can}} = \|\mathbf{z}\|_\infty = B = |m_j| = \|m\|_\infty$. In particular, there is no contraction as we move from $\mathbb{C}^{N/2}, \|\cdot\|_\infty$ to $\mathbb{R}^N, \|\cdot\|_\infty$ but there is an expansion as we move the other way. The prior result on this bound is as follows.

Lemma 2 ([19, 26]). *Let $m \in \mathbb{R}^N$. Then $\|m\|_\infty^{\mathsf{can}} \leqslant N\|m\|_\infty$.*

Using generic proof methods and properties of the norm, we can shave this factor to $N/\sqrt{2}$.

We improve this result further. Before doing so, we require some definitions and a Lemma. The proof technique of the following Lemmas 3 and 4 is adapted from [8]. We introduce the notation $I(N,j)$ and $I(N)$ as follows:

$$I(N,j) := \sum_{k=0}^{N-1} \left|\sin\left(\frac{jk\pi}{N}\right)\right|,$$

$$I(N) := \max_{0\leqslant j\leqslant N-1} I(N,2j+1).$$

Lemma 3. *For $j \in \mathbb{Z}$, we have that $I(N,2j+1) = I(N,1)$, so that $I(N) = I(N,1)$.*

Proof. $2j+1 \in \mathbb{Z}_N^\times$, so $\{(2j+1)k \mod N : k = 0,...,N-1\} = \{k \mod N : k = 0,...,N-1\}$. Therefore

$$I(N,2j+1) = \sum_{\substack{x=\frac{(2j+1)k}{N},\\ 0\leqslant k\leqslant N-1}} |\sin(x\pi)| = \sum_{\substack{x=\frac{k}{N},\\ 0\leqslant k\leqslant N-1}} |\sin(x\pi)| = I(N,1).$$

Here, the central equality follows from the π-periodicity of $|\sin(\cdot)|$. $\square$

Lemma 4. $\lim_{N\to\infty} \frac{1}{N} I(N) = \frac{2}{\pi}$, *and this limit is approached from below.*

Proof. We have that $\frac{1}{N} I(N) = \frac{1}{N} I(N,1) = \frac{1}{N} \sum_{k=0}^{N-1} \left|\sin\left(\frac{k\pi}{N}\right)\right|$, while using Riemann sums we get:

$$\int_0^\pi |\sin(x)| = \lim_{N\to\infty} \frac{\pi}{N} \sum_{k=0}^{N-1} \left|\sin\left(\frac{k\pi}{N}\right)\right| = \pi \lim_{N\to\infty} \frac{1}{N} I(N).$$

As the LHS is equal to 2, we get the claimed limit. Moreover, as $|\sin(\cdot)|$ is concave, we get this sequence of Riemann sums is increasing. $\square$

Theorem 1. *Let $m \in \mathbb{R}^N$. Then $\|m\|_\infty^{\mathsf{can}} \leqslant \sqrt{I(N)^2+1}\, \|m\|_\infty$, and this bound is best possible for N fixed.*

Proof. Fix $\|m\|_\infty \leqslant 1$, and consider optimising $\|m\|_\infty^{\mathsf{can}}$ subject to this constraint. Since we can rotate the entries of $\sigma(m) \in \mathbb{C}^{N/2}$ via performing automorphisms on m, we may without loss of generality consider maximising $|m(\zeta)|$, where $\zeta = \exp(\frac{\pi i}{N})$.

Let $M(X) = \sum_{k=0}^{N-1} X^j$. We will show that, if $m(X)$ achieves a maximum of $|m(\zeta)|$ with $\|m\|_\infty \leqslant 1$, then $m(X) = X^k M(X)$ for some $k \in \mathbb{Z}_{2N}$. This is sufficient to prove our result as $\left\|X^k M(X)\right\|_\infty^{\mathsf{can}} = \sqrt{I(N)^2+1}$.

So, fix some maximising polynomial $m(X)$ with $\|m\|_\infty \leqslant 1$, and let $k \in \mathbb{Z}_{2N}$ be such that $k = \arg\max_j \mathrm{Im}(\zeta^{-j} m(\zeta))$. If $m(X) = X^k M(X)$, we are done – otherwise we will derive a contradiction to either the maximality of k or the maximality of m. By negating m as necessary, we say $\mathrm{Im}(\zeta^{-k} m(\zeta)) \geqslant 0$. In fact, we can say $\mathrm{Im}(\zeta^{-k} m(\zeta)) \geqslant 1$, since otherwise the maximality of m would give $\mathrm{Re}(\zeta^{-k} m(\zeta))^2 > \mathrm{Im}(\zeta^{-k} m(\zeta))^2$, contradicting the maximality of k via either $k + N/2$ or $k - N/2$.

The polynomial $M(X)$ (up to sign, uniquely) maximises the imaginary component $|\mathrm{Im}(M(\zeta))|$. We therefore have $\left|\mathrm{Im}(\zeta^{-k} m(\zeta))\right| \leqslant |\mathrm{Im}(M(\zeta))|$. Comparing $|m(\zeta)|$ and $|M(\zeta)|$ we find

$$\begin{aligned} |m(\zeta)|^2 - |M(\zeta)|^2 &= \left|\zeta^{-k} m(\zeta)\right|^2 - \mathrm{Im}(M(\zeta))^2 - \mathrm{Re}(M(\zeta))^2 && (1)\\ &= \underbrace{\mathrm{Im}(\zeta^{-k} m(\zeta))^2 - \mathrm{Im}(M(\zeta))^2}_{(1)} + \underbrace{\mathrm{Re}(\zeta^{-k} m(\zeta))^2 - 1}_{(2)}. && (2) \end{aligned}$$

As discussed, we certainly have $(1) \leqslant 0$. We will show that we also have $(2) \leqslant 0$, since otherwise k is not the maximal choice.

Suppose $\mathrm{Re}(\zeta^{-k} m(\zeta)) > 1$, so that $\sum_{j=0}^{N-1} m_j \cos\left(\frac{(j-k)\pi}{N}\right) > 1$. Then:

$$\mathrm{Im}(\zeta^{-(k-1)} m(\zeta)) - \mathrm{Im}(\zeta^{-k} m(\zeta)) = \sum_{j=0}^{N-1} m_j \left(\sin\left(\frac{(j-k+1)\pi}{N}\right) - \sin\left(\frac{(j-k)\pi}{N}\right)\right)$$

$$
\begin{aligned}
&= \left(\cos\left(\frac{\pi}{N}\right) - 1\right) \mathrm{Im}(\zeta^{-k} m(\zeta)) + \sin\left(\frac{\pi}{N}\right) \mathrm{Re}(\zeta^{-k} m(\zeta)) \\
&> \cos\left(\frac{\pi}{N}\right) + \sin\left(\frac{\pi}{N}\right) - 1 \\
&\geqslant 0,
\end{aligned}
$$

since $N \geqslant 2$. We can derive a similar contradiction using $\zeta^{-(k+1)} m(\zeta)$ in the case $\mathrm{Re}(\zeta^{-k} m(\zeta)) < -1$.

We have therefore shown $|m(\zeta)|^2 \leqslant |M(\zeta)|^2$, with equality if and only if there exists a $k \in \mathbb{Z}_{2N}$ with $m(X) = X^k M(X)$. The maximum for $|m(\zeta)|$ is therefore given by $|M(\zeta)| = \sqrt{I(N)^2 + 1}$ as claimed. □

Corollary 1. *Suppose for all $m \in \mathbb{R}^N$ we have $\|m\|_\infty^{\mathsf{can}} \leqslant N \cdot M(N) \|m\|_\infty$ with $M(N)$ a least upper bound. Then $M(N) \to \frac{2}{\pi}$ as $N \to \infty$.*

Proof. Immediate from Lemma 4 and Theorem 1. □

We will additionally be interested in bounding just the real component of the canonical embedding of m, although the following results apply equally to the imaginary component. We use the notation $\|m\|_\infty^{\mathsf{can,Re}} = \max_j |\mathrm{Re}(m(\zeta_j))|$.

We find that, in the limit, the upper bound on expansion of just the real component equals the upper bound on the entire expansion:

Lemma 5. *Let $m \in \mathbb{R}^N$. Then $\|m\|_\infty^{\mathsf{can,Re}} \leqslant I(N) \|m\|_\infty$, and this result is best possible.*

Proof. This proof technique is again adapted from [8].

Consider the j^{th} component of the canonical embedding z_j, given by evaluating $m(X)$ at the j^{th} primitive $2N$ root of unity. Then the real part of z_j is given by:

$$
\mathrm{Re}(z_j) = \sum_{k=0}^{N-1} m_k \mathrm{Re}(\zeta_j^k).
$$

We maximise the magnitude of this quantity, subject to $\|m\|_\infty = B$, by setting each $m_k = B \cdot \mathrm{Sign}(\mathrm{Re}(\zeta_j^k))$. We therefore have:

$$
\begin{aligned}
\|m\|_\infty^{\mathsf{can,Re}} &\leqslant B \max_{j=0,\ldots,N-1} \sum_{k=0}^{N-1} \left|\cos\left(\frac{k(2j+1)\pi}{N}\right)\right| \\
&= I(N) \|m\|_\infty .
\end{aligned}
$$

Corollary 2. *Let $m \in \mathbb{R}^N$. Then if for all N we have that $\|m\|_\infty^{\mathsf{can,Re}} \leqslant kN \|m\|_\infty$ then $k \leqslant \frac{2}{\pi}$ and $k \to \frac{2}{\pi}$ as $N \to \infty$.*

Proof. Immediate from Lemma 4 and Lemma 5. □

3.2 Application to Encoding

In this section, we apply the results from Section 3.1 to produce bounds on the growth of polynomials under encoding and decoding. Our first result enables us to produce bounds in the plaintext space given bounds in the message space.

Lemma 6. *Suppose* $m \in \mathbb{R}^N$ *and* $\mathbf{z} \in \mathbb{C}^{N/2}$ *are such that* $m = \text{Encode}(\mathbf{z}, \Delta)$. *Then* $\|m\|_\infty \leqslant \Delta \|\mathbf{z}\|_\infty + \frac{1}{2}$.

Proof. Immediate from Lemma 1 and that encoding rounds coefficient wise. □

The result in Theorem 1 enables us to give bounds in the message space given bounds in the plaintext space.

Lemma 7. *Suppose* $m \in \mathbb{R}^N$ *has* $\|m\|_\infty \leqslant B$. *Then if* $z = \text{Decode}(m, \Delta)$ *we have that* $\|\mathbf{z}\|_\infty \leqslant \frac{\sqrt{I(N)^2+1}}{\Delta} B$, *and this bound is best possible.*

Due to fast convergence of $I(N)$ to $\frac{2N}{\pi}$, in what follows we will replace this result by its limiting value, $\frac{2NB}{\pi\Delta}$. We can therefore precisely bound the error introduced during encoding.

Corollary 3. *Suppose* $\mathbf{z} \in \mathbb{C}^{N/2}$ *is encoded under scale factor* Δ. *Then the precision lost on each slot as a result of encoding is bounded by* $\frac{\sqrt{I(N)^2+1}}{2\Delta}$, *and this bound tends to* $\frac{N}{\pi\Delta}$ *as* $N \to \infty$.

Proof. Immediate from Lemma 7 and the fact that the encoding error polynomial has coefficients in $[-\frac{1}{2}, \frac{1}{2}]$. □

We can also give analogous results for the real and imaginary components alone.

Lemma 8. *Suppose* $m \in \mathbb{R}^N$ *has* $\|m\|_\infty \leqslant B$. *Then if* $\mathbf{z} = \text{Decode}(m, \Delta)$ *we have that* $\|Re(\mathbf{z})\|_\infty, \|Im(\mathbf{z})\|_\infty \leqslant \frac{2N}{\pi\Delta} B$, *and this bound is best possible.*

Corollary 4. *Suppose* $\mathbf{z} \in \mathbb{C}^{N/2}$ *is encoded under scale factor* Δ. *Then both the precision lost on the real and imaginary components of each slot is bounded by* $\frac{N}{\pi\Delta}$.

A surprising result of our investigations is that, if using a worst case analysis, it is not possible to achieve tighter analysis of precision loss by considering only the error on the real part of the message. This is due to the worst case expansion factors being the same in both cases. To benefit from restricting our attention to only the real part, we must be able to specify statistical, rather than worst case, behaviour, as explained in Section 5.

We note that the CKKS encoding/decoding process breaks the isometry τ by rounding to integral polynomials only. In particular, this means that we cannot decode to the whole of $\mathbb{C}^{N/2}$ but rather only to points on the lattice $\tau(R)$. We call this lattice the *lattice of preserved vectors*, and vectors in $\mathbb{C}^{N/2}$ on the lattice such that $\mathbf{z} = \text{Decode}(\text{Encode}(\mathbf{z}))$ *preserved*. More information is given in Supplementary Material Section B.

4 Noise

In this section, we consider how to bound the noise from encryption. We calculate worst-case bounds on noise in the ring and in the canonical embedding, as well as an average-case bound using Central Limit Theorem (CLT) techniques of [38] to obtain much tighter bounds. We begin by introducing new notation and methodology. We next give distributional results on the noise after certain important operations, before giving noise bounds for each operation following each of the three noise analysis methods.

4.1 Noise Analysis Method

Worst-case canonical embedding analysis. This is the standard way of viewing noise bounds used in previous works [14], improved by following the Iliashenko method [31]. In this method we track the size of error polynomials under the canonical embedding, $\|\epsilon\|_\infty^{\mathsf{can}}$, as follows. First, we derive the coefficient variance of the noise polynomial in the ring. Then we use Lemma 9 or triangle inequalities to derive bounds in the canonical embedding. We use the fact that $\|p(X)q(X)\|_\infty^{\mathsf{can}} \leqslant \|p(X)\|_\infty^{\mathsf{can}} \|q(X)\|_\infty^{\mathsf{can}}$ is the worst case bound on a product of polynomials.

Worst-case analysis in the ring. In this method, we track a bound on the size of the largest coefficient of the noise in the ring, $\|\epsilon\|_\infty$. We then track how the coefficients of ϵ grow in the worst case with each operation, using triangle inequalities and Gaussian tail bounds as indicated in Lemma 9. We use the fact that $\|p(X)q(X)\|_\infty \leqslant N \|p(X)\|_\infty \|q(X)\|_\infty$ is the worst case bound on a product.

Average-case analysis using CLT techniques. In this method, we develop a Central Limit argument that sees coefficients of the noise polynomial as sums of independent random variables, so their distribution can be approximated as a Normal distribution. This enables us to present an average-case approach that traces through the variance of the noise after each operation. Using Lemma 9 we can then derive a bound on the noise in the output ciphertext based on its variance.

Variance. We will use the variances of the followings distributions. For a polynomial f with coefficients distributed uniformly in $[-k/2, k/2]$, the variance of the coefficients is $\rho_f^2 = k^2/12$. For a polynomial s with coefficients drawn from the uniform ternary distribution $\{-1, 0, 1\}$ the variance of the coefficients is $\rho_s^2 = 2/3$.

Bounding failure probability. Given a (multivariate) random variable, we wish to identify a reasonable upper bound on the size of the components of the random variable(s). It has been common practice [26, 17, 18] to give an upper bound using the fact that $\texttt{erfc}(6) \approx 2^{-55}$. Instead of deferring to such a bound in all contexts, we express our worst case bounds on distributions in terms of a new failure probability parameter α, defined as follows.

Definition 1. *Suppose a random variable Z has real support. We will say B* is a probability $1 - \alpha$ bound *on Z if $\Pr(Z > B) = \alpha$. Equivalently, we will say B has* failure probability α*, or that B has* error tolerance α*.*

We therefore have the following Lemma, which gives probabilistic bounds on relevant random variables.

Lemma 9. *Suppose $Z \sim N(\mathbf{0}, \rho^2 I_N)$. Then:*

1. *A probability $(1-\alpha)$ bound on the random variable $\|Z\|_\infty$ is given by*

$$B = \sqrt{2}\ \rho\ \mathtt{erf}^{-1}((1-\alpha)^{\frac{1}{N}}).$$

2. *Let τ denote the map used in encoding and decoding and consider $\tau(Z)$. Then we have that $Re(\tau(Z)), Im(\tau(Z)) \sim N(\mathbf{0}, \frac{N}{2}\rho^2 I_{N/2})$, and a probability $(1-\alpha)$ bound on both $\|Re(\tau(Z))\|_\infty$ and $\|Im(\tau(Z))\|_\infty$ is given by*

$$B = \sqrt{N}\ \rho\ \ \mathtt{erf}^{-1}((1-\alpha)^{\frac{2}{N}}).$$

3. *A probability $(1-\alpha)$ bound on the random variable $\|Z\|_\infty^{\mathsf{can}}$ is given by*

$$B = \sqrt{N}\rho\ (-\log(1-(1-\alpha)^{\frac{2}{N}}))^{\frac{1}{2}}.$$

Proof. A proof is given in Supplementary Material Section C. □

When applying Lemma 9 in real and complex settings respectively, we use the following functions for notational convenience:

$$H_{\mathbb{R}}(\alpha, N) := \mathtt{erf}^{-1}((1-\alpha)^{\frac{1}{N}}) \ \text{ and } \ H_{\mathbb{C}}(\alpha, N) := (-\log(1-(1-\alpha)^{\frac{2}{N}}))^{\frac{1}{2}}\,.$$

4.2 Distribution of Error Polynomials

In this section, we derive the distribution of the noise polynomial of a ciphertext that is output from various homomorphic operations. In more detail, for a ciphertext $(\mathtt{ct}_0, \mathtt{ct}_1)$ encrypting a message m, we say its noise is ϵ, where $\mathtt{ct}_0 + \mathtt{ct}_1 s = m + \epsilon$. We denote by ρ^2 the (component) variance of a noise polynomial ϵ. Some operations, such as key switching, introduce an additive noise term, whose variance we denote by η^2, so that (for example) a ciphertext with noise variance ρ^2 that is input to key switching will have an output noise variance of $\rho_{\mathsf{ks}}^2 = \rho^2 + \eta_{\mathsf{ks}}^2$.

Fresh. First, we consider the encryption noise of a fresh ciphertext. The fresh encryption noise polynomial is given by $\epsilon_{\mathsf{fresh}} = ve + e_0 + e_1 s$. The Ring-LWE error terms e, e_0, e_1 are sampled from a discrete Gaussian with standard deviation σ, and s, v are the secret and ephemeral secret respectively, sampled from a uniform ternary distribution.

Heuristic 1 *The noise random variable $\epsilon_{\mathsf{fresh}}$ for a fresh ciphertext has a Normal distribution given by $\epsilon_{\mathsf{fresh}} \sim N(\mathbf{0}; \rho_{\mathsf{fresh}}^2 I_N)$ with variance*

$$\rho_{\mathsf{fresh}}^2 = (\tfrac{4}{3}N + 1)\sigma^2.$$

Proof. A proof is given in Supplementary Material Section D. □

Round. Component-wise rounding of a ciphertext $(\mathtt{ct}_0, \mathtt{ct}_1) \in (\mathbb{R}[X]/(X^N+1))^2$ in $\mathcal{R}_q$ introduces an additive rounding error $\epsilon_{\text{round}} = \tau_0 + \tau_1 s$, where τ_i is the rounding error on $\mathtt{ct}_i$. We model the τ_i as being drawn uniformly at random with coefficients in $[-\frac{1}{2}, \frac{1}{2}]$, leading to the following lemma.

Heuristic 2 *$\epsilon_{round} \sim N(\mathbf{0}, \eta^2_{round} I_N)$, where $\eta^2_{round} = \frac{N}{18} + \frac{1}{12}$.*

Proof. A proof is given in Supplementary Material Section D. □

Keyswitch. The keyswitch operation introduces an additive error $\epsilon_{\mathsf{ks}} = P^{-1} q_l \cdot d_2 e' + \epsilon_{\mathsf{round}}$, where e' is the Ring-LWE noise term in the evaluation key, $d_2 = [\mathtt{ct}_0[1]\mathtt{ct}_1[1]]_{q_\ell}$ is a component from the output of pre-multiply, and ϵ_{round} is a possible rounding error created by dividing by P^{-1}. The rounding error only applies if P does not divide q_ℓ, which is uncommon in implementations [44].

Heuristic 3 *$\epsilon_{ks} \sim N(\mathbf{0}, \eta^2_{ks} I_N)$, where $\eta^2_{ks} = \frac{1}{12} P^{-2} N q_\ell^2 \sigma^2 + \mathbb{1}_{P \nmid q_\ell} \left(\frac{N}{18} + \frac{1}{12}\right)$.*

Proof. A proof is given in Supplementary Material Section D. □

4.3 Worst-case noise bounds

In this section, we use Lemma 9 and the variances calculated in Section 4.2 to give worst-case bounds on the noise ϵ in a ciphertext obtained after each homomorphic evaluation operation. We give probabilistic bounds, defined by the parameter α, both in the ring (i.e., bounding $\|\epsilon\|_\infty$), and in the canonical embedding (i.e., bounding $\|\epsilon\|^{\mathsf{can}}_\infty$).

Heuristic 4 *Let $\mathtt{ct}$ be a fresh ciphertext encrypting a message m with noise ϵ. In the canonical embedding, with probability $1-\alpha$, we have that $\|\epsilon\|^{\mathsf{can}}_\infty \leq B_{fresh}$, where*

$$B_{fresh} = \sqrt{N \rho^2_{fresh}}\, H_{\mathbb{C}}(\alpha, N) = \sigma \sqrt{\tfrac{4}{3}N^2 + N}\, H_{\mathbb{C}}(\alpha, N).$$

In the ring, with probability $1-\alpha$, we have that $\|\epsilon\|_\infty \leq B_{fresh}$, where

$$B_{fresh} = \sqrt{2\rho^2_{fresh}}\, H_{\mathbb{R}}(\alpha, N) = \sigma \sqrt{\tfrac{8}{3}N + 2}\, H_{\mathbb{R}}(\alpha, N).$$

Proof. Heuristic 1 shows that the encryption noise has a multivariate Normal $\mathrm{N}(0; \rho^2_{\mathsf{fresh}} I_N)$ distribution, where $\rho^2_{\mathsf{fresh}} = (\frac{4}{3}N + 1)\sigma^2$. The worst case bounds on the encryption noise under the canonical embedding norm and in the ring then follow from Lemma 9 (1) and Lemma 9 (3) respectively. □

Lemma 10. *Let $\mathtt{ct}_{add}$ with noise ϵ_{add} be the ciphertext resulting from the homomorphic addition of two ciphertexts $\mathtt{ct}_1$ and $\mathtt{ct}_2$. Let $\mathtt{ct}_1$ and $\mathtt{ct}_2$ have noises ϵ_1 and ϵ_2, where $\|\epsilon_1\|^{\mathsf{can}}_\infty \leq B_1$ and $\|\epsilon_2\|^{\mathsf{can}}_\infty \leq B_2$ (respectively, $\|\epsilon_1\|_\infty \leq B_1$ and $\|\epsilon_2\|_\infty \leq B_2$). Then we can bound $\|\epsilon_{add}\|^{\mathsf{can}}_\infty \leq B_{add}$ (respectively $\|\epsilon_{add}\|_\infty \leq B_{add}$) where:*

$$B_{add} = B_1 + B_2\,.$$

Heuristic 5 *Consider rounding a non-integer ciphertext. In the canonical embedding, a worst-case bound on the additive noise introduced by this process, that holds with probability* $1-\alpha$*, is given by*

$$B_{\mathsf{round}} = \sqrt{N\eta_{\mathsf{round}}^2}\, H_{\mathbb{C}}(\alpha, N) = \sqrt{\tfrac{1}{18}N^2 + \tfrac{1}{12}N}\, H_{\mathbb{C}}(\alpha, N).$$

In the ring, a worst-case bound on the additive noise introduced by this process, that holds with probability $1-\alpha$*, is given by*

$$B_{\mathsf{round}} = \sqrt{2\eta_{\mathsf{round}}^2}\, H_{\mathbb{R}}(\alpha, N) = \sqrt{\tfrac{1}{9}N + \tfrac{1}{6}}\, H_{\mathbb{R}}(\alpha, N).$$

Heuristic 6 *Let* $\mathtt{ct}_{\mathsf{mult}}$ *be the ciphertext resulting from the pre-multiply operation applied on two ciphertexts* $\mathtt{ct}_1$ *and* $\mathtt{ct}_2$ *encrypting* m_1 *and* m_2 *respectively. Then* $\mathtt{ct}_{mult}$ *is an encryption of* $m_1 \cdot m_2$ *with noise* $\epsilon_{\mathsf{mult}} = m_1\epsilon_2 + m_2\epsilon_1 + \epsilon_1\epsilon_2$*. Let* $\mathtt{ct}_1$ *and* $\mathtt{ct}_2$ *have noises* ϵ_1 *and* ϵ_2*, where* $\|\epsilon_1\|_\infty^{\mathsf{can}} \leq B_1$ *and* $\|\epsilon_2\|_\infty^{\mathsf{can}} \leq B_2$ *(respectively,* $\|\epsilon_1\|_\infty \leq B_1$ *and* $\|\epsilon_2\|_\infty \leq B_2$*). In the canonical embedding, we have that* $\|\epsilon_{\mathsf{mult}}\|_\infty^{\mathsf{can}} \leq B_{\mathsf{mult}}$*, where*

$$B_{\mathsf{mult}} = \left(\|m_1\|_\infty^{\mathsf{can}} B_2 + \|m_2\|_\infty^{\mathsf{can}} B_1 + B_1B_2\right).$$

In the ring, we have that $\|\epsilon_{\mathsf{mult}}\|_\infty \leq B_{\mathsf{mult}}$*, where*

$$B_{\mathsf{mult}} = N \cdot \left(\|m_1\|_\infty B_2 + \|m_2\|_\infty B_1 + B_1B_2\right).$$

Heuristic 7 *Let* $\mathtt{ct}_{\mathsf{mult}}$ *be the ciphertext resulting from the key switch operation applied on the ciphertext* $\mathtt{ct}$*. This operation introduces an additive error term* ϵ_{ks} *that can be bounded as follows. In the canonical embedding, with probability* $1-\alpha$*, we have that* $\|\epsilon_{\mathsf{ks}}\|_\infty^{\mathsf{can}} \leq B_{\mathsf{ks}}$*, where*

$$\begin{aligned} B_{\mathsf{ks}} &= \sqrt{N\eta_{\mathsf{ks}}^2}\, H_{\mathbb{C}}(\alpha, N) \\ &= \sqrt{N\left(\tfrac{1}{12}P^{-2}Nq_\ell^2\sigma^2 + \mathbb{1}_{P\nmid q_\ell}\left(\tfrac{1}{18}N + \tfrac{1}{12}\right)\right)} H_{\mathbb{C}}(\alpha, N). \end{aligned}$$

In the ring, with probability $1-\alpha$*, we have that* $\|\epsilon_{\mathsf{ks}}\|_\infty \leq B_{\mathsf{ks}}$*, where*

$$\begin{aligned} B_{\mathsf{ks}} &= \sqrt{2\eta_{\mathsf{ks}}^2} H_{\mathbb{R}}(\alpha, N) \\ &= \sqrt{\tfrac{1}{6}\left(P^{-2}Nq_\ell^2\sigma^2 + \mathbb{1}_{P\nmid q_\ell}\left(\tfrac{2}{3}N + 1\right)\right)} H_{\mathbb{R}}(\alpha, N). \end{aligned}$$

Rescale: Let $\mathtt{ct}_{\mathsf{rs}}$ be the ciphertext resulting from the rescale operation applied to a ciphertext $\mathtt{ct}$ encrypting m with noise ϵ. Then $\mathtt{ct}_{\mathsf{rs}}$ encrypts m/Δ with noise $\epsilon_{\mathsf{rs}} = \epsilon/\Delta + \epsilon_{\mathsf{round}}$. Thus, if $\|\epsilon\|_\infty^{\mathsf{can}} \leq B$ (respectively, if $\|\epsilon\|_\infty \leq B$), then $\|\epsilon_{\mathsf{rs}}\|_\infty^{\mathsf{can}} \leq B_{\mathsf{rs}}$ (respectively, $\|\epsilon_{\mathsf{rs}}\|_\infty \leq B_{\mathsf{rs}}$), where

$$B_{\mathsf{rs}} = \Delta^{-1}B + B_{\mathsf{round}}.$$

Multiplication: We finally give a bound on the noise in a ciphertext resulting from homomorphic multiplication, i.e., the combination of pre-multiply, key switch and rescale.

Heuristic 8 *In either the ring or the canonical embedding, the worst-case bound on the noise in a ciphertext resulting from homomorphic multiplication is given by*

$$B_{\mathsf{finalmult}} = \Delta^{-1}(B_{\mathsf{mult}} + B_{\mathsf{ks}}) + B_{\mathsf{round}},$$

where the bounds $B_{\mathsf{mult}}, B_{\mathsf{ks}}$, and B_{round} are calculated with respect to the relevant norm.

4.4 Central Limit Theorem (CLT) approach

In this section we present an average-case noise analysis for CKKS developed using a Central Limit Theorem or CLT approach. In particular, we consider distributions of the random variables used in CKKS "in the ring". Our analysis is based on continuous noise random variables for simplicity, but the distributional results are applicable to the corresponding discrete noise random variables. Theorem 2 is our main result and shows that the polynomial product (modulo $X^N + 1$) of multivariate Normal vectors with covariance matrices a multiple of the identity matrix I_N can be well-approximated as a multivariate Normal distribution with a covariance matrix of the form $\rho_*^2 I_N + S$ for a specified off-diagonal matrix S. For the purposes of our analysis, we make the *Small-S assumption* that this off-diagonal matrix S is negligible compared to $\rho_*^2 I_N$ and we disregard it. Examination of the form of S indicates that this assumption is reasonable in many circumstances of interest, though we note that this is not the case for example if the mean vectors have large constant components. This is discussed further in Supplementary Material F. Corollary 5 then gives some related results under the Small-S assumption. Our CLT noise analysis follows from these results.

Theorem 2. *Suppose that $Z \sim N(\boldsymbol{\mu}; \rho^2 I_N)$ and $Z' \sim N(\boldsymbol{\mu}'; \rho'^2 I_N)$, then the polynomial product ZZ' (modulo $X^N + 1$) is well-approximated as a multivariate Normal distribution for large N given by*

$$ZZ' \sim N\left(\boldsymbol{\mu}\boldsymbol{\mu}' \,;\, \rho_*^2 I_N + S\right),$$

where $\rho_*^2 = N\rho^2\rho'^2 + \rho'^2 \left\|\boldsymbol{\mu}\right\|_2^2 + \rho^2 \left\|\boldsymbol{\mu}'\right\|_2^2$ and S is an off-diagonal matrix with $S_{i,i'} = \rho'^2 \sum_{j=0}^{N-1} \xi(i-j)\xi(i'-j)\mu_{i-j}\mu_{i'-j} + \rho^2 \sum_{j=0}^{N-1} \xi(i-j)\xi(i'-j)\mu'_{i-j}\mu'_{i'-j}$, for a modified sign function ξ given by $\xi(z) = Sign(z)$ for $z \neq 0$ and $\xi(0) = 1$.

Proof. Let $Y = ZZ'$, so Y has components $Y_i = \sum_{j=0}^{N-1} \xi(i-j) Z_{i-j} Z'_j$, and the mean of such a component Y_i is given by

$$\mathbf{E}(Y_i) = \sum_{j=0}^{N-1} \mathbf{E}(\xi(i-j) Z_{i-j} Z'_j) = \sum_{j=0}^{N-1} \xi(i-j)\mu_{i-j}\mu'_j = (\mu\mu')_i.$$

The summands of a component Y_i are independent, so the variance is given by

$$\begin{aligned}
\mathrm{Var}(Y_i) &= \sum_{j=0}^{N-1} \mathrm{Var}\left(Z_{i-j} Z'_j\right) = \sum_{j=0}^{N-1} \mathbf{E}\left(Z_{i-j}^2\right) \mathbf{E}\left(Z_j'^2\right) - \mathbf{E}\left(Z_{i-j}\right)^2 \mathbf{E}\left(Z'_j\right)^2 \\
&= \sum_{j=0}^{N-1} \left(\rho^2 + \mu_{i-j}^2\right)\left(\rho'^2 + \mu_j'^2\right) - \mu_{i-j}^2 \mu_j'^2 \\
&= \sum_{j=0}^{N-1} \rho^2\rho'^2 + \rho^2\mu_j'^2 + \rho'^2\mu_{i-j}^2 = N\rho^2\rho'^2 + \rho'^2 \left\|\boldsymbol{\mu}\right\|_2^2 + \rho^2 \left\|\boldsymbol{\mu}'\right\|_2^2.
\end{aligned}$$

A similar argument shows that the covariance of distinct components Y_i and $Y_{i'}$ (so $i \neq i'$) is given by

$$\mathrm{Cov}(Y_i, Y_{i'}) = \rho'^2 \sum_{j=0}^{N-1} \xi(i-j)\xi(i'-j)\mu_{i-j}\mu_{i'-j} + \rho^2 \sum_{j=0}^{N-1} \xi(i-j)\xi(i'-j)\mu'_{i-j}\mu'_{i'-j}.$$

Thus $Y = ZZ'$ has covariance matrix $\rho_*^2 I_N + S$. The distribution of Y_i can be addressed by considering the related sum

$$\begin{aligned} Y'_i &= \textstyle\sum_{j=0}^{N-1} \xi(i-j)(Z_{i-j} - \mu_{i-j})(Z'_j - \mu'_j) \\ &= Y_i - \textstyle\sum_{j=0}^{N-1} \xi(i-j)\left(\mu_{i-j}Z'_j + \mu'_j Z_{i-j} - \mu_{i-j}\mu'_j\right). \end{aligned}$$

These summands are a product of $Z_{i-j} - \mu_{i-j} \sim \mathrm{N}(0, \rho^2)$ and $Z'_j - \mu'_j \sim \mathrm{N}(0, \rho'^2)$, so have mean 0 and variance $\rho^2\rho'^2$. Furthermore, the summands of Y'_i are independent and identically distributed, so a Central Limit argument shows that the distribution of Y'_i is well-approximated by a Normal distribution for large N. However, Y_i differs from Y'_i by a Normal random variable so Y_i also well-approximated by a Normal distribution. Thus Y can be approximated by a multivariate Normal distribution as each of its components has a Normal distribution, so we have $Y = ZZ' \sim \mathrm{N}\left(\boldsymbol{\mu}\boldsymbol{\mu}' \,;\, \rho_*^2 I_N + S\right)$. □

Corollary 5. *Suppose that $Z \sim N(\boldsymbol{\mu}; \rho^2 I_N)$ and $Z' \sim N(\boldsymbol{\mu}'; \rho'^2 I_N)$ are independent, λ is a constant vector. Approximations to the distribution of $Z + Z'$, λZ and the rounding $\lfloor Z \rceil$ are then given by:*

$$Z+Z' \sim N(\mathbf{0}, (\rho^2+\rho'^2)I_N), \lambda Z \sim N\Big(\lambda\boldsymbol{\mu} \,;\, \rho^2 \left\|\lambda\right\|_2^2 I_N\Big) \text{ and } \lfloor Z \rceil \sim N\big(\boldsymbol{\mu} \,,\, \rho^2 + \tfrac{1}{12}\big).$$

Furthermore, an approximation to the distribution of ZZ' when the Small-S assumption is valid for ZZ' and an approximation to the distribution of Z^2 when the Small-S assumption is valid for Z^2 are given by:

$$\begin{aligned} ZZ' &\sim N\Big(\boldsymbol{\mu}\boldsymbol{\mu}' \,;\, (N\rho^2\rho'^2 + \rho'^2 \left\|\boldsymbol{\mu}\right\|_2^2 + \rho^2 \left\|\boldsymbol{\mu}'\right\|_2^2) I_N\Big) \\ \text{and} \quad Z^2 &\sim N\Big(\boldsymbol{\mu}^2 \,;\, 2\rho^2(N\rho^2 + 2\left\|\boldsymbol{\mu}\right\|_2^2) I_N\Big). \end{aligned}$$

The idea of the CLT approach is to track the variance ρ^2 of the noise coefficients through a circuit, and only at the end of the circuit convert this variance into a bound. The following Lemmas outline how to update this variance over the course of a circuit, and Lemma 9, part (1) indicates how to derive a bound in the ring given a final variance.

Heuristic 9 *Let $\mathtt{ct}$ be a ciphertext encrypting m with noise $\epsilon \sim N(\mathbf{0}; \rho^2 I_N)$. Let $\mathtt{ct}_{op}$ encrypting m be the ciphertext with noise ϵ_{op} resulting from the rounding or keyswitch operations. Then $\epsilon_{op} \sim N(\mathbf{0}; \rho^2 + \eta_{op}^2 I_N)$, where the respective variances η_{op}^2 of the additive noise introduced by these operations are given by:*

$$\eta_{round}^2 = \tfrac{N}{18} + \tfrac{1}{12}, \quad \text{and} \quad \eta_{ks}^2 = \tfrac{1}{12}P^{-2}Nq_\ell^2\sigma^2 + \mathbb{1}_{P\nmid q_\ell}\left(\tfrac{N}{18} + \tfrac{1}{12}\right).$$

Proof. Follows from results in Section 4.2 and Corollary 5. □

Heuristic 10 *Let* $\mathtt{ct}_1$, $\mathtt{ct}_2$ *be two independent ciphertexts encrypting* m_1 *and* m_2 *with noises* $\epsilon_1 \sim N(\mathbf{0}; \rho_1^2 I_N)$ *and* $\epsilon_2 \sim N(\mathbf{0}; \rho_2^2 I_N)$ *respectively. Let* $\mathtt{ct}_{\mathsf{op}}$ *with noise* ϵ_{op} *be the ciphertext resulting from applying addition, pre-multiply, or multiplication to the ciphertexts* $\mathtt{ct}_1$ *and* $\mathtt{ct}_2$*. Then* $\epsilon_{\mathsf{op}} \sim N(\mathbf{0}; \rho_{\mathsf{op}}^2 I_N)$*. The noise variance* ρ_{add}^2 *for the addition operation is given by:*

$$\rho_{\mathsf{add}}^2 = \rho_1^2 + \rho_2^2.$$

Furthermore, if the Small-S assumption is valid for the distributions $m_1 + \epsilon_1$ *and* $m_2 + \epsilon_2$*, then the noise variances* $\rho_{\mathsf{pre\text{-}mult}}^2$ *and* ρ_{mult}^2 *are given by:*

$$\rho_{\mathsf{pre\text{-}mult}}^2 = N\rho_1^2\rho_2^2 + \rho_2^2 \left\|m_1\right\|_2^2 + \rho_1^2 \left\|m_2\right\|_2^2 \text{ and } \rho_{\mathsf{mult}}^2 = \frac{1}{\Delta^2}\left(\rho_{\mathsf{pre\text{-}mult}}^2 + \eta_{\mathsf{ks}}^2\right) + \eta_{\mathsf{round}}^2,$$

where $\eta_{\mathsf{round}}^2 = \frac{N}{18} + \frac{1}{12}$ *and* $\eta_{\mathsf{ks}}^2 = \frac{1}{12}P^{-2}Nq_\ell^2\sigma^2 + \mathbb{1}_{P \nmid q_\ell}\left(\frac{N}{18} + \frac{1}{12}\right)$.

Proof. Follows from results in Section 4.2, Corollary 5 and Heuristic 9. □

5 Precision

Having analysed the noise resulting from encryption and considered the relationships between bounds in the plaintext space and the message space, we now seek to combine these findings to quantify the impact of both encryption and encoding noise on the precision of plaintexts. First, we give strategies for accounting for precision loss during encoding. Second, we show how to translate noise analysis in the plaintext space into the message space. Lastly, we discuss precision in the CKKS context and provide a precision tracker tool to allow users to gauge how ciphertext operations impact the precision of the output ciphertext.

5.1 Combining Encryption and Encoding Noise

When we encode and encrypt we have two sources of error: encryption noise and the rounding error from encoding. If we were to only ever add or rescale, the relationship between these errors would remain additive, and we could bound them separately, either in the ring or in the message space, and combine at the end. However, multiplying ciphertexts causes these errors to interact, and so an additive approach will not work.

We propose the following solution. The decryption of a freshly encoded and encrypted message has the following form

$$\Delta m(X) + \epsilon_1(X) + \epsilon_2(X),$$

where $\epsilon_1(X)$ is the noise introduced in encryption and $\epsilon_2(X)$ is the encoding noise. We can therefore account for ϵ_2 via modifying the treatment of fresh noise. This is done for each noise analysis method as follows.

Worst Case in the Ring: Here, we observe encoding error is a rounding error so can be accounted for by simply adding $\frac{1}{2}$ to B_{fresh}.
Worst Case in the Canonical Embedding: Following Corollary 1 we add $\frac{N}{\pi}$ to B_{fresh}.
CLT: Unlike previous methods, here we must model encoding error as a random variable to be compatible with the CLT noise analysis. As each coefficient of the encoding error polynomial is taken from $[-\frac{1}{2}, \frac{1}{2}]$, we model the encoding noise as a polynomial with i.i.d. Gaussian coefficients with variance equal to a continuous uniform over $[-\frac{1}{2}, \frac{1}{2}]$. We therefore account for encoding by adding $\frac{1}{12}$ to ρ^2_{fresh}.

We remark that the first two strategies are worst case, and so the resulting bound could be made tighter with additional knowledge of the plaintexts. The CLT approach requires modelling data (via its rounding error) as a random variable, which may not be appropriate in all contexts.

5.2 Message Space Precision Loss from Ring Bounds

Given a bound or variance derived as described in the previous section, we now seek to understand the impact this noise has in the message space. For the Worst Case in the Ring, we use Lemma 7 to give the following result.

Lemma 11. *Suppose in the ring we have $\|\epsilon\|_\infty \leqslant B$. Then after decoding with scale factor Δ we have the error is bounded by $\frac{\sqrt{I(N)^2+1}}{\Delta}B$.*

As $I(N)$ converges from below and quickly (with $\log N$) to $\frac{2N}{\pi}$, we use the limiting value of this expression, $\frac{2NB}{\pi}$.

For the Worst Case in the Canonical Embedding, we observe that decoding consists of evaluating the half canonical embedding and then dividing by Δ, so that we get the following result.

Lemma 12. *Suppose in the canonical embedding we have $\|\epsilon\|^{\mathsf{can}}_\infty \leqslant B$. Then after decoding with scale factor Δ we have that the error is bounded by $\frac{B}{\Delta}$.*

Our CLT analysis shows that the error is normally distributed (in R) as $Z \sim N(\mathbf{0}, \rho^2 I_N)$ for some variance ρ^2. We now give a worst case bound on either total or real error in $\mathbb{C}^{N/2}$. For the total error, we wish to find a bound $B_{\max}$ and an error tolerance α satisfying $\alpha = \mathcal{P}(\|\mathtt{Decode}(m, \Delta)\|_\infty > B_{\max}) = \mathcal{P}(\|Z\|^{\text{can}}_\infty > \Delta B_{\max})$. Thus Lemma 9 (3) shows that

$$B_{\max} = \sqrt{N}\frac{\rho}{\Delta}\, H_{\mathbb{C}}(\alpha, N).$$

For the real error, we wish to find a bound B_{realmax} and an error tolerance $\alpha = \mathcal{P}(\max\{\text{Re}(\tau(Z)_1), \ldots, \text{Re}(\tau(Z)_{N/2})\} > \Delta B_{\text{realmax}})$. Thus Lemma 9 (2) shows that

$$B_{\text{realmax}} = \sqrt{N}\frac{\rho}{\Delta}\, H_{\mathbb{R}}(\alpha, N/2).$$

5.3 Precision Tracker

We now consider how to define a function of a ciphertext that returns some measure of how "correct" a ciphertext is – we will define this as the precision.

We will say a ciphertext has precision k bits if the slotwise difference between the computed result in $\mathbb{C}^{N/2}$ is within 2^{-k} of the true result, or $\|\tilde{\mathbf{z}} - \mathbf{z}\|_\infty \leqslant 2^{-k}$, where $\tilde{\mathbf{z}} \in \mathbb{C}^{N/2}$ is the result after decrypting and decoding, and $\mathbf{z} \in \mathbb{C}^{N/2}$ is the true result of the computation. Observe that choosing negative values for k corresponds to only guaranteeing correctness before the decimal point.

We use this definition and our noise analyses to give precision trackers for each ciphertext, which give different probabilistic (with respect to α) guarantees on the precision of a ciphertext. If an application requires a ciphertext be correct to at least t bits we can modify our function to be a budget (rather than a tracker) by subtracting t.

We now explain how to update the tracker on a ciphertext, using the three noise analysis methods from Section 4 and the methods for translating these to the message space given in Section 5.2. For the worst case in the ring (WCR) and worst case in the canonical embedding (CE), only a total precision tracker is given, while for the CLT approach a tracker for both total and real precision is given. The trackers requires keeping track of a bound B on the noise for the WCR and CE approaches, and a variance ρ^2 for the CLT approach, as well as appropriate message bounds.

Initial.

$$
\begin{aligned}
P_{\mathsf{CE}} &= \log \Delta - \log\left(B_{\mathsf{fresh}} + \frac{N}{\pi}\right) \\
P_{\mathsf{WCR}} &= \log \pi\Delta - \log 2N(B_{\mathsf{fresh}} + 0.5) \\
P_{\mathsf{CLT}} &= \log \Delta - \frac{1}{2}\log N\left(\rho^2_{\mathsf{fresh}} + \frac{1}{12}\right) - \log H_{\mathbb{C}}(\alpha, N) \\
P^{\mathsf{Real}}_{\mathsf{CLT}} &= \log \Delta - \frac{1}{2}\log N\left(\rho^2_{\mathsf{fresh}} + \frac{1}{12}\right) - \log H_{\mathbb{R}}\left(\alpha, \frac{N}{2}\right).
\end{aligned}
$$

Addition. We consider adding $\mathtt{ct}_2$ to $\mathtt{ct}_1$ and updating the tracker on $\mathtt{ct}_1$. Bounds (respectively variances) for $\mathtt{ct}_i$ are given by B_i (respectively ρ_i^2).

$$
\begin{aligned}
P_{\mathsf{CE}} &= P_{\mathsf{CE}} + \log B_1 - \log(B_1 + B_2) \\
P_{\mathsf{WCR}} &= P_{\mathsf{WCR}} + \log B_1 - \log(B_1 + B_2) \\
P_{\mathsf{CLT}} &= P_{\mathsf{CLT}} + \log \rho_1^2 - \log\left(\rho_1^2 + \rho_2^2\right) \\
P^{\mathsf{real}}_{\mathsf{CLT}} &= P^{\mathsf{real}}_{\mathsf{CLT}} + \log \rho_1^2 - \log\left(\rho_1^2 + \rho_2^2\right).
\end{aligned}
$$

(Pre-)Multiplication. We consider multiplying $\mathtt{ct}_1$ by $\mathtt{ct}_2$ and updating the tracker on $\mathtt{ct}_1$, before performing a keyswitch or rescale. Bounds (respectively variances) for $\mathtt{ct}_i$ are given by B_i (respectively ρ_i^2), and ν_1,ν_2 are message bounds in appropriate norms.

$$
P_{\mathsf{CE}} = P_{\mathsf{CE}} + \log B_1 - \log(\nu_1 B_2 + \nu_2 B_1 + B_1 B_2)
$$

$$
\begin{aligned}
P_{\mathsf{WCR}} &= P_{\mathsf{WCR}} + \log B_1 - \log(N(\nu_1 B_2 + \nu_2 B_1 + B_1 B_2)) \\
P_{\mathsf{CLT}} &= P_{\mathsf{CLT}} + \log \rho_1^2 - \log\left(N\rho_1^2\rho_2^2 + \rho_1\nu_2^2 + \rho_2^2\nu_1^2\right) \\
P_{\mathsf{CLT}}^{\mathsf{real}} &= P_{\mathsf{CLT}}^{\mathsf{real}} + \log \rho_1^2 - \log\left(N\rho_1^2\rho_2^2 + \rho_1\nu_2^2 + \rho_2^2\nu_1^2\right).
\end{aligned}
$$

Keyswitch. The error from keyswitch is additive, so the update is given by:

$$
\begin{aligned}
P_{\mathsf{CE}} &= P_{\mathsf{CE}} + \log B - \log(B + B_{\mathsf{ks}}) \\
P_{\mathsf{WCR}} &= P_{\mathsf{WCR}} + \log B - \log(B + B_{\mathsf{ks}}) \\
P_{\mathsf{CLT}} &= P_{\mathsf{CLT}} + \log \rho^2 - \log\left(\rho^2 + \rho_{\mathsf{ks}}^2\right) \\
P_{\mathsf{CLT}}^{\mathsf{real}} &= P_{\mathsf{CLT}}^{\mathsf{real}} + \log \rho^2 - \log\left(\rho^2 + \rho_{\mathsf{ks}}^2\right).
\end{aligned}
$$

Rescale. The update is given by:

$$
\begin{aligned}
P_{\mathsf{CE}} &= P_{\mathsf{CE}} + \log B - \log\left(\frac{B}{\Delta} + B_{\mathsf{round}}\right) \\
P_{\mathsf{WCR}} &= P_{\mathsf{WCR}} + \log B - \log\left(\frac{B}{\Delta} + B_{\mathsf{round}}\right) \\
P_{\mathsf{CLT}} &= P_{\mathsf{CLT}} + \log \rho^2 - \log\left(\frac{\rho^2}{\Delta^2} + \rho_{\mathsf{round}}^2\right) \\
P_{\mathsf{CLT}}^{\mathsf{real}} &= P_{\mathsf{CLT}}^{\mathsf{real}} + \log \rho^2 - \log\left(\frac{\rho^2}{\Delta^2} + \rho_{\mathsf{round}}^2\right).
\end{aligned}
$$

We omit plaintext operations, but comment that, unlike other HE schemes, we advocate treating the encoding error on plaintexts in the same manner as the encryption error on ciphertexts. We give an example for scalar plaintexts as part of the analysis in the next section.

6 Experimental Results

In this section we report on experimental results that validate our results on both noise and precision analysis using our three noise analysis approaches: worst case in the ring (denoted as WCR), worst case in the canonical embedding (denoted as CE) and average case using a Central Limit approach (denoted as CLT). The experiments were run on a 2020 M1 MacBook Pro with 8 cores and 16GB RAM.

Methodology: We ran experiments in the HEAAN [29] library as this implementation most closely resembles "pure CKKS" (the focus of our study). In contrast, other available libraries such as [40, 44] implement RNS variants and other optimisations that may have distorted our conclusions.

Following the recommendations in the HE Standard [1], the LWE parameters (dimension N, ciphertext modulus q, error standard deviation σ) were set as $(\log_2(N), \log_2(q)) \in \{(13, 109), (14, 219), (15, 443)\}$ and $\sigma = 3.2$, and we modified HEAAN to use a uniform ternary secret distribution. In the experiments reported on in this section, we set the error tolerance to $\alpha = 0.0001$ and the scale parameter $\Delta = 2^{40}$. We present additional results, for smaller α and larger Δ,

in Supplementary Material F. Due to the KeySwitch method used in HEAAN, the value of q must be kept lower than is necessary in alternative libraries. For this reason, parameter sets with smaller values of N are too small to support our evaluation circuit.

We run two sets of experiments. The first set corresponds to our analysis in Section 4. In these experiments, in each trial we generate a random plaintext with coefficients in $[-\Delta, \Delta]$, evaluate a specific circuit, and measure noise in the ring after each operation. This approach in the first set of experiments is very similar to previous noise analyses for exact schemes in the literature [14, 17, 18, 26].

In more detail, similarly to [18], the circuit that we evaluate is as follows. We generate fresh ciphertexts $\mathtt{ct}_0$, $\mathtt{ct}_1$ and $\mathtt{ct}_2$ and evaluate the circuit $\mathtt{ct}_2 * (\mathtt{ct}_1 + \mathtt{ct}_0)$, i.e. a homomorphic addition, followed by a (full) homomorphic multiplication. We convert our three methods to bounds on the noise in the ring via (1) reporting the bound for the WCR (2) using the function $H_{\mathbb{R}}(\alpha, N)$ for the CLT variance estimates and (3) reporting the CE bound, as Lemma 1 gives this is the tightest possible ring bound given a canonical embedding bound. For the multiplication operation estimates we use worst-case message bounds, specifically Δ for the WCR, $N\Delta^2$ for the CLT, and $\frac{2N\Delta}{\pi}$ for the CE.

For the second set of experiments, we do not begin and end directly in the plaintext ring, but instead begin and end in the complex message space. These experiments correspond to results in Section 5. In each experiment, in each trial, we generate a vector of random numbers, encode them, encrypt them, and proceed by homomorphically evaluating the circuit as described above. Then, we decrypt and decode and measure the precision loss in the ring. In Table 2, the first two rows correspond to generating random real numbers in $[0, 1]$, encoding and decoding with scale factor Δ, and reporting only the real error on the computation. The second corresponds to generating random complex numbers with real part and imaginary part both uniform in $[0, 1]$ and reporting the magnitude of the largest error.

The results for the first, respectively second sets of experiments are presented in Table 1, respectively Table 2. In both tables, the results presented in the column "average" and "max" are the logarithmic values of the average and maximum noise observed over 1000 trials.

While in exact schemes, it is trivial to observe the noise, this is not so straightforward for CKKS. Our methodology therefore was to generate three plaintexts m_1, m_2 and m_3, and to run those circuits both in the plaintext space and in the ciphertext space. In other words, the noise reported in the tables below is the result of

$$((m_1 + m_2) \cdot m_3) - (\mathsf{Dec}((\mathsf{Enc}(m_1) + \mathsf{Enc}(m_2) \cdot \mathsf{Enc}(m_3))) .$$

Results and discussion:

Table 1 shows that both of our two new noise analysis approaches (WCR and CLT) are much tighter than the previous approach (CE), with WCR being more conservative. Thus, if one is interested in estimating the noise in the ring, we recommend following the CLT method for efficiency, and the WCR for a still

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition noise.						
13	109	10.88	11.26	11.40	11.90	17.95
14	219	11.44	11.86	11.93	12.43	18.98
15	443	12.00	12.33	12.45	12.95	20.01
Multiplication noise.						
13	109	17.31	17.74	18.69	25.90	31.30
14	219	18.38	18.84	19.72	27.43	33.33
15	443	19.43	19.80	20.75	28.95	35.35

Table 1: Average and maximum bits of noise observed in the ring over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses, for $\alpha = 0.0001$ and $\Delta = 2^{40}$.

efficient, but more conservative approach. We note that both of these effectively close the "heuristic-to-practical gap" identified in [18].

Table 2 shows that, perhaps unsurprisingly, the WCR noise analysis method does not perform well in the message space. The CE method is the second best, which is to be expected, since this is its native space. But more importantly, we see the CLT method is approximating the precision loss extremely well in this setting. Indeed, the table shows that the CLT bounds are extremely tight in the message space. We see that in some cases, they may in fact slightly underestimate the precision loss. To conclude, we recommend using the CLT method for an efficient parameter setting, while bearing in mind, since this method is so accurate, careful thought is required in picking the failure probability α. We see in our table that for our parameter choices ($\alpha = 0.0001$ and 1000 runs), the bounds may fail by around -0.2 bits. If we were to use our results to set parameters for an application that might require an even lower failure probability, we could pick parameters according to the CE method, which is more conservative. Alternatively, we could still use the CLT method, but lower the failure probability. Finally, we observe that the results in Table 2 show that this work effectively closes the "heuristic-to-practical gap" observed in [18].

7 Application to Iterative Algorithms

In this section we illustrate the applicability of our encoding and encryption noise analyses in the context of iterative algorithms. These algorithms are widely used in applications of homomorphic encryption, either due to polynomial approximation of functions that are not natively supported [11, 15, 27, 39], or as an artefact of the program being evaluated [9, 32, 34]. The second category includes programs that call Gradient Descent or any (quasi-)Newton method, including (L-)BFGS.

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition, real error.						
13	109	-22.68	-21.89	-22.63	-15.76	-22.02
14	219	-21.59	-20.95	-21.60	-14.23	-20.99
15	443	-20.50	-19.80	-20.57	-12.70	-19.97
Multiplication, real error.						
13	109	-22.26	-21.56	-21.84	-1.76	-21.00
14	219	-21.13	-20.12	-20.81	0.77	-19.97
15	443	-19.97	-18.62	-19.78	3.30	-18.94
Addition, complex error.						
13	109	-22.48	-21.92	-22.55	-15.76	-22.02
14	219	-21.39	-20.72	-21.52	-14.23	-20.99
15	443	-20.31	-19.70	-20.49	-12.70	-19.97
Multiplication, complex error.						
13	109	-23.17	-21.51	-21.26	-1.26	-20.50
14	219	-21.68	-19.92	-20.23	1.27	-19.48
15	443	-20.13	-18.72	-19.20	3.80	-18.45

Table 2: Average and maximum bits of error (either real or complex, as indicated) observed in the message space over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses, for $\alpha = 0.0001$ and $\Delta = 2^{40}$.

When working with an exact scheme, once correctness is assured, then the cleartext convergence properties of the chosen algorithm can be converted to get formal guarantees on the quality of solution in the homomorphic domain. However, when working with an approximate scheme such as CKKS, it is less clear how to obtain such guarantees. In the clear, we would exactly calculate the $(n+1)^{\text{th}}$ estimate from the n^{th} estimate. When working with CKKS, the noise associated with the $(n+1)^{\text{th}}$ calculation may mean that calculating the $(n+1)^{\text{th}}$ estimate may not provide an advantage over the n^{th} estimate.

For a given set of parameters, function, and data set, we want to identify the point at which the evaluation noise begins to interfere with the accurate bits of the calculated solution. More precisely, if $A^{(r)} = \left\| f(\mathbf{z}) - \mathbf{z}^{(r)} \right\|_\infty$ is the absolute error associated with iteration r in the clear, and $\beta^{(r)}$ is a bound on the real noise in $\mathbb{C}^{N/2}$ at iteration r, we wish to identify the point at which

$$\log \beta^{(r)} > \log A^{(r)}.$$

Once this critical point is reached, it is no longer beneficial to perform further iterations. If this critical point is correctly identified, the maximum possible accuracy that can be guaranteed is $-\log A^{(r-1)}$, as by iteration r encryption noise has begun to interfere with the accuracy $-\log A^{(r)}$.

7.1 Encoding as a Change of Argument

When evaluating a function f using CKKS, we have to be mindful that the rounding during encoding effectively changes the argument of f. If the function f is very smooth, or Δ is chosen sufficiently large, this change of argument may not matter. In the iterative algorithm setting, this change of argument strictly limits the number of iterations we can perform. Put another way, we should carefully consider the difference between $f(\mathbf{z})$ and $f(\mathsf{Decode} \circ \mathsf{Encode}(\mathbf{z}))$ when choosing our iteration number.

In this section, we reframe the encoding noise on the data as a change of argument rather than noise. This means that we need not consider the noise from encoding, since it will not interfere with operations until we reach a so called *encoding threshold*, the iteration at which a worst case analysis on the distance between $f(\mathbf{z})$ and f evaluated at its image under encoding gives that the absolute error has become smaller than this distance. This is described in more detail for our case study in Supplementary Material Section E.1.

Instead, we may consider the noise associated with iterations $\mathbf{z}_k^*$, where $\mathbf{z}^*$ is the exact image of $\mathbf{z}$ under encoding.

Measuring. We will consider accurate bits, so letting $x \in \mathbb{R}$ be accurate bits and $e \in \mathbb{R}$ be error, we use the formula [43]

$$x := -\log_2(|e|),$$

naturally extended to work over vectors in $\mathbb{R}^{N/2}$. Thus, if $\mathbf{e} \in \mathbb{R}^{N/2}$ is a vector of errors we define:

$$x := \min_{i=0,\ldots,N/2-1} -\log_2(|e_i|) = -\log_2\left(\|\mathbf{e}\|_\infty\right).$$

7.2 Newton-Raphson Division

So far, the scenario we have described is general to techniques with successive approximations. In order to make our analysis more concrete, we introduce the Newton-Raphson method for division. These analyses could provide a template for application to convergence properties of other iterative algorithms.

The remainder of this section uses the notation of [24, 43]. Recall that the Newton-Raphson method finds the root of a function $g(x)$ via the following update:

$$x_{i+1} = x_i - \frac{g(x_i)}{g'(x_i)}.$$

If certain conditions are satisfied, these updates exhibit quadratic convergence. Roughly speaking, the number of correct decimal places doubles each iteration.

To find $\frac{1}{d}$ using this method, we require a function g which is zero when $x = \frac{1}{d}$. We choose $g(x) = \frac{1}{x} - d$. This gives the following update circuit, of depth 2:

$$x_{i+1} = x_i - \frac{\frac{1}{x_i} - d}{-1/x_i^2} = 2x_i - dx_i^2.$$

We consider absolute and relative error at the i^{th} iteration, $E_i(d)$ and $R_i(d)$ respectively. These shrink quadratically as follows:

$$R_{i+1}(d) = \left|\frac{x_{i+1} - \frac{1}{d}}{1/d}\right| = |dx_{i+1} - 1| = \left|2dx_i - d^2x_i^2 - 1\right| = |dx_i - 1|^2 = R_i(d)^2.$$

Following from this,

$$E_{i+1}(d) = R_{i+1}(d)/d = R_i(d)^2/d = dE_i(d)^2.$$

Thus, at the n^{th} iteration, the relative error is given by $R_n(d) = R_0(d)^{2^n}$, while the absolute error is given by $E_n(d) = (dE_0(d))^{2^n}/d = R_0(d)^{2^n}/d$. Clearly, the starting point for the iterations has a huge impact on the accuracy of the final result. In Table 1 of [43], the authors give different optimal starting points when d is assumed to come from a fixed positive interval $[a, b]$. These starting points are either constant or linear in d, and minimise either the final relative or initial absolute error.

7.3 Initial Approximation, Growth of $A^{(r)}$

Following [43], we take the linear initial approximation which minimises the relative error of the final result. The initial approximation is therefore given by $T_1 d + T_0$, where

$$T_1 = \frac{-8}{a^2 + 6ab + b^2}, \qquad T_0 = \frac{8(a+b)}{a^2 + 6ab + b^2}.$$

This gives the maximum possible error observed after r iterations when inverting d is given by:

$$E_r(d) = R_0(d)^{2^r}/d \leqslant \left(\frac{(b-a)^2}{(a^2 + 6ab + b^2)}\right)^{2^r}/d.$$

We therefore upper bound $E_n(d)$ to give $A^{(r)} = \left(\frac{(b-a)^2}{(a^2+6ab+b^2)}\right)^{2^r}/a$, so the RHS of our critical value criterion is given by:

$$\log A^{(r)} = 2^r \log \frac{(b-a)^2}{(a^2 + 6ab + b^2)} - \log a.$$

We use the values $A^{(r)}$ to derive an encoding threshold in Supplementary Material Section E.1.

7.4 Growth of $\beta^{(r)}$

In this section, we give results on the growth of $\beta^{(r)}$ as we track the growth of error using the three different noise analysis techniques set out in Section 4. As discussed, the CLT method has additional requirements on the independence

of noise polynomials: we consider two degrees of independence assumption, *full independence* (CLT1) and *inter-iteration independence* (CLT2), and our results show that, although assuming less independence gives tighter noise bounds, in this context, both methods predict the same critical points.

We present the expressions for noise growth for each iteration. To avoid introducing redundant notation, we again recycle the terms $B_{\text{fresh}}, B_{\text{ks}}, B_{\text{rs}}$ to denote the bounds derived in the worst case in the ring and canonical embedding. Each analysis also requires different bounds on relevant messages – we derive relevant upper bounds in Supplementary Material Section E.2.

Iteration 0

Recall for Iteration 0, we homomorphically evaluate $T_1 z + T_0$.

Worst Case in the Ring. In the ring, we upper bound the error on the iteration via:

$$\begin{aligned}
B = \frac{1}{\Delta} \left| \lceil \Delta T_1 \rfloor - \Delta T_1 \right| \left\| \Delta \tau(z) \right\|_\infty & \qquad \text{(encoding error on } T_1\text{)} \\
+ \frac{1}{\Delta} \lceil \Delta T_1 \rfloor B_{\text{fresh}} + B_{\text{rescale}} & \qquad \text{(scalar multiplication)} \\
+ \left| \lceil \Delta T_0 \rfloor - \Delta T_0 \right| & \qquad \text{(encoding error on } T_0\text{)} \\
\leqslant b \left| \lceil \Delta T_1 \rfloor - \Delta T_1 \right| + \frac{1}{\Delta} \lceil \Delta T_1 \rfloor B_{\text{fresh}} + \left| \lceil \Delta T_0 \rfloor - \Delta T_0 \right| + B_{\text{rescale}} & \\
=: B_0. &
\end{aligned}$$

Recall from Section 5 that for this method,

$$\beta^{(0)} := \log 2N B_0 - \log \pi \Delta.$$

Canonical Embedding. Similarly, we can upper bound noise in the canonical embedding via:

$$B_0 := b \left| \lceil \Delta T_1 \rfloor - \Delta T_1 \right| + \frac{1}{\Delta} \lceil \Delta T_1 \rfloor B_{\mathsf{fresh}} + \left| \lceil \Delta T_0 \rfloor - \Delta T_0 \right| + B_{\mathsf{rescale}}.$$

And for this method,

$$\beta^{(0)} = \log B_0 - \log \Delta.$$

CLT. We assume the two encoding errors $\lceil \Delta T_1 \rfloor - \Delta T_1$ and $\lceil \Delta T_0 \rfloor - \Delta T_0$ are uniformly distributed on $[\pm(\lceil \Delta T_i \rfloor - \Delta T_i)]$, and so each have variance $\sigma_i^2 = \frac{1}{3} \left(\lceil \Delta T_i \rfloor - \Delta T_i \right)^2$. This gives the initial variance is bounded by:

$$\rho_0^2 = \frac{1}{\Delta^2} \lceil \Delta T_1 \rfloor^2 \rho_{\mathsf{fresh}}^2 + \sigma_1^2 b + \sigma_0^2 + \rho_{\mathsf{rescale}}^2.$$

This gives:

$$\beta^{(0)} = \log \rho_0 \sqrt{N} H_{\mathbb{R}}(\alpha, \tfrac{1}{2} N) - \log \Delta.$$

Further Iterations Suppose after iteration $(k-1)$ our noise is bounded by B_{k-1}, or for the CLT has variance ρ_{k-1}^2. Recall our update circuit is given by $\mathbf{z}^{(k)} = 2\mathbf{z}^{(k-1)} - \mathbf{z}(\mathbf{z}^{(k-1)})^2$, where all operations are component-wise.

Worst Case in the Ring. Let $m_k = \Delta\tau^{-1}(\mathbf{z}^{(k)})$. Assuming we are at level l, after the square we have noise bounded by:

$$B_{\mathsf{square}} := \frac{1}{\Delta}(NB_{k-1}(B_{k-1} + 2\left\|m_{k-1}\right\|_\infty) + B_{\mathsf{ks}}) + B_{\mathsf{rescale}}.$$

The noise bound after the full update is given by:

$$B_k = 2B_{k-1} + \frac{1}{\Delta}\left(N\left(B_{\mathsf{fresh}}B_{\mathsf{square}} + \left\|m\right\|_\infty B_{\mathsf{square}} + \frac{1}{\Delta}\left\|m_{k-1}^2\right\|_\infty B_{\mathsf{fresh}}\right) + B_{\mathsf{ks}}\right) + B_{\mathsf{rescale}},$$

where m is the freshly encoded vector of values.

Canonical Embedding. Similarly, we bound:

$$B_{\mathsf{square}} = \frac{1}{\Delta}(B_{k-1}(B_{k-1} + 2\left\|m_{k-1}\right\|_\infty^{\mathsf{can}}) + B_{\mathsf{ks}}) + B_{\mathsf{rescale}},$$

so that:

$$B_k = 2B_{k-1} + \frac{1}{\Delta}\left(\left(B_{\mathsf{fresh}}B_{\mathsf{square}} + \left\|m\right\|_\infty^{\mathsf{can}} B_{\mathsf{square}} + \frac{1}{\Delta}\left\|m_{k-1}^2\right\|_\infty^{\mathsf{can}} B_{\mathsf{fresh}}\right) + B_{\mathsf{ks}}\right) + B_{\mathsf{rescale}}.$$

CLT - Full Independence. In this setting we assume independence between the noise of $2\mathbf{z}^{(k)}$ and $\mathbf{z}(\mathbf{z}^{(k-1)})^2$, as well as with earlier iterations, so that using results on squaring from Section 4 we have:

$$\rho_{\mathsf{square}}^2 = \frac{1}{\Delta^2}(2N\rho_{k-1}^4 + 4\rho_{k-1}^2\left\|m_{k-1}\right\|_2^2 + \rho_{\mathsf{ks}}^2) + \rho_{\mathsf{rescale}}^2.$$

After the full update:

$$\rho_k^2 = 4\rho_{k-1}^2 + \frac{1}{\Delta^2}\left(N\rho_{\mathsf{fresh}}^2\rho_{\mathsf{square}}^2 + \rho_{\mathsf{fresh}}^2\left\|\frac{1}{\Delta}m_{k-1}^2\right\|_2^2 + \rho_{\mathsf{square}}^2\left\|m\right\|_2^2 + \rho_{\mathsf{ks}}^2\right) + \rho_{\mathsf{rescale}}^2.$$

CLT2 – Inter-Iteration Independence. We additionally consider the effect of dropping the independence assumption between the noise of $2\mathbf{z}^{(k-1)}$ and $\mathbf{z}(\mathbf{z}^{(k-1)})^2$ and find we can reduce the variance each iteration. As experimentally this did not provide any advantage, we defer details to Supplementary Material Section E.3.

7.5 Experimental Results

To evaluate our approach, we sampled vectors of real numbers in the interval $[a, b]$ and evaluated the described division algorithm for 10 iterations. We report on the average behaviour over 100 trials as well as the worst behaviour at each iteration over 100 trials. We remark that we do not advocate the security level of the displayed parameters, but use them to demonstrate the plateau of accuracy.

We additionally indicate the critical points identified by each noise analysis method. Recall that these are the iterations at which each method indicates the encryption noise may have begun to interfere with the accurate bits of the approximation, and so the last accuracy that can be guaranteed by the algorithm is $A^{(r-1)}$. Newton-Raphson exhibits quadratic convergence, so slotwise we should see the number of accurate bits doubling each iteration.

Reinforcing the conclusions of Section 6, the Worst Case in the Ring analysis overestimates the noise in the message space, while the Canonical Embedding and CLT methods frequently concur on the critical point value. We exhibit a parameter set for which the Canonical Embedding overestimates the noise, and so predicts the critical point as too early, while the CLT method(s) fare better in Figure 1c.

For all parameter sets considered, CLT1 (complete independence) and CLT2 (inter-iteration independence) predicted the same critical point. In other words, we observe that weakening the independence heuristic does not change the critical point identified by the CLT, although did produce smaller noise estimates. We suggest this is due to the speed of convergence of this algorithm, and would be a valuable analysis on other iterative algorithms.

Although the strict interpretation of the critical point is the iteration at which noise has already begun to interfere with the convergence, so that convergence should have begun to plateau in some cases the iteration before, we observe that the critical points identified by the CLT often predicts the point at which the plateau begins, rather than the iteration after. One exception to this is Figure 1a, where the plateau begins at iteration 1, as identified by both the CLT methods and the Canonical Embedding method.

An interesting behaviour our experiments identified was that after the convergence stops, the number of accurate bits plateaus, rather than decreases. We observed this with many parameter sets, suggesting that for fast converging iterative methods like Newton-Raphson, precision is not harmed by performing more iterations than the critical point recommends.

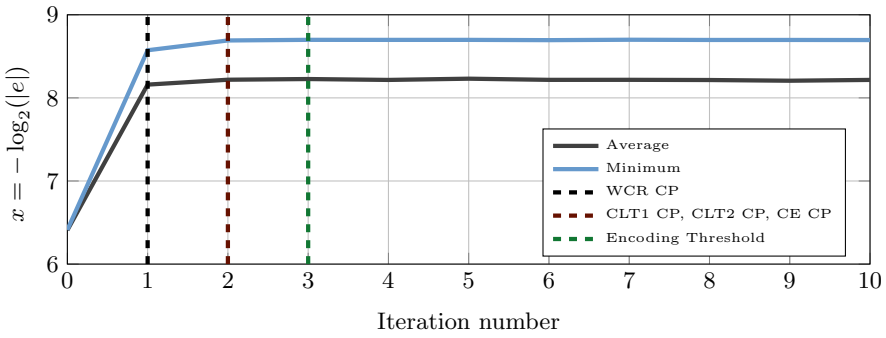
One potential application of the method demonstrated is to embed an algorithm for division at library level. The user could then call division as an operation for a desired precision. The precision tracker would then be updated not with the homomorphic operation updates, but with the error associated with the number of iterations given by the critical point (or critical point minus one).

8 Discussion and Future Work

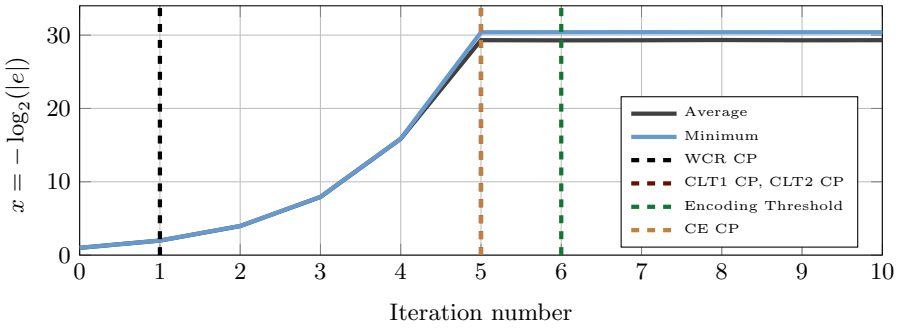
8.1 Li-Micciancio attack and mitigations

We now discuss the implications of our work to the recent CKKS key recovery attack of Li and Micciancio [36], with respect to three countermeasures that have been proposed [13, 36].

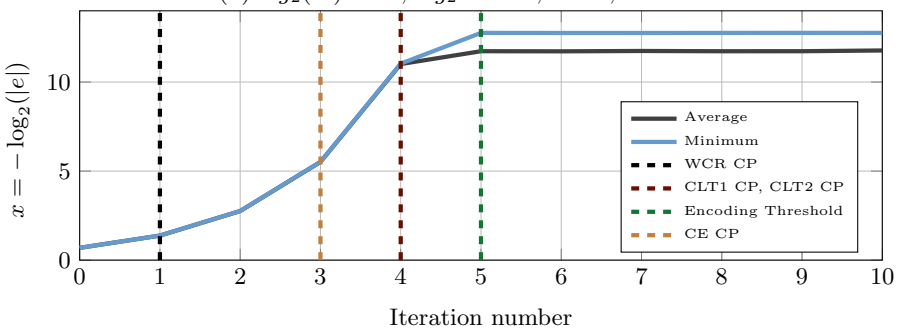
Exact CKKS On possible countermeasure is to make the CKKS scheme exact [36]. We describe how to do so under certain conditions, and prove this



(a) $log_2(N) = 13$, $log_2\Delta = 28$, $a = 5$, $b = 10$.



(b) $log_2(N) = 14$, $log_2\Delta = 50$, $a = 1$, $b = 10$.



(c) $log_2(N) = 15$, $log_2\Delta = 35$, $a = 1$, $b = 15$.

Fig. 1: Accuracy change over successive iterations. Critical Points displayed as vertical lines, using $\alpha = 0.0001$. Note that, in (a) and (b), the values of the CLT1, CLT2 and CE critical points collide, so we plot them as a single line. Similarly, in (c), the value of the CLT1 and CLT2 critical points collide, so we plot them as a single line. All experiments are considered over 100 loops. The number of accurate bits is given by x, as defined in Section 7.1.

modified version is sufficient to satisfy the security notion IND-CPA^D intro-

duced in [36]. In particular, we rely on a correctability condition for the circuit being homomorphically evaluated, defined as follows.

Definition 2. *(Condition for correctability). Fix parameters, and a circuit* $g : (\mathbb{C}^{N/2})^l \to \mathbb{C}^{N/2}$*. Suppose that the message* $g(\mathbf{z}_1, \ldots, \mathbf{z}_l) + \mathbf{e}$ *is obtained from the decoding and decryption of the output ciphertext of the homomorphic evaluation of the circuit* g *such that* $\|\mathbf{e}\|_\infty < B$ *for some bound* $B > 0$*, with all but negligible probability over the choice of inputs and randomness of encryption. Then* g *is* correctable *for these parameters if* $\frac{1}{\Delta'} g(\mathbf{z}_1, ..., \mathbf{z}_l) \in \mathbb{Z}[i]^{N/2}$*, where* $\Delta' = 2^{\lceil \log B \rceil + 1}$*, for all feasible inputs* $\mathbf{z}_i$*. We will call this* Δ' *a* correcting factor*.*

We note that, in this definition, the bound B and the factor Δ' are properties of the function g itself, and not specific to any particular noise analysis method. Informally, correctability ensures that for set parameters, the evaluation noise of g and the desired result $g(\mathbf{z}_1, \ldots, \mathbf{z}_l)$ never interact. Observe that it should always be possible to select parameters for which g is correctable.

We also note that, for real circuits, analogous definitions could be made using bounds on the real noise. We could instead define correctability in the ring, but we opt for a definition in the message space due to its easy interpretability in terms of the precision of input messages.

We now proceed to show how define an exact version of CKKS in the case of correctable circuits, show it is exact in the sense of [36], and outline why our adapted version achieves the IND-CPA^D security.

Lemma 13. *Suppose* $\mathbf{z} \in \mathbb{C}^{N/2}$ *has* $\|\mathbf{z}\|_\infty < B$*, and let* $\Delta' = 2^{\lceil \log B \rceil + 1}$*. Then* $\lceil \frac{1}{\Delta'} \mathbf{z} \rfloor = 0$*.*

Proof. We have that for all i, $|z_i| < B \leqslant \frac{\Delta'}{2}$. We must therefore have that $\left|\frac{z_i}{\Delta'}\right| < \frac{1}{2}$, so that $\lceil \frac{z_i}{\Delta'} \rfloor = 0$, and as claimed $\lceil \frac{1}{\Delta'} \mathbf{z} \rfloor = 0$. □

We therefore introduce the following procedure, performed after decoding, which converts an approximate homomorphic evaluation of a correctable function to an exact one:

Definition 3. *Suppose we have a message* $\mathbf{z} \in \mathbb{C}^{N/2}$*. Then we define the algorithm* $\text{Correct} : \mathbb{C}^{N/2} \times \mathbb{R}^+ \to \mathbb{C}^{N/2}$ *via*

$$\text{Correct}(\mathbf{z}, \Delta') = \Delta' \left\lceil \frac{1}{\Delta'} \mathbf{z} \right\rfloor .$$

Definitions 2 and 3 allow us to derive a correct scheme in the following sense.

Lemma 14. *Fix parameters, and suppose* g *is a correctable circuit with correcting factor* Δ'*. Suppose* $\mathbf{z} = g(\mathbf{z}_1, ..., \mathbf{z}_l) + \mathbf{e}$ *is the result of the homomorphic evaluation of the circuit* g *on inputs* $\mathbf{z}_1, ..., \mathbf{z}_l$*. Then with all but negligible probability, we have* $\text{Correct}(\mathbf{z}, \Delta') = g(\mathbf{z}_1, ..., \mathbf{z}_l)$*.*

Proof.

$$\begin{aligned}\text{Correct}(\mathbf{z}, \Delta') &= \Delta' \left\lceil \frac{1}{\Delta'}\mathbf{z} \right\rfloor \\ &= \Delta' \left\lceil \frac{1}{\Delta'}(g(\mathbf{z}_1, ..., \mathbf{z}_l) + \mathbf{e}) \right\rfloor \\ &= g(\mathbf{z}_1, ..., \mathbf{z}_l) + \left\lceil \frac{1}{\Delta'}\mathbf{e} \right\rfloor \\ &= g(\mathbf{z}_1, ..., \mathbf{z}_l),\end{aligned}$$

as required, with the third equality following due the correctability of g, and the final equality following with all but negligible probability from definition of Δ' and Lemma 13. □

If we therefore augment the decryption procedure by performing `Correct` after decoding, we have that the resulting scheme is exact, or correct, for correctable circuits g in the sense of [36]:

$$\Pr\left(\begin{array}{l}\mathtt{ct}_i \leftarrow \mathtt{Enc}_{\mathtt{pk}}(\mathbf{z}_i) \text{ for } 1 \leqslant i \leqslant l, \\ \quad \mathtt{Dec}_{\mathtt{sk}}(\mathtt{Eval}(g, (\mathtt{ct}_i)_{i=1}^l)) = g((m_i)_{i=1}^l)\end{array}\right) = 1 - \text{negl}(\kappa)\,,$$

where κ is the security parameter. Therefore, by [36, Lemma 1], we have that our corrected scheme is IND-CPA^D secure. In more detail, where the IND-CPA adversary is unable to provide decryptions to the IND-CPA^D adversary in the pure CKKS case due to their inability to simulate the noise, for our corrected scheme the noise is eliminated and so the decryptions are simply a function of messages the adversary possesses.

Note that we have formulated correctability with respect to a negligible failure probability on the bound on the error. This is to keep consistency with the definition of correctness in [36]. However, we could also consider relaxations to a larger failure probability, parameterised by α, as developed in this work. Future work could quantify the security in the IND-CPA^D model of such a relaxation, giving a parametrisable trade off between security and efficient evaluation.

Another natural direction for future work is to explore whether such a corrected version of CKKS can be developed for arbitrary circuits, whose correctability may not be guaranteed. The difficulty here is the rounding during correction may corrupt higher bits, thus the IND-CPA adversary cannot necessarily simulate decryptions. This may yield an effective attack on the corrected CKKS scheme in the IND-CPA^D model.

Adopting such a correcting approach may have been undesirable up to now, given the difficulty of achieving accurate noise bounds B. Indeed, overestimating noise significantly has the undesirable effect of eliminating correct bits during correction. However, we have demonstrated that it is in fact feasible to develop tight noise analyses for CKKS.

Noise flooding Recall that the attack exploits the fact that CKKS decryption outputs $m+e$, where e depends on the secret key. One possible countermeasure, adopted for example in [28, 40], is to 'noise flood': at decryption, sample a fresh Gaussian noise e' of width wide enough to drown out the noise e, and output $m+e+e'$. In HEAAN [13], it is suggested to keep track of an upper bound $B_{\texttt{ctxt}}$ of the noise e and use this bound as the width for e'. Not only does our work provide tighter bounds for $B_{\texttt{ctxt}}$ (resolving an open problem in [13]), but also our analysis enables us to directly characterise the distribution of e and its variance.

In PALISADE [40], the width used for noise flooding is determined by estimating the sample variance of the imaginary component resulting from decoding. Our CLT analysis enables us to provide theoretical justification for this mitigation approach. In particular, we show that both the real and imaginary error are distributed as $N(0, \frac{\rho^2 N}{2\Delta^2} I_{N/2})$. For a real computation, the entire imaginary component is error, so we can take each $\mathrm{Im}(z_i)$ as an i.i.d. sample from $N(0, \frac{\rho^2 N}{2\Delta^2})$, so that we can use the sample variance $(\frac{2}{N}\sum_{j=0}^{N/2-1}(x_i - \overline{x}))$ as an estimator of the variance, and thus specify the distribution of the real error.

Only real decoding In the linear Li-Micciancio attack [36, Theorem 1] the attacker constructs $m = \mathrm{Encode}(\mathrm{Decode}(\mathbf{z}))$ and the attack is successful if $m = \mathrm{Decrypt}(\texttt{ct})$. In this case, the error in the ciphertext component b and the error in the approximate decryption cancel out. A possible mitigation, adopted for example by [40], is to only ever reveal the real result of decoding, $\mathrm{Re}(\mathbf{z})$. This can be justified by showing that a unique m cannot be recovered from $\mathrm{Re}(\mathbf{z})$. In more detail, given $m(X) \in \mathbb{Z}[X]$, we can construct a second polynomial $\tilde{m}(X) \in \mathbb{Z}[X]$ such that $\mathrm{Re}(\mathrm{Decode}(m, \Delta)) = \mathrm{Re}(\mathrm{Decode}(\tilde{m}, \Delta))$. Indeed, let $\tilde{m}(X) = m(X) + X^{N/2}$, so that $\tilde{m}(\zeta_j) = m(\zeta_j) \pm i$ for all primitive roots ζ_j. We have that $\mathrm{Decode}(m, \Delta) - \mathrm{Decode}(\tilde{m}, \Delta)$ is a purely imaginary vector with each entry $\pm\frac{1}{\Delta}i$.

8.2 Future work

In this work we analyse the original CKKS scheme as outlined in [14]. Since then, there have been several modifications and improvements to CKKS. This includes RNS variants, [12, 27] and improved encoding methods [33]. However, we work with the standard 'fresh out of the box' CKKS scheme. This is since not every implementation [40, 44] uses these methods. Not considering RNS variants of CKKS allows our analysis to be more generally applicable. A possible extension to this work would be to consider applying our new analysis methods to these different scenarios. In particular, with some small changes we could apply our methods to the scheme of [33], which improves noise management by encoding with Δ^2.

Our CLT analysis for multiplication required us to make the Small-S assumption. Future work could seek to refine the CLT analysis and remove this assumption through appropriate modelling of the message distribution and an

analysis of "non-spherical" noise. Another interesting direction for future work is to apply the new evaluation noise analysis methods that we have developed (worst-case in the ring, and average-case via CLT arguments) in this work to similar exact schemes, such as BGV and BFV. We expect that both approaches would lead to tighter analyses than a worst-case canonical embedding analysis.

References

[1] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan. Homomorphic encryption security standard. Technical report, HomomorphicEncryption.org, 2018.

[2] David W. Archer, José Manuel Calderón Trilla, Jason Dagit, Alex J. Malozemoff, Yuriy Polyakov, Kurt Rohloff, and Gerard W. Ryan. RAMPARTS: A programmer-friendly system for building homomorphic encryption applications. In Brenner et al. [7], pages 57–68.

[3] Fabian Boemer, Anamaria Costache, Rosario Cammarota, and Casimir Wierzynski. ngraph-he2: A high-throughput framework for neural network inference on encrypted data. In Brenner et al. [7], pages 45–56.

[4] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[5] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[6] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, August 2011.

[7] Michael Brenner, Tancrède Lepoint, and Kurt Rohloff, editors. *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, WAHC@CCS 2019, London, UK, November 11-15, 2019*. ACM, 2019.

[8] Nicolas Brisebarre, Mioara Joldeş, Jean-Michel Muller, Ana-Maria Naneş, and Joris Picot. Error analysis of some operations involved in the cooley-tukey fast fourier transform. *ACM Transactions on Mathematical Software (TOMS)*, 46(2):1–27, 2020.

[9] S. Carpov, N. Gama, M. Georgieva, and J. R. Troncoso-Pastoriza. Privacy-preserving semi-parallel logistic regression training with fully homomorphic encryption. *BMC medical genomics*, 13:88, 2020.

[10] Hao Chen, Ilaria Chillotti, and Yongsoo Song. Improved bootstrapping for approximate homomorphic encryption. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 34–54. Springer, Heidelberg, May 2019.

[11] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. Bootstrapping for approximate homomorphic encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 360–384. Springer, Heidelberg, April / May 2018.

[12] Jung Hee Cheon, Kyoohyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. A full RNS variant of approximate homomorphic encryption. In Carlos Cid and

Michael J. Jacobson Jr:, editors, *SAC 2018*, volume 11349 of *LNCS*, pages 347–368. Springer, Heidelberg, August 2019.

[13] Jung Hee Cheon, Seungwan Hong, and Duhyeong Kim. Remark on the security of ckks scheme in practice. Cryptology ePrint Archive, Report 2020/1581, 2020. `https://eprint.iacr.org/2020/1581`.

[14] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 409–437. Springer, Heidelberg, December 2017.

[15] Jung Hee Cheon, Dongwoo Kim, Duhyeong Kim, Hun-Hee Lee, and Keewoo Lee. Numerical method for comparison on homomorphically encrypted numbers. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 415–445. Springer, Heidelberg, December 2019.

[16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2016.

[17] Ana Costache and Nigel P. Smart. Which ring based somewhat homomorphic encryption scheme is best? In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 325–340. Springer, Heidelberg, February / March 2016.

[18] Anamaria Costache, Kim Laine, and Rachel Player. Evaluating the effectiveness of heuristic worst-case noise analysis in FHE. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *ESORICS 2020, Part II*, volume 12309 of *LNCS*, pages 546–565. Springer, Heidelberg, September 2020.

[19] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Heidelberg, August 2012.

[20] Roshan Dathathri, Blagovesta Kostova, Olli Saarikivi, Wei Dai, Kim Laine, and Madan Musuvathi. EVA: an encrypted vector arithmetic language and compiler for efficient homomorphic computation. In Alastair F. Donaldson and Emina Torlak, editors, *Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2020, London, UK, June 15-20, 2020*, pages 546–561. ACM, 2020.

[21] Roshan Dathathri, Olli Saarikivi, Hao Chen, Kim Laine, Kristin Lauter, Saeed Maleki, Madanlal Musuvathi, and Todd Mytkowicz. Chet: an optimizing compiler for fully-homomorphic neural-network inferencing. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 142–156, 2019.

[22] Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 617–640. Springer, Heidelberg, April 2015.

[23] Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `https://eprint.iacr.org/2012/144`.

[24] M.J. Flynn. On division by functional iteration. *IEEE Transactions on Computers*, C-19(8):702–706, 1970.

[25] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[26] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 850–867. Springer, Heidelberg, August 2012.

[27] Kyoohyung Han and Dohyeong Ki. Better bootstrapping for approximate homomorphic encryption. In Stanislaw Jarecki, editor, *CT-RSA 2020*, volume 12006 of *LNCS*, pages 364–390. Springer, Heidelberg, February 2020.

[28] Heaan v2.1. Online: `https://github.com/snucrypto/HEAAN`, September 2021.

[29] Heaan v1.0. Online: `https://github.com/snucrypto/HEAAN/releases/tag/1.0`, September 2018.

[30] HElib. `https://github.com/shaih/HElib`, January 2019.

[31] I. Iliashenko. *Optimisations of fully homomorphic encryption*. PhD thesis, KU Leuven, 2019.

[32] A. Kim, Y. Song, M. Kim, and J. H. Lee, K.and Cheon. Logistic regression model training based on the approximate homomorphic encryption. *BMC medical genomics*, 11(4):83, 2018.

[33] Andrey Kim, Antonis Papadimitriou, and Yuriy Polyakov. Approximate homomorphic encryption with reduced approximation error. Cryptology ePrint Archive, Report 2020/1118, 2020. `https://eprint.iacr.org/2020/1118`.

[34] M. Kim, Y. Song, S. Wang, Y. Xia, and X. Jiang. Secure logistic regression based on homomorphic encryption: Design and evaluation. *JMIR medical informatics*, 6(2):e19, 2018.

[35] Yongwoo Lee, Joon-Woo Lee, Young-Sik Kim, HyungChul Kang, and Jong-Seon No. High-precision approximate homomorphic encryption by error variance minimization. *IACR Cryptol. ePrint Arch.*, page 1549, 2020.

[36] Baiyu Li and Daniele Micciancio. On the security of homomorphic encryption on approximate numbers. Cryptology ePrint Archive, Report 2020/1533, 2020. `https://eprint.iacr.org/2020/1533`.

[37] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

[38] Sean Murphy and Rachel Player. A central limit framework for ring-lwe decryption. Cryptology ePrint Archive, Report 2019/452, 2019. `https://eprint.iacr.org/2019/452`.

[39] Tabitha Ogilvie, Rachel Player, and Joe Rowell. Improved privacy-preserving training using fixed-hessian minimisation. Cryptology ePrint Archive, Report 2020/1514, 2020. `https://ia.cr/2020/1514`.

[40] PALISADE Lattice Cryptography Library (release 1.11.5). `https://palisade-crypto.org/`, September 2021.

[41] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[42] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[43] M. J. Schulte, J. Omar, and E. E. Swartzlander. Optimal initial approximations for the newton-raphson division algorithm. *Computing*, 53(3-4):233–242, 1994.

[44] Microsoft SEAL (release 3.6). `https://github.com/Microsoft/SEAL`, November 2020. Microsoft Research, Redmond, WA.

[45] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, December 2009.

Supplementary Material

A CKKS scheme

We give a specification of the CKKS scheme [14], a levelled HE scheme and is parameterised by L, p, q_0, N, λ, χ, S and Δ.

The base $p > 0$ and modulus q_0 are used to form the chain of moduli (one for each level) as follows: $q_\ell = p^\ell q_0$ for $1 \leqslant \ell \leqslant L$. The dimension N is a typically chosen as a power of two, and we will only use such N in this work. The dimension N and the chain of moduli parameterise the underlying plaintext and ciphertext rings. The plaintext space is $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$. We denote by q some fixed level in the description below, so that the ciphertext space at any given moment is $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$.

The security parameter is λ. The Ring-LWE error distribution is denoted by χ and is such that each coefficient is sampled as a discrete gaussian with standard deviation $\sigma = 3.2$ [1]. The parameter S denotes the secret key distribution, which we take as uniform ternary of fixed length.

The CKKS scheme is comprised of the following algorithms: `SecretKeyGen`, `PublicKeyGen`, `EvaluationKeyGen`, `Encrypt`, `Decrypt`, `Add`, `Pre-Multiply`, `KeySwitch`, `Rescale`, and `Multiply`.

`SecretKeyGen`(λ)**:** Sample $s \leftarrow S$ and output $\mathtt{sk} = (1, s)$.

`PublicKeyGen`(`sk`)**:** Recall $\mathtt{sk} = (1, s)$ and sample $a \leftarrow R_q$ uniformly at random and $e \leftarrow \chi$. Output $\mathtt{pk} = ([-as + e]_q, a)$.

`EvaluationKeyGen`(`sk`, w)**:** Sample $a' \leftarrow R_{P \cdot q}$ uniformly at random and $e' \leftarrow \chi$. Output $\mathtt{evk} = \left([-a's + e' + Ps^2]_{P \cdot q}, a'\right)$.

`Encrypt`(`pk`, m)**:** For the message $m \in R$. Let $\mathtt{pk} = (p_0, p_1)$, sample $v \leftarrow S$ and $e_1, e_2 \leftarrow \chi$. Output $\mathtt{ct} = ([m + p_0 v + e_1]_q, [p_1 v + e_2]_q)$.

`Decrypt`(`sk`, `ct`)**:** Let $\mathtt{ct} = (c_0, c_1)$. Output $m' = [c_0 + c_1 s]_q$.

`Add`($\mathtt{ct}_0, \mathtt{ct}_1$)**:** Output $\mathtt{ct} = ([\mathtt{ct}_0[0] + \mathtt{ct}_1[0]]_q, [\mathtt{ct}_0[1] + \mathtt{ct}_1[1]]_q)$.

`Pre-Multiply`($\mathtt{ct}_0, \mathtt{ct}_1$)**:** Set $d_0 = [\mathtt{ct}_0[0]\mathtt{ct}_1[0]]_q$, $d_1 = [\mathtt{ct}_0[0]\mathtt{ct}_1[1] + \mathtt{ct}_0[1]\mathtt{ct}_1[0]]_q$, and $d_2 = [\mathtt{ct}_0[1]\mathtt{ct}_1[1]]_q$. Output $\mathtt{ct} = (d_0, d_1, d_2)$.

`KeySwitch`(`ct`, `evk`)**:** Here, `ct` is an output of `Pre-Multiply`. Let $\mathtt{ct}[0] = d_0$, $\mathtt{ct}[1] = d_1$ and $\mathtt{ct}[2] = d_2$. Recall $\mathtt{evk}[0] = -a's + e' + Ps^2$ and $\mathtt{evk}[1] = a'$. Set $c'_0 = [d_0 + \lfloor P^{-1} \cdot d_2 \cdot (-a's + e' + Ps^2) \rceil]_q$, and $c'_1 = [d_1 + \lfloor P^{-1} \cdot d_2 \cdot a' \rceil]_q$. Output $\mathtt{ct}' = (c'_0, c'_1)$.

`Rescale`(`ct`, q') : Let $\mathtt{ct} = (c_0, c_1)$. Set $c'_0 = \left[\left\lfloor \frac{q'}{\Delta} c_0 \right\rceil\right]_{q'}$ and $c'_1 = \left[\left\lfloor \frac{q'}{\Delta} c_1 \right\rceil\right]_{q'}$. Output $\mathtt{ct} = (c'_0, c'_1)$.

`Multiply`(`ct`)**:** Here, `ct` is an output of `Pre-Multiply`. We apply `KeySwitch` followed by `Rescale`. Output $\mathtt{ct} = \frac{1}{\Delta}\left[(d_0, d_1) + \lfloor P^{-1} d_2 \mathtt{evk} \rceil\right]_q$.

B Lattice of Preserved Vectors

In section 3 we introduced the notion of *preserved* vectors on a lattice. In this supplementary material, we give a definition for the lattice of preserved vectors and outline some preliminary theoretical results.

Definition 4. *A vector $\boldsymbol{z} \in \mathbb{C}^{N/2}$ is **preserved under scale factor Δ** if it does not get rounded during encoding, or equivalently if $\Delta\tau^{-1}(\boldsymbol{z}) \in \mathbb{Z}[X]/(X^N+1)$.*

Lemma 15. *Let $N \geqslant 4$. The lattice of preserved vectors does not contain any canonical vectors for Δ integral.*

Proof. The lattice of preserved vectors does not contain any canonical vectors for Δ integral means that there are no vectors which have zeroes in all but one entry. Equivalently, there are no scalar multiples of the standard canonical basis vectors.

We prove this by assuming such a vector, showing by consideration of the 0^{th} and $N/2^{\text{th}}$ coefficients that both its real and imaginary parts are integral when multiplied by Δ, and deriving a contradiction by considering the $N/4^{\text{th}}$ and $3N/4^{\text{th}}$ coefficients.

Suppose we have some preserved vector $\mathbf{z} \in \mathbb{C}^{N/2}$ with $z_j = \alpha\delta_{jk}$. This vector is zero in all but the k^{th} entry, where it has value $\alpha \in \mathbb{C}$. Suppose $\mathbf{z}$ is preserved under scale factor Δ, so that by definition $m(X) = \tau^{-1}(z) \in \frac{1}{\Delta}\mathbb{Z}[X]/(X^N+1)$. We have an explicit formula for the coefficients of m.

$$\tau^{-1}(\boldsymbol{z}) = \frac{1}{N}\left(\overline{U}^T\boldsymbol{z} + U^T\overline{\boldsymbol{z}}\right). \tag{3}$$

Combining this with the definition of $\mathbf{z}$ gives the following:

$$\begin{aligned} m_j &= \frac{1}{N}\sum_{l=0}^{\frac{N}{2}-1} \overline{\zeta}_l^j z_l + \zeta_l^j \overline{z}_l \\ &= \frac{1}{N}\left(\overline{\zeta}_k^j \alpha + \overline{\alpha}\zeta_k^j\right). \end{aligned}$$

We require $m_j \in \frac{1}{\Delta}\mathbb{Z}$ for each $0 \leqslant j \leqslant N-1$.

Setting $j = 0$, gives:

$$m_0 = \frac{1}{N}(\alpha + \overline{\alpha}) \in \frac{1}{\Delta}\mathbb{Z},$$

so that $\mathrm{Re}(\alpha) \in \frac{1}{\Delta}\mathbb{Z}$.

Conversely, considering $j = N/2$ gives:

$$m_{N/2} = \pm\frac{1}{N}(\alpha i - \overline{\alpha} i) \in \frac{1}{\Delta}\mathbb{Z}.$$

Hence, $\mathrm{Im}(\alpha) \in \frac{1}{\Delta}\mathbb{Z}$, and we have $\alpha \in \frac{1}{\Delta}\mathbb{Z}[i]$.

Now as before, let $\zeta_j = \exp\left(\frac{(2j+1)\pi i}{N}\right)$, and consider $j = N/4$:

$$Nm_{N/4} = \overline{\zeta}_k^{N/4}\alpha + \overline{\alpha}\zeta_k^{N/4} \tag{4}$$

$$= 2\mathrm{Re}(\alpha)\cos\left(\frac{(2k+1)\pi}{4}\right) + 2\mathrm{Im}(\alpha)\sin\left(\frac{(2k+1)\pi}{4}\right) \tag{5}$$

$$= \sqrt{2}\left[\mathrm{Re}(\alpha)\left(\cos\left(\frac{k\pi}{2}\right) - \sin\left(\frac{k\pi}{2}\right)\right) + \mathrm{Im}(\alpha)\left(\cos\left(\frac{k\pi}{2}\right) + \sin\left(\frac{l\pi}{2}\right)\right)\right]. \tag{6}$$

To ensure this equation is rational requires that the contents of the square brackets is irrational or zero. In order to demonstrate that it cannot be zero, we must consider one more coefficient and show they cannot both be zero. Now let $j = 3N/4$. Skipping intermediate expressions, gives:

$$Nm_{3N/4} = \overline{\zeta}_k^{3N/4}\alpha + \overline{\alpha}\zeta_k^{3N/4} \tag{7}$$

$$= 2\mathrm{Re}(\alpha)\cos\left(\frac{3(2k+1)\pi}{4}\right) + 2\mathrm{Im}(\alpha)\sin\left(\frac{3(2k+1)\pi}{4}\right) \tag{8}$$

$$= \sqrt{2}(-1)^k\left[\mathrm{Re}(\alpha)\left(-\cos\left(\frac{k\pi}{2}\right) - \sin\left(\frac{k\pi}{2}\right)\right) + \mathrm{Im}(\alpha)\left(\cos\left(\frac{k\pi}{2}\right) - \sin\left(\frac{k\pi}{2}\right)\right)\right]. \tag{9}$$

Now we proceed according to parity of k: if $k = 0 \mod 2$, take $(8) - (11)$ to give $\sqrt{2}\mathrm{Re}(\alpha) \in \frac{1}{\Delta}\mathbb{Z}$, giving $\mathrm{Re}(\alpha) = 0$ so that $\mathrm{Im}(\alpha) = 0$. If instead $k = 1 \mod 2$, take $(8) + (11)$ to give $\sqrt{2}\mathrm{Im}(\alpha) \in \frac{1}{\Delta}\mathbb{Z}$, so that $\mathrm{Im}(\alpha) = 0$ and again $\mathrm{Re}(\alpha) = 0$, so that our original vector is the zero vector. □

C Proof of Lemma 9

Proof of Lemma 9 Part 1
If $Z_j \sim \mathrm{N}(0, \rho^2)$, then $|Z_j| \sim \rho\chi_1$ has a scaled χ-distribution with 1 degree of freedom and cumulative distribution function

$$F_{|Z_j|}(t) = \mathcal{P}(|Z_j| < t) = \texttt{erf}\left(\frac{t}{\sqrt{2}\rho}\right) \qquad [t > 0].$$

Thus $\|Z\|_\infty = \max\{|Z_1|, \ldots, |Z_N|\}$ has distribution function

$$F_{\|Z\|_\infty}(t) = \mathcal{P}(\|Z\|_\infty \leq t) = \mathcal{P}(|Z_j| < t)^N = \texttt{erf}\left(\frac{t}{\sqrt{2}\rho}\right)^N.$$

The probability of $\|Z\|_\infty^{\mathsf{can}}$ exceeding a bound B is therefore given by

$$\mathcal{P}(\|Z\|_\infty > B) = 1 - \mathcal{P}(\|Z\|_\infty \leq t) = 1 - \texttt{erf}\left(\frac{B}{\sqrt{2}\rho}\right)^N,$$

setting this equal to α gives the claimed probability $(1-\alpha)$ bound.

Proof of Lemma 9 Part 2
We define a "real" version $\tau' \colon \mathbb{R}^N \to \mathbb{R}^N$ of the function $\tau \colon \mathbb{R}^N \to \mathbb{C}^{N/2}$ by the $N \times N$ matrix

$$T' = \begin{pmatrix} (\mathrm{Re}(\zeta_j^k)) \\ (\mathrm{Im}(\zeta_j^k)) \end{pmatrix} \qquad \begin{bmatrix} j = 0, \ldots, \frac{1}{2}N - 1 \\ k = 1, \ldots, N \end{bmatrix}.$$

The matrix $\frac{1}{2}N \times N$ matrix $T = \left(I_{N/2} \middle| \, i\, I_{N/2} \right) T'$ then gives the canonical embedding τ. The matrix T' satisfies $T'T'^T = \frac{1}{2}NI_N$ as τ is a scaled isometry with $|\tau(x)|^2 = \frac{1}{2}N|x|^2$ (see Section 3 preamble), so that for $Z \sim \mathrm{N}(\mathbf{0}; \rho^2 I_N)$, we have

$$X = T'Z \sim \mathrm{N}\left(\mathbf{0}; \frac{N}{2}\rho^2 I_N\right).$$

The vector $Y := \tau(Z)$ therefore has real part $\mathrm{Re}(Y) = (X_0, X_1, ..., X_{\frac{N}{2}-1})$ and imaginary part $\mathrm{Im}(Y) = (X_{\frac{N}{2}}, X_{\frac{N}{2}+1}, ..., X_{N-1})$, so that the distributions of $\mathrm{Re}(Y)$ and $\mathrm{Im}(Y)$ can be read off. The bounds follows by part (1) applied to $\mathrm{Im}(Y)$ and $\mathrm{Re}(Y)$.

Proof of Lemma 9 Part 3
If we write $Y_j = X_j + iX_{j+\frac{1}{2}N}$, then $|Y_j|^2 = X_j^2 + X_{j+\frac{1}{2}N}^2$, which has a scaled χ_2^2-distribution as the sum of two independent squared standard Normal random variables. Thus

$$|Y_0|, \ldots, |Y_{\frac{1}{2}N-1}| \;\sim\; (\tfrac{1}{2}N)^{\frac{1}{2}}\rho\chi_2$$

are independent and identically distributed random variables having a scaled χ distribution with 2 degrees of freedom with cumulative distribution function

$$F_{|Y_j|}(t) = \mathcal{P}\left(|Y_j| \le t\right) = 1 - \exp\left(-\frac{t^2}{N\rho^2}\right) \qquad [t > 0].$$

Thus $\|Y\|_\infty$ has distribution function

$$F_{\|Y\|_\infty}(t) = \mathcal{P}(\|Y\|_\infty \le t) = \mathcal{P}\left(|Y_j| \le t\right)^{\frac{1}{2}N} = \left(1 - \exp\left(-\frac{t^2}{N\rho^2}\right)\right)^{\frac{1}{2}N}.$$

For a bound B on $\|Y\|_\infty$, the failure probability α is therefore given by

$$\alpha = \mathcal{P}(\|Y\|_\infty > B) = 1 - \mathcal{P}(\|Y\|_\infty \le t) = 1 - \left(1 - \exp\left(-\frac{B^2}{N\rho^2}\right)\right)^{\frac{1}{2}N},$$

so we derive a $1-\alpha$ probability on $\|Y\|_\infty$ as

$$B = \sqrt{N}\rho\left(-\log(1 - (1-\alpha)^{\frac{2}{N}})\right)^{\frac{1}{2}},$$

which is also a $1-\alpha$ probability bound on $\|Z\|_\infty^{\mathsf{can}}$ since $\|Z\|_\infty^{\mathsf{can}} = \|\tau(Z)\|_\infty = \|Y\|_\infty$.

D Proof of Heuristic 1, 2 and 3

Proof of Heuristic 1
Let the ciphertext $(\mathtt{ct}_0, \mathtt{ct}_1)$ be a fresh ciphertext encrypting a message m. Then $\mathtt{ct}_0 + \mathtt{ct}_1 s = m - sav + ve + e_1 + sav + se_2$. Therefore, the ciphertext has noise $\epsilon = ve + se_2 + e_1$, where $e, e_1, e_2 \sim \mathrm{N}(\mathbf{0}; \sigma^2 I_N)$ are independent and v, s are sampled from a uniform ternary distribution. We treat these as continuous random variables for simplicity, but the distributional results are applicable to the corresponding discrete random variables. For a fixed ternary vector v, the random variable ve has distribution $ve \sim \mathrm{N}(\mathbf{0}; \|v\|_2^2 \sigma^2 I_N)$, and similarly for se_2. By independence, we can therefore say that the noise on a fresh ciphertext has distribution $ve + se_2 + e_1 \sim \mathrm{N}(\mathbf{0}; (\|v\|_2^2 + \|s\|_2^2 + 1)\sigma^2 I_N)$.

We note that $\|v\|_2^2$ and $\|s\|_2^2$ are typically very close to $\frac{2}{3}N$ for large N, so we can approximate this distribution by $\mathrm{N}(\mathbf{0}; (\frac{4}{3}N + 1)\sigma^2 I_N)$.

Proof of Heuristic 2
Let $\mathtt{ct} = (\mathtt{ct}_0, \mathtt{ct}_1)$ be a ciphertext encrypting a message m with noise ε. Let $\mathtt{ct}_{\mathsf{round}} = (\mathtt{ct}'_0, \mathtt{ct}'_1)$ be the result of applying component-wise rounding to $\mathtt{ct}$. Then $\mathtt{ct}'_0 + \mathtt{ct}'_1 s = \mathtt{ct}_0 + \tau_0 + (\mathtt{ct}_1 + \tau_1)s = \mathtt{ct}_0 + \mathtt{ct}_1 s + \tau_0 + \tau_1 s = m + \varepsilon + \tau_0 + \tau_1 s$, where τ_i is the rounding error introduced in each component. The rounding process thus introduces an additive noise $\epsilon := \tau_0 + \tau_1 s$. The τ_i can be modelled as being drawn uniformly at random with components in $[-\frac{1}{2}, \frac{1}{2}]$. Thus the j^{th} component of this additive error is given by

$$\epsilon_j = \tau_{0,j} + \sum_{k=0}^{N-1} \xi(k-j) s_k \, \tau_{1,j-k},$$

where ξ is a modified sign function given by $\xi(z) = \mathrm{Sign}(z)$ for $z \neq 0$ and $\xi(0) = 1$ arising from the multiplication modulo $X^n + 1$ (also see Theorem 2). This component ϵ_j therefore has mean 0 and variance satisfying

$$\begin{aligned} \eta^2_{\mathsf{round}} = \mathrm{Var}(\epsilon_j) &= \mathrm{Var}(\tau_{0,j}) + \textstyle\sum_{k=0}^{N-1} (\xi(k-j)s_k)^2 \, \mathrm{Var}(\tau_{1,j-k}) \\ &= \tfrac{1}{12} + \|s\|_2^2 \cdot \tfrac{1}{12} \approx \tfrac{1}{18}N + \tfrac{1}{12} \end{aligned}$$

in most situations, as $\|s\|_2^2$ is typically close to $\frac{2}{3}N$. Similarly, we can show for $j' \neq j$ that

$$\begin{aligned} \mathrm{Cov}(\epsilon_j, \epsilon_{j'}) = \mathbf{E}(\epsilon_j \epsilon_{j'}) &= \textstyle\sum_{k=0}^{N-1} \xi(k-j)^2 s_k s_{j'-j+k} \, \mathrm{Var}(\tau_{1,j-k}) \\ &= \tfrac{1}{12} \textstyle\sum_{k=0}^{N-1} s_k s_{j'-j+k}, \end{aligned}$$

as s_i are uniformly distributed over $\{-1, 0, 1\}$. Thus $\mathrm{Cov}(\epsilon_j, \epsilon_{j'})$ can be modelled by a Normal $\mathrm{N}(0, \frac{1}{72}N)$ distribution, so is far smaller than η^2_{round}, and we can regard $\mathrm{Cov}(\epsilon_j, \epsilon_{j'}) \approx 0$.

We have now found the mean and variance and established that the covariance is negligible. It remains to show normality. As shown above, the error component ϵ_j is the sum of the independent random variables $\tau_{0,j}$ and $\xi(k-j)s_k\tau_{1,j-k}$

(for $k = 0, \ldots, N-1$) which are each independent random variables uniformly distributed on $[-\frac{1}{2}, \frac{1}{2}]$ (when $s_k \neq 0$). Thus, ϵ_j is the sum of about $(\frac{2}{3}N + 1)$ independent and identically distributed random variables with mean 0 and variance $\frac{1}{12}$, so the Central Limit Theorem shows that ϵ_j has an approximate Normal $\mathrm{N}(0, \eta^2_{\mathsf{round}})$ distribution. Thus ϵ can be modelled as a multivariate Normal $\mathrm{N}(\mathbf{0}; \eta^2_{\mathsf{round}} I_N)$ distribution.

Proof of Heuristic 3 The keyswitch operation introduces an additive error $\epsilon_{\mathsf{ks}} = P^{-1} q_l \cdot d_2 e' + \epsilon_{\mathsf{round}}$ (see for example [14, Lemma 3]). In this expression, $e' \sim \mathrm{N}(\mathbf{0}; \sigma^2 I_N)$ is the Ring-LWE noise term in the evaluation key, $d_2 = [\mathtt{ct}_0[1]\mathtt{ct}_1[1]]_{q_\ell}$ is a component from the output of pre-multiply, and $\epsilon_{\mathrm{round}}$ is a possible rounding error created by dividing by P^{-1}. We can regard d_2 as having a componentwise uniform distribution on $(-\frac{1}{2}q, \frac{1}{2}q)^N$, so has mean $\mathbf{0}$ and component variance $\frac{1}{12}q^2$. A Central Limit argument similar to the proof of Lemma 2 then shows that $P^{-1}(d_2 e') \sim \mathrm{N}(\mathbf{0}\ ;\ \frac{1}{12} P^{-2} N q_\ell^2 \sigma^2 I_N)$.

The additional rounding error e_{round} is required when P does not divide q_ℓ, in which case Lemma 2 shows that $e_{\mathsf{round}} \sim \mathrm{N}(\mathbf{0}; (\frac{1}{18}N + \frac{1}{12}) I_N)$. In these circumstances, ϵ_{ks} is the sum of two independent multivariate Normal distributions, so has a multivariate Normal distribution itself with mean $\mathbf{0}$ and component variance $\frac{1}{12} P^{-2} N q_\ell^2 \sigma^2 + (\frac{1}{18}N + \frac{1}{12})$.

E Iterative Algorithms

E.1 Derivation of Encoding Threshold

Recall that we are interested in identifying the distance between $f(\mathbf{z})$ and $f(\mathbf{z}^*)$, and then identifying the iteration for which the error in a Newton-Raphson approximation becomes smaller than this difference. Using that $z_i \in [a, b]$ we can bound:

$$\begin{aligned}
\log(\|f(\mathbf{z}) - f(\mathbf{z}^*)\|_\infty) &= \log(\max_i \left| \frac{1}{z_i} - \frac{1}{z_i^*} \right|) \\
&= \log\left(\max_i \left| \frac{z_i - z_i^*}{z_i z_i^*} \right| \right) \\
&\leqslant \log\left(\max_i |z_i - z_i^*| \right) - \log\left(\min_i |z_i z_i^*| \right) \\
&\leqslant \log N + \log \pi - \log \Delta - \log a - \log\left(a - \frac{N}{\Delta \pi} \right),
\end{aligned}$$

where we use that $|z_i - z_i^*| \leqslant \frac{N}{\pi \Delta}$. Taking this RHS as k, we use the values $A^{(r)}$ to identify the iteration r for which $\log A^{(r)} < k$, and indicate this as an encoding threshold.

E.2 Message Bounds for Newton-Raphson Division

Iteration 0. Let $m = \mathrm{Encode}(\mathbf{z}, \Delta) = \Delta \tau^{-1}(\mathbf{z})$, where we have no rounding as by assumption $\mathbf{z}$ is preserved. Then using that the entries of $\mathbf{z}$ are in $[a, b]$ we can

bound $\|m\|_\infty^{\mathsf{can}} \leqslant \Delta b$, use Lemma 1 to bound $\|m\|_\infty \leqslant \Delta b$, and finally use that τ is a scaled isometry in the 2-norm to give $\|m\|_2^2 \leqslant \Delta^2 b^2$.

Iteration 1. For this iteration, we require bounds on $m_0 = \Delta\tau^{-1}(\mathbf{z}^{(0)})$. Since T_1 is negative we have that $T_0 + T_1 x$ is decreasing in x, and so can bound $\|m_0\|_\infty^{\mathsf{can}} \leqslant \Delta(T_1 - T_0 a)$,$\|m_k\|_\infty \leqslant \Delta(T_1 - T_0 a)$, and $\|m_k\|_2^2 \leqslant \Delta^2(T_1 - T_0 a)^2$.

Further Iterations. To bound $m_k = \Delta\tau^{-1}(\mathbf{z}^{(k)})$ for $k > 0$, we use that $2x - dx^2 \leqslant \frac{1}{d} \leqslant \frac{1}{a}$ for $d \in [a, b]$ to give that $\|m_k\|_\infty^{\mathsf{can}}, \|m_k\|_\infty \leqslant \Delta/a$ $\|m_k\|_2^2 \leqslant \Delta^2/a^2$.

As we have derived bounds on the entries in the canonical embedding, we can use that m_k^2 corresponds to component-wise squaring in the canonical embedding to give bounds on $\left\|m_k^2\right\|_\infty, \left\|m_k^2\right\|_\infty^{\mathsf{can}}$ and $\left\|m_k^2\right\|_2^2$.

We remark that tighter bounds, and thus tighter noise analysis, can be obtained by considering the actual data in question, rather than deferring to worst case behaviour in the interval $[a, b]$. However, we leave our results general, for easier applicability, especially if practitioners wish to call the division algorithm as part of a larger homomorphic program.

E.3 CLT2 – Inter Iteration Independence

For this method, we will assume the error at iteration k is independent of the initial data, but we will not assume independence of the errors on the terms $2m_{k-1}(X)$ and $m(X)(m_{k-1})^2$. We are therefore interested in calculating the variance of $(2M - \frac{1}{\Delta^2} Z \odot M \odot M)$, where $M \sim N(\mu, \rho^2 I_N)$ and $Z \sim N(d, \rho'^2 I_N)$.

$$\mathrm{Var}(2M_i - \frac{1}{\Delta^2}(Z \odot M \odot M)_i) = \mathrm{Var}(2M_i) + \mathrm{Var}(\frac{1}{\Delta^2}(Z \odot M \odot M)_i) - \frac{4}{\Delta^2}\mathrm{Cov}(M_i, ((Z \odot M \odot M)_i)),$$

thus we obtain the variance terms we obtained when assuming full independence, but if we can show $\mathrm{Cov}(M_i, ((Z \odot M \odot M)_i)) > 0$ we will have that our variance is strictly smaller without this assumption.

By previous results, we have that $\mathbf{E}(M_i) = \mu_i, \mathbf{E}((Z \odot M \odot M)_i) = (d \odot \mu \odot \mu)_i$, so that to calculate the covariance we only need to calculate $\mathbf{E}(M_i(Z \odot M \odot M)_i)$. For this, we use the following result.

Lemma 16. $\mathbf{E}[M_i(M \odot M)_j] = \mu_i(\mu \odot \mu)_j + 2\rho^2(\mu_{j-i}\mathbb{1}_{i \leqslant j} - \mu_{j-i+N}\mathbb{1}_{i \geqslant j+1})$.

Proof.

$$\begin{aligned}
\mathbf{E}[M_i(M \odot M)_j] &= \mathbf{E}\left[\sum_{k=0}^{j} M_i M_k M_{j-k} - \sum_{k=j+1}^{N-1} M_i M_k M_{j-k+N}\right] \\
&= \mu_i(\mu \odot \mu)_j \\
&\quad + \rho^2(2\mu_{j-i}\mathbb{1}_{i \leqslant j}\mathbb{1}_{j \neq 2i} + \mu_i \mathbb{1}_{j \text{ even}}\mathbb{1}_{j \neq 2i} + 3\mu_i \mathbb{1}_{j=2i}) \\
&\quad - \rho^2(2\mu_{j-i+N}\mathbb{1}_{i \geqslant j+1}\mathbb{1}_{j \neq 2i-N} + \mu_i \mathbb{1}_{j \text{ even}}\mathbb{1}_{j \neq 2i-N} + 3\mu_i \mathbb{1}_{j=2i-N}) \\
&= \mu_i(\mu \odot \mu)_j + 2\rho^2(\mu_{j-i}\mathbb{1}_{i \leqslant j} - \mu_{j-i+N}\mathbb{1}_{i \geqslant j+1}).
\end{aligned}$$

□

We can therefore compute

$$\begin{aligned}
\mathbf{E}[M_i(Z \odot M \odot M)_i] &= \mathbf{E}\left[\sum_{j=0}^{i} M_i Z_{i-j}(M \odot M)_j - \sum_{j=i+1}^{N-1} M_i Z_{i-j+N}(M \odot M)_j\right] \\
&= \sum_{j=0}^{i} d_{i-j}\mathbf{E}[M_i(M \odot M)_j] - \sum_{j=i+1}^{N-1} d_{i-j+N}\mathbf{E}[M_i(M \odot M)_j] \\
&= \mu_i(d \odot \mu \odot \mu)_i \\
&\qquad + 2\rho^2(d_0\mu_0 - \sum_{j=0}^{i-1} d_{i-j}\mu_{j-i+N} - \sum_{j=i+1}^{N-1} d_{i-j+N}\mu_{j-i}) \\
&= \mu_i(d \odot \mu \odot \mu)_i + 2\rho^2(d \odot \mu)_0,
\end{aligned}$$

so that $\mathrm{Cov}(M_i, ((Z \odot M \odot M)_i)) = 2\rho^2(d \odot \mu)_0$. Returning to the notation of previous sections, if we can show that $(m \odot m_k)_0 > 0$ for all k, we can argue that we can achieve tighter behaviour by dropping the independence heuristic.

Consider lower bounding $(m \odot m_k)_0$. We have that

$$\tau(m \odot m_k) = \tau(m) \cdot \tau(m_k) = \Delta^2 z \cdot z^{(k)},$$

where on the RHS $\cdot$ is the component-wise product. We therefore need to bound entries of the form $d(2x - dx^2)$ away from zero, for $d \in [a, b]$ and x generated via the Newton-Raphson procedure described above. If $xd \in [\alpha, 1]$ with $\alpha > 0$ then $d(2x - dx^2) \in [\alpha, 1]$, so if we can identify some $\alpha > 0$ for which we can guarantee $1 \leqslant x^{(k)}d \geqslant \alpha$ then we can say $x^{(K)}d \geqslant \alpha$ for all $K \geqslant k$ and we are done by induction. As argued in [43], the minimum of $dx^{(0)}$ is attained at both a and b, and so we can say that, for all $k \geqslant 0$,

$$1 \geqslant z_j z_j^{(0)} \geqslant a(T_0 + T_1 a)$$

so we can say that

$$\begin{aligned}
\frac{4}{\Delta^2}\mathrm{Cov}(M_i, (Z \odot M \odot M)_i) &= \frac{8\rho^2}{\Delta^2}(m \odot m_k)_0 \\
&= \frac{8\rho^2}{N}\sum_{j=0}^{N/2-1}(z \cdot z^{(k)}) + \overline{(z \cdot z^{(k)})} \\
&\geqslant 8\rho^2 a(T_0 + T_1 a),
\end{aligned}$$

and each iteration with $k \geqslant 1$ we may subtract the RHS from the variance derived when assuming full independence.

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition, real error.						
13	109	-22.68	-21.89	-22.54	-15.66	-21.93
14	219	-21.59	-20.95	-21.51	-14.14	-20.91
15	443	-20.50	-19.80	-20.48	-12.61	-19.88
Multiplication, real error.						
13	109	-22.26	-21.56	-21.74	-1.66	-20.91
14	219	-21.13	-20.12	-20.72	0.86	-19.89
15	443	-19.97	-18.62	-19.69	3.39	-18.86
Addition, complex error.						
13	109	-22.48	-21.92	-22.46	-15.66	-21.93
14	219	-21.39	-20.72	-21.43	-14.14	-20.91
15	443	-20.31	-19.70	-20.41	-12.61	-19.88
Multiplication, complex error.						
13	109	-23.17	-21.51	-21.17	-1.16	-20.42
14	219	-21.68	-19.92	-20.14	1.36	-19.39
15	443	-20.13	-18.72	-19.12	3.89	-18.37

Table 3: Average and maximum bits of error (either real or complex, as indicated) observed in the message space over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses, for $\alpha = 0.00001$ and $\Delta = 2^{40}$.

F Additional Experiments

F.1 Smaller values of α in the complex space

F.2 Smaller values of α in the ring

F.3 Fixed vector

The Small-S assumption will not hold in the situation of Heuristic 10, if, for example, m_1 and m_2 have fixed constant large components. In Table 5 we illustrate this experimentally by considering the encryption of a fixed constant vector. The experimental results for homomorphic multiplication should be contrasted with those in Table 1, which encrypt vectors with values chosen at random in $[0, 1]$. We see that for the fixed vector, the maximal noise observed is larger, and further from the average, than for the encryption of a random vector.

F.4 Larger Δ

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition noise.						
13	109	10.9	11.3	11.49	11.99	18.04
14	219	11.4	12.0	12.02	12.52	19.07
15	443	12.0	12.5	12.54	13.04	20.09
Multiplication noise.						
13	109	17.3	17.7	18.78	25.99	31.39
14	219	18.4	18.8	19.81	27.52	33.41
15	443	19.4	19.8	20.83	29.04	35.44

Table 4: Average and maximum bits of noise observed in the ring over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses, for $\alpha = 0.00001$ and $\Delta = 2^{40}$.

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition noise.						
13	109	10.88	11.23	11.40	11.90	17.95
14	219	11.44	11.76	11.93	12.43	18.98
15	443	11.99	12.36	12.45	12.95	20.01
Multiplication noise.						
13	109	17.12	18.48	18.69	25.90	31.30
14	219	18.09	19.61	19.72	27.43	33.33
15	443	19.12	20.79	20.75	28.95	35.35

Table 5: Average and maximum bits of noise observed in the ring over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses. Fixed vectors, for $\alpha = 0.0001$ and $\Delta = 2^{40}$.

$\log(N)$	$\log(q)$	Average	Maximum	CLT	WCR	CE
Addition, real error.						
13	109	-32.70	-31.97	-32.63	-25.76	-32.02
14	219	-31.58	-30.98	-31.60	-24.23	-30.99
15	443	-30.49	-29.80	-30.57	-22.70	-29.97
Multiplication, real error.						
13	109	-32.27	-31.49	-31.84	-11.76	-31.00
14	219	-31.11	-30.26	-30.81	-9.23	-29.97
15	443	-29.96	-28.68	-29.78	-6.70	-28.94
Addition, complex error.						
13	109	-32.49	-31.84	-32.55	-25.76	-32.02
14	219	-31.39	-30.73	-31.52	-24.23	-30.99
15	443	-30.32	-29.75	-30.49	-22.70	-29.51
Multiplication, complex error.						
13	109	-33.19	-31.64	-31.26	-11.26	-30.50
14	219	-31.70	-30.17	-30.23	-8.73	-29.48
15	443	-30.20	-28.68	-29.20	-6.20	-27.99

Table 6: Average and maximum bits of error (either real or complex, as indicated) observed in the message space over 1000 trials in HEAAN compared with noise predicted by the CLT, WCR and CE noise analyses, for $\alpha = 0.0001$ and $\Delta = 2^{50}$.