Differential Privacy for Free? Harnessing the Noise in Approximate Homomorphic Encryption

Tabitha Ogilvie

Intel Labs
tabitha.ogilvie@intel.com

Abstract. Homomorphic Encryption (HE) is a type of cryptography that allows computing on encrypted data, enabling computation on sensitive data to be outsourced securely. Many popular HE schemes rely on noise for their security. On the other hand, Differential Privacy (DP) seeks to guarantee the privacy of data subjects by obscuring any one individual's contribution to an output. Many mechanisms for achieving DP involve adding appropriate noise. In this work, we investigate the extent to which the noise native to Homomorphic Encryption can provide Differential Privacy “for free".

We identify the dependence of HE noise on the underlying data as a critical barrier to privacy, and derive new results on the Differential Privacy under this constraint. We apply these ideas to a proof of concept HE application, ridge regression training using gradient descent, and are able to achieve privacy budgets of $\varepsilon \approx 2$ after 50 iterations.

Keywords: Differential Privacy · Homomorphic Encryption · Machine Learning

1 Introduction

Homomorphic Encryption (HE) is a technology which allows computing on encrypted data without knowing the decryption key. Efficient and secure Homomorphic Encryption has the potential to make many standard scenarios private, including outsourced computation, database queries, machine learning inference, and many more. On the other hand, Differential Privacy is a technique for ensuring the privacy of each data contributor while outputting some statistic or function or a database. At a high level, the goal of both technologies is the same – *keep the data secret.*

Standard techniques for achieving this secrecy are also superficially similar between the two technologies: namely, both introduce noise. Many popular HE schemes [11,12,37,20,22] use the Learning with Errors (LWE) [72] or Ring Learning with Errors (RLWE) [62] problem, which involve adding noise during encryption for security. Differential Privacy (DP) is also achieved by adding noise, but in this context the noise serves to obscure the contribution of any one individual [35,36,15,2].

In this work, we investigate the following question: *does the noise in Homomorphic Encryption give Differential Privacy "for free"?*

At first glance, this may seem unlikely for several reasons. Firstly, as noise is only introduced in HE as part of encryption, standard schemes will remove the noise during decryption. Secondly, to formally quantify Differential Privacy guarantees, we must be able to specify how the noise is distributed, and not just an upper bound, which is the typical output of an HE noise analysis. Lastly, as we only introduce noise in HE for security, and do not want it to corrupt the result of our computation, typically the noise is small relative to the message, and so cannot be sufficient to guarantee privacy.

However, there are contexts where none of these difficulties apply. For the first, the CKKS, or HEAAN, scheme [20] departs from traditional constructions in that noise is *not* removed during decryption, and is retained in the least significant bits of the final output. For this reason, it is sometimes called "Approximate Homomorphic Encryption". For the second, a recent work [26] analyzed noise growth when computing over CKKS, and argues that the noise can be modelled as normally distributed throughout the course of an algorithm. For the final challenge, we make the following observation: when evaluating a very complex or deep algorithm using CKKS, noise can start to overwhelm the message, becoming as large as is required to guarantee privacy. It is this intuition, of noise growing over the course of an application, which we explore in the case study in this paper.

We therefore have HE applications where noise is never removed, grows over the course of an algorithm, and is normally distributed, which suggests these HE applications can achieve DP without further processing; in other words, for free. We just need the noise to grow large enough to mask the contribution of any one individual.

But there is an additional complication. If we examine the results of [26], we find the variance after a multiplication depends on the messages being multiplied. In other words, changing the entries in a database will not only change the "true" output of the algorithm, but the variance of the noise we will add to it. We derive novel results on the impact this has on Differential Privacy, and find that, at least for our case study, this message dependent variance is more important than noise growth in preventing us from achieving Differential Privacy for free.

Building on this result, we are able to formally quantify the Differential Privacy of our case study, and show that, for the proposed parameters, we can achieve privacy budgets of $\varepsilon \approx 2$. In contexts where this is an acceptable level of privacy, we therefore find that Homomorphic Encryption gives Differential Privacy for free.

1.1 Contributions

In this work, we identify a connection between noise in Homomorphic Encryption and Differential Privacy, and explore this correspondence when Homomorphic Encryption noise is treated as a database dependent output perturbation. We present novel results on the Differential Privacy guarantees of adding database dependent noise in both the one- and multi-dimensional case. We believe this is the first time database dependent noise has been analyzed in the Differential Privacy literature, and may be of independent interest.

We explore these results with a proof of concept case study from the HE literature: ridge regression training using gradient descent [66]. Two factors constrain the choice of case study. Firstly, we require the noise growth analysis from [26], which does not extend to powers beyond the square, so any polynomial evaluation must be at most quadratic[1]; quadratic circuits are however not required in general, and in principle our techniques extend to higher degrees. Secondly, we are guided by the intuition that the noise growth over high depth algorithms is what will guarantee privacy, and so require an application of high multiplicative complexity. In the HE literature, the only application we are aware of that meets both of these criteria is the ridge regression training introduced in [66]. For this case study, we provide a blueprint for how to derive all parameters relevant to Differential Privacy, and present findings on noise growth, message dependence, and finally privacy.

1.2 Related Work

Homomorphic Encryption Since Gentry's breakthrough construction of a fully homomorphic encryption (FHE) scheme based on lattices [40], many schemes have been proposed following similar principles [12,11,22,20,41,34,74]. Development of HE solutions has also been aided by an ecosystem of software tools, including libraries [43,76,67,58,6,23], compilers [30,31,29,42] and toolkits [1,3].

Due to its ability to handle approximate real numbers, CKKS has been applied to various Machine Learning problems, including ridge and logistic regression training [66,53,54,13], neural network inference [16,49,8,7], federated learning [64], and decision tree training [4]. CKKS has also been successfully applied to problems with discrete message spaces [33].

In this work, we will restrict our attention to CKKS as originally presented [20]. However, there have been many works improving and extending the functionality of CKKS, including a Residue Number System (RNS) variant [18], noise reduction techniques [33,52], and hardware acceleration [9,73,75].

Differential Privacy For an in depth overview of the core concepts and literature for differentially private machine learning, we refer to [47]. There, the authors give a helpful representation of the different stages in an algorithm where we can add noise to achieve Differential Privacy, which we have reproduced in Algorithm 1. The inclusion of “Input Perturbation" is our own. We present relevant works following this taxonomy, cataloguing output, gradient, objective, and input perturbations.

Output perturbation methods involve analyzing how much the algorithm's output can change from one database to another, and adding enough noise to

[1] More practical use cases might include [54,53], which use degree 3,5, or 7 approximations to the sigmoid function.

Algorithm 1 Mechanisms for Achieving Differential Privacy in Machine Learning Training (based on Algorithm 1 of [47]).

```
Input: Data $X$
Output: A result $\beta$
$\beta \leftarrow 0$
#1 Input perturbation: add noise to data, $X \leftarrow X + e$.
#2 Objective perturbation: add noise to the loss function, $J(\beta, X) \leftarrow J(\beta, X) + e$.
for $t = 1, ..., T$ do
    #3 Gradient perturbation: add noise to the gradient updates,
    $\nabla J(\beta, X) \leftarrow \nabla J(\beta, X) + e$.
    $\beta \leftarrow \beta - \alpha \nabla J(\beta, X)$
end for
#4 Output perturbation: add noise to the final result $\beta \leftarrow \beta + e$
return $\beta$
```

mask this difference. This technique is pursued when the training algorithm uses a finite number of updates in [82,83], and the algorithm outputs a global minimum in [61,15,48].

For deep learning tasks, when loss functions may be non-convex or non-smooth, it can be difficult to quantify and bound how sensitive the training algorithm is to a single data point in general. Instead, perturbing gradients during training serves to obscure the contribution, and so preserve the privacy, of any one individual each iteration, and then a composition theorem may be used to find the total privacy loss over the course of the algorithm. This approach is adopted in many works, including [78,2,77,48,68].

Objective perturbations broadly fall into two categories. The first seeks to ensure the objective function itself can be released without compromising privacy, while the second seeks to ensure that the minimizer of the objective function can be released without compromising privacy. In other words, the first approach achieves Differential Privacy before training begins, while the second achieves Differential Privacy for the returned value only. The first approach is also called a “Functional Mechanism", and is explored in [84,32], while for the second we refer to [51,15,46].

Input perturbation is the least established of these methods. The authors of [38] use the terminology to refer to perturbing each data contributor’s contribution to the loss function coefficients, and so in our framework is better categorized as an objective perturbation. In [39], the authors achieve a differentially private chi-squared test by perturbing each of the frequencies at the outset, so that we have differential privacy before the algorithm starts. In [50,79], the authors observe that input perturbations in turn perturb the gradients, and so argue that it is sufficient to lower bound this induced perturbation with a known threshold from the gradient perturbation literature. However, this reasoning fails to consider that the width of this induced perturbation will depend on the data itself for almost all loss functions, and so constitutes an additional leakage which needs to be analyzed.

DP and HE The authors of [69] use both HE and DP in their work, training models homomorphically and then adding a perturbation after decryption to give differential privacy. The protocol proposed in [71] processes sensitive queries homomorphically, and then adds an optional perturbation to achieve Differential Privacy, depending on the clearance of the user. In [80], a two-party protocol is constructed where the input data is differentially private while the output is calculated homomorphically.

Differential Privacy is also combined with Homomorphic Encryption in [60], where DP is used as a tool to harden the security of CKKS in certain security models [59]. Here, the authors' goal is to give privacy to the HE noise itself. Due to the security model considered in [60], and in particular that the adversary there observes the input data, we consider our work orthogonal, but future work may wish to examine whether the modification to the CKKS scheme proposed also gives Differential Privacy "for free". We additionally remark that the issues the authors identify in Section 5, which considers Dynamic Error Estimation, seem analogous to the message dependent variance problem we identify in this work.

1.3 Paper Outline

In Section 2 we give the necessary background material. In Section 3 we present a formal analysis of the Differential Privacy guarantees of algorithms evaluated homomorphically. In Section 4 we apply our results to Ridge Regression to provide a proof of concept analysis, including experimental outcomes in Section 4.2. We conclude and outline future research directions in sections 5 and 6. Omitted details and proofs are provided in the appendices.

2 Background

2.1 Basic Notation

We will use $\log(\cdot)$, for the base-2 logarithm, and $\ln(\cdot)$ for the natural logarithm. For a vector v, we will write v_j to denote the j^{th} component of v, and $||v||$ to denote the 2-norm. For a polynomial m, we will write $||m||$ to denote the 2-norm of the vector of its coefficients. If the coefficients are modulo q, we consider their absolute value to be their representative in $[-q/2, q/2]$. We use $v \leftarrow D$ to denote sampling v according to the distribution D, and $N(\mu, \Sigma)$ for the multivariate Gaussian distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$. We extend this notation to the 1-dimensional case, where we will use $N(\mu, \sigma^2)$ for the univariate Gaussian with mean μ and variance σ^2. We will write $\chi^2_d(\nu)$ for the non-central chi-squared distribution with d degrees of freedom and non-centrality parameter ν. In more detail, if we have for $X_i \sim N(\mu_i, 1)$ for $i = 1, ..., d$ then $Y = \sum_{i=1}^d X_i^2$ has distribution $\chi^2_d(\nu)$, where $\nu = \sum_{i=1}^d \mu_i^2$.

We will use the following standard tail bound for Gaussian distributions.

Lemma 1. *Let $X \sim \mathcal{N}(0, \rho^2)$. Then for any $t > 0$ we have*

$$\Pr[X > t] \le \frac{\rho}{\sqrt{2\pi} t} \exp\left(\frac{-t^2}{2\rho^2}\right).$$

Rearranging for t gives the following corollary.

Corollary 1. *Let $X \sim \mathcal{N}(0, \rho^2)$. Then if $t > 0$ and*

$$\ln(t/\rho) + t^2/2\rho^2 > \ln\left(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}\right)$$

we have $\Pr[X > t] < \delta/2$.

As we will need to quote results from two literatures, which frequently use the same symbol for different concepts, we additionally provide a table of the notation we follow in this paper in Section 2.6.

2.2 CKKS

Our work focuses on the CKKS scheme [20]. We present a description of the original scheme in Appendix A.1. Typically, HE schemes [11,37,62,41] maintain a strict separation between message bits and noise bits, enabling the noise bits to be efficiently removed during decryption. On the other hand, the authors of CKKS argue that in some contexts this is unnecessary, and it is sufficient to allow the noise to interact with the lower bits of the message, controlling the noise growth via a Rescale() procedure. This relaxation is compared to floating point precision errors, which are tolerable in some contexts. This relaxation also enables efficiently encrypting and homomorphically processing high precision real numbers.

For our purposes, we will only need to consider a few key features of the CKKS scheme: CKKS is parametrised by a power of 2 polynomial modulus N, as well as a (typically power of 2) precision parameter Δ. The scheme has *message space* given by $\mathbb{C}^{N/2}$, where $\mathbb{C}$ is the set of complex numbers, and *plaintext space* given by the ring $\mathbb{Z}[X]/(X^N+1)$. Data is *encoded* from the native message space into the plaintext space before encryption. For a more comprehensive exploration of the CKKS scheme, we refer to [20,70,52,10,17]

The approximate nature of CKKS requires additional security considerations in many scenarios. Although this is beyond the scope of this work, it should be taken into consideration whenever deploying CKKS based solutions. Further details can be found in [59,60,19].

CKKS Noise Growth Understanding and bounding noise growth is critical to applications of HE. A typical approach[2] [14,27] involves arguing that “fresh"

[2] For TFHE and related schemes [22,24], a so-called “average case", or variance tracking, approach is more common – see for example [21,25,56].

sources of noise are normally distributed, and so uses a Gaussian tail bound to give a high probability bound B, which can be used to bound the noise in plaintexts. Together with the triangle inequality, this gives high probability bounds for the noise resulting from homomorphic computation. For example, if we add two ciphertexts with noise bounds B_1 and B_2 respectively, we can bound the noise of the resulting ciphertext by $B_1 + B_2$.

A different approach is explored in [65,28,26], where it is argued that, due the Central Limit Theorem, noise remains Gaussian throughout the entire computation. This approach generates much tighter noise bounds. For example, in [26], the authors show that if we multiply encryptions of plaintexts m_1 and m_2 with noise variances σ_1^2 and σ_2^2 respectively, the resulting ciphertext has Gaussian noise in the ring with variance

$$\sigma^2 = N\sigma_1^2\sigma_2^2 + \sigma_1^2 \left|\left|m_2\right|\right|_2^2 + \sigma_2^2 \left|\left|m_1\right|\right|_2^2 .$$

In order to translate a variance σ^2 in the plaintext space to a variance ρ^2 in $\mathbb{R}^{N/2}$, we have to understand the impact of decoding on a Gaussian distribution. As shown in [26], and observed in other contexts [63,70], if we restrict ourselves to real messages, we have that the noise in the message space is also normally distributed, with variance $\rho^2 = \frac{N}{2\Delta^2}\sigma^2$, where Δ is the decoding scale factor. We give full details of how to update the ring variance after various homomorphic operations in Appendix A.2 using the analysis of [26].

2.3 Differential Privacy

We draw extensively from [35,47]. Differential Privacy seeks to formally quantify and minimise the extent to which an algorithm's output depends on the input of any one individual to the dataset. To this end, we will say two databases $\mathcal{D}$ and $\mathcal{D}'$ are *neighboring* or *adjacent* if they differ on at most one row. We define Differential Privacy as follows:

Definition 1 ((ε, δ)-Differential Privacy). *A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ) differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all neighboring databases $\mathcal{D}$ and $\mathcal{D}'$,*

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{S}] \leqslant \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{S}] + \delta,$$

where the probability is over the randomness of $\mathcal{M}$.

We can interpret this as any outcome $\mathcal{S}$ does not become much more or less likely by modifying a single entry, except with a small tolerance probability δ. There are various alternative notions of Differential Privacy in the literature: we refer to [47] for an overview.

Observe that the algorithm $\mathcal{M}$ can take many forms, from individual database entries, to summary statistics, to the outcome of a machine learning training algorithm with training set $\mathcal{D}$, and in each case the Differential Privacy considerations will be different. It will be necessary to understand the extent to which an algorithm can be influenced by a single entry, which is captured in the following definition.

Definition 2 **(l_p Sensitivity).** *The l_p sensitivity of an algorithm $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:*

$$\delta_f^{(p)} = \max_{\substack{\mathcal{D},\mathcal{D}' \in \mathbb{N}^{|\mathcal{X}|} \\ adjacent}} ||f(\mathcal{D}) - f(\mathcal{D}')||_p$$

In this work we will only consider $p = 2$, and so will omit the (p)-superscript.

Frequently Differential Privacy is achieved via adding random noise. One such method is the *Gaussian Mechanism*, defined as follows.

Definition 3 (Gaussian Mechanism). *The Gaussian Mechanism with parameter ρ^2 adds zero-mean Gaussian noise with variance ρ^2 to each of the d coordinates of an algorithm's output.*

The relationship between (ε, δ)-Differential Privacy, sensitivity, and the Gaussian Mechanism is a standard result in the literature, but we will defer it's presentation to Section 3, as well as recapping the proof in Appendix B, in order to properly motivate our own analysis.

Finally, we repeat the *post-processing* principle, which guarantees that if an algorithm is (ε, δ)-differentially private, any further data-independent computation preserves this privacy.

Proposition 1 (Proposition 2.1 of [35]). *Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomized algorithm that is (ε, δ)-differentially private. Let $f : R \to R'$ be an arbitrary mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R'$ is (ε, δ)-differentially private.*

The parameter ε is sometimes referred to as a *privacy budget*, and an acceptable value will depend on the context. Indeed, in [47], it is observed that for simpler applications, a privacy budget in the range $(0, 1)$ is sufficient to have a performant algorithm. On the other hand, for more complex learning tasks, a typical privacy budget can be around 10. As observed by the authors of [47] however, in standard applications, the privacy implications of budgets this high may be unacceptable. Indeed, if $\varepsilon = 10$, an outcome can go from almost impossible to almost certain by changing just one value in the dataset, completely compromising the privacy of the individual added.

2.4 Update Rules

In the analysis of sensitivity for our case study, we will use the terminology and notation of [82] and [44] for clarity, namely *update rules*. An update rule is simply a function $G : \Omega \to \Omega$ for some arbitrary set Ω. For gradient descent, $\Omega = \mathbb{R}^d$, and we would have an update rule of the following form

$$G(\beta) = \beta - \alpha \nabla J(\beta, \mathcal{D}).$$

where J is our cost function, ∇J its derivative, and α the learning rate. In particular, the update rule *G depends on the database.* The authors of [82,44] use this terminology to analyze, respectively, the stability and differential privacy of Permutation Based Stochastic Gradient Descent (PSGD). PSGD differs from

Gradient Descent in that it only uses one randomly selected training example per update.

If we have an iterative procedure with updates $G_1, G_2, ..., G_T$, and a fixed starting point $\beta^{(0)}$, the full algorithm is therefore equivalent to the composition $G_T \circ G_{T-1} \circ ... \circ G_1$ applied to $\beta^{(0)}$.

2.5 Ridge Regression Case Study

We will explore our results with an application to ridge regression training using gradient descent. This algorithm was originally implemented using CKKS in [66].

As an additional assumption, we will assume databases satisfy each $|y_i|, |x_{ij}| \leq 1$ which can be achieved via normalization.

Ridge regression takes n entries $x_i \in \mathbb{R}^d$ with n labels y_i, and seeks weights $\beta_1, \beta_2, ...\beta_d$ such that[3]:

$$y_i \approx \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_d x_{id}$$

for each i. Ridge regression additionally seeks to prevent overfitting by penalising large values of β_j with $l2$ regularisation. The cost function for ridge regression is of the form:

$$J(\beta, \mathcal{D}) = \frac{1}{2}\lambda \, ||\beta||^2 + \frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta \cdot x_i)^2, \tag{1}$$

where the parameter λ is the *regularization parameter*, and determines the degree of penalization of large coefficients. We will sometimes omit the second argument $\mathcal{D}$.

We will be considering minimising (1) using gradient descent. We will initialize $\beta = 0$, and then iteratively update the parameters via

$$\beta \leftarrow \beta - \alpha \nabla J(\beta), \tag{2}$$

where α is the *learning rate*, which possibly changes from iteration to iteration. Differentiating our cost function, this is equivalent to updating each weight via

$$\beta_j \leftarrow (1 - \lambda\alpha)\beta_j + \frac{\alpha}{n}\sum_{i=1}^{n} x_{ij}\,(y_i - \beta \cdot x_i) \tag{3}$$

Without additional restrictions, the unbounded gradients of the cost function make it impossible to bound the sensitivity of ridge regression parameters after a fixed number of updates. We therefore assume the following additional heuristic.

Heuristic 2.51 *The learning rate is such that $J(\beta, \mathcal{D})$ decreases each iteration.*

Observe that such a learning rate can be chosen whenever the cost function is Lipschitz continuous, which is not the case here. In our experiments, we will use a decaying learning rate, which perhaps makes this heuristic more reasonable.

[3] In this work, for simplicity we do not train a constant weight β_0.

For our case study, as a consequence, we will only be able to claim differential privacy over the subset of databases for which this heuristic holds. If the heuristic fails, it may be possible to distinguish which database was used during training.

From the heuristic, we can derive the following[4].

Lemma 2. *Assuming Heuristic 2.51, at each iteration we have that* β *satisfies* $||\beta||_2 \leq \frac{1}{\sqrt{\lambda}}$.

Proof. As the cost decreases each iteration, letting $\beta^{(k)}$ be the output of the k^{th} update, so that $\beta^{(0)} = 0$, and recalling each $|y_i| \leq 1$,

$$J(\beta^{(k)}, \mathcal{D}) \leq J(\beta^{(0)}, \mathcal{D}) = \frac{1}{2n} \sum_{i=1}^{n} y_i^2 \leq \frac{1}{2},$$

while on the other hand

$$\lambda \left|\left|\beta^{(k)}\right|\right|_2^2 \leq \lambda \left|\left|\beta^{(k)}\right|\right|_2^2 + \frac{1}{n} \sum_{i=1}^{n} (y_i - \beta^{(k)} \cdot x_i)^2 = 2J(\beta^{(k)}, \mathcal{D}),$$

so we can conclude $\left|\left|\beta^{(k)}\right|\right|_2^2 \leq 1/\lambda$.

2.6 Notation Key

We present a guide to the notation we will use throughout this paper.

HE	N	polynomial modulus
	Δ	precision parameter
	$\mathsf{ct}.X$	a ciphertext encrypting X
	σ_X^2	ring variance when calculating X homomorphically
	ρ_X^2	real variance when calculating X homomorphically
DP	ε	privacy budget
	δ	failure probability
	δ_f	sensitivity of the function f
Database	n	number of database rows
	d	number of database columns
	x_{ij}	feature j of i^{th} entry
	y_i	label of i^{th} entry
ML	β	vector of model weights
	λ	regularization coefficient
	α	learning rate

Table 1. A guide to the notation we follow in this paper.

[4] This bound applies unconditionally to the *minimum* of the cost function – see [61]. However, in our case study we will only evaluate a fixed number of iterations of gradient descent, and so cannot assume we converge to the minimum.

As we are unifying concepts from various fields, some of these depart from standard notation. Most prominently, in Differential Privacy, it is typical to use Δ to denote sensitivity, whereas in CKKS, this symbol is used for the precision parameter. For the noise analysis, we consistently use σ^2 for noise variances in the plaintext space (ring), and ρ^2 for the variance of the real error.

3 Differential Privacy Analysis

Let us now consider how best to analyze the Differential Privacy guarantees of an algorithm evaluated homomorphically using the CKKS scheme. Let us first note that, since all intermediary stages are encrypted, it is sufficient to consider the privacy of the final output. Secondly, we can use the results in [26] to argue that the final output follows a normal distribution of the following form

$$\beta + N(0, \rho^2) \tag{4}$$

where β is the “true" output of the algorithm. Thus an initial approach may attempt to ensure ρ is large enough to mask the difference $\beta - \beta'$ over adjacent databases. For this approach, we have the following classical result on the privacy guarantee of the Gaussian Mechanism [35,51]. The proof is recapped in Appendix B.1 for completeness.

Theorem 1. *Let $\varepsilon \in (0,1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism is (ε, δ)-differentially private whenever $\rho \geq c\delta_f/\varepsilon$, where δ_f is the sensitivity.*

Therefore, if we bound $||\beta - \beta'||$ at iteration k by δ_k, and have the variance at iteration k is (at least) ρ^2, then if we have $\rho^2 > 2\ln(1.25/\delta)\delta_k/\varepsilon$ it may seem we can argue we have (ε, δ) Differential Privacy at the k^{th} iteration[5]. We present experimental results for this approach applied to our case study in 4.2.

However, let's look more closely at how we update the variance after a multiplication. If we multiply encryptions of plaintexts m_1 and m_2 with ring noise variances σ_1^2 and σ_2^2 respectively, the resulting encryption has ring noise with variance:

$$N\sigma_1^2\sigma_2^2 + \sigma_1^2 \left|\left|m_2\right|\right|^2 + \sigma_2^2 \left|\left|m_1\right|\right|^2 . \tag{5}$$

Therefore, the variance of an algorithm's output when evaluated homomorphically using CKKS *is dependent on the input data*. Therefore, our situation is more accurately modelled as outputs from the distribution

$$\beta_{\mathcal{D}} + N\left(0, \rho_{\mathcal{D}}^2\right) \tag{6}$$

where mean *and* variance depend on the underlying database. We therefore need to properly quantify the impact of this additional database dependency on Differential Privacy. This is accomplished in the one dimensional case by the following theorem.

[5] In the multivariate case, we may have the variance ρ_i^2 differs from component to component. Here we would say instead $\min \rho_i^2 > 2\ln(1.25/\delta)\delta_k/\varepsilon$.

Theorem 2. *Suppose we use a Gaussian mechanism with variance $\rho_{\mathcal{D}}^2$ dependent on the underlying database. Then the resulting mechanism is (ε, δ) differential private if $D = 2\ln\left(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}\right) > 1$ and*

$$\varepsilon > T^2K\sqrt{D} + \frac{1}{2}T^2K^2 + \frac{1}{2}(T^2-1)D + \ln T$$

where $\max \frac{\rho_{\mathcal{D}}}{\rho_{\mathcal{D}'}} \leq T$*,* $\max \frac{|\beta_{\mathcal{D}} - \beta_{\mathcal{D}'}|}{\rho_{\mathcal{D}}} \leq K$*, where the maximums are taken over adjacent databases.*

Proof. Fix adjacent databases $\mathcal{D}$ and $\mathcal{D}'$, and suppose the respective mechanisms are given by

$$\mathcal{A} \sim N(\beta, \rho^2) \quad \text{and} \quad \mathcal{A}' \sim N(\beta', \rho'^2)$$

Our proof strategy will follow the standard proof of Theorem 1: namely, to examine the ratio of probability density functions $\frac{f_{\mathcal{A}}(\alpha)}{f_{\mathcal{A}'}(\alpha)}$, and isolate a subset $R \subset \mathbb{R}$ with the following properties:

1. $\Pr[\mathcal{A} \in R] \geq 1 - \delta$
2. $\alpha \in R \implies e^{-\varepsilon} \leq \frac{f_{\mathcal{A}}(\alpha)}{f_{\mathcal{A}'}(\alpha)} \leq e^{\varepsilon}$.

So consider the ratio of the probability density functions at a fixed point $\alpha \in \mathbb{R}$:

$$\begin{aligned}\frac{f_A(\alpha)}{f_{A'}(\alpha)} &= \frac{\rho'}{\rho}\exp\left(\frac{1}{2\rho'^2}(\alpha-\beta')^2 - \frac{1}{2\rho^2}(\alpha-\beta)^2\right) \\ &= \frac{1}{\tau}\exp\left(\frac{1}{2}\tau^2\left(\frac{\alpha-\beta}{\rho} - \kappa\right)^2 - \frac{1}{2}\left(\frac{\alpha-\beta}{\rho}\right)^2\right)\end{aligned}$$

Where $\tau = \frac{\rho}{\rho'}$, and $\kappa = \frac{\beta'-\beta}{\rho}$. Observe that $\frac{1}{T} \leq \tau \leq T$, and $|\kappa| \leq K$.

Without loss of generality[6] let $\tau \geq 1$. Setting this ratio to lie in the range $[e^{-\varepsilon}, e^{\varepsilon}]$, and letting α follow the distribution of $\mathcal{A}$, and observing $\frac{\mathcal{A}-\beta}{\rho} \sim N(0,1)$, we are interested in the probability

$$-2\varepsilon + 2\ln\tau \leq \tau^2(Z-\kappa)^2 - Z^2 \leq 2\varepsilon + 2\ln\tau, \tag{7}$$

where $Z \sim N(0,1)$. We will show that if ε satisfies the condition of the theorem, this inequality is satisfied with probability at least $1-\delta$.

The random variable $\tau^2(Z-\kappa)^2 - Z^2 = (\tau^2-1)Z^2 - 2\kappa\tau^2 Z + \tau^2\kappa^2$ is a quadratic in Z, with minimum at $\left(\frac{\kappa\tau^2}{\tau^2-1}, -\frac{\tau^2\kappa^2}{\tau^2-1}\right)$. The upper bound in (7) is achieved at $z_1^{\pm}$, where

$$z_1^{\pm} = \frac{\kappa\tau^2}{\tau^2-1} \pm \frac{\sqrt{\tau^2\kappa^2 + (\tau^2-1)(2\varepsilon + 2\ln\tau)}}{\tau^2-1}$$

[6] Indeed, if instead $\tau < 1$, we can exchange $\mathcal{D}$ and $\mathcal{D}'$ and use the lower $e^{-\varepsilon}$ bound.

The lower bound in (7) is achieved whenever $-2\varepsilon + 2\ln\tau \geq -\frac{\tau^2\kappa^2}{\tau^2-1}$, at $z_2^\pm$ where

$$z_2^\pm = \frac{\kappa\tau^2}{\tau^2-1} \pm \frac{\sqrt{\tau^2\kappa^2 - (\tau^2-1)(2\varepsilon - 2\ln\tau)}}{\tau^2-1}$$

Our inequality is met therefore whenever $Z \in [z_1^-, z_2^-] \cup [z_2^+, z_1^+]$, or in the case the lower equality is never met, $[z_1^-, z_1^+]$. Consider the case $\kappa \geq 0$: we will show the probability $Z \in [z_1^-, z_2^-] \geq 1-\delta$, from which we will be able to conclude the proof. The case $\kappa < 0$ follows almost identically, instead considering $Z \in [z_1^+, z_2^+]$.

We first can show that $z_1^- \leq -1$, since this is true if and only if

$$\varepsilon \geq \kappa\tau^2 + \frac{1}{2}\kappa^2\tau^2 + \frac{1}{2}(\tau^2-1) - \ln\tau,$$

which follows from our condition on ε, and the additional requirement that $D \geq 1$.

So we can use the tail bound (1) as follows:

$$\Pr[Z < z_1^-] = \Pr[Z > -z_1^-] \leq \frac{1}{\sqrt{2\pi}(-z_1^-)} \exp\left(-\frac{(z_1^-)^2}{2}\right) \tag{8}$$

$$\leq \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(z_1^-)^2}{2}\right) \tag{9}$$

To bound this tail by $\delta/2$, we need to show

$$-z_1^- \geq \sqrt{2\ln\left(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}\right)} =: \sqrt{D}.$$

But this is true if and only if

$$\varepsilon \geq \kappa\tau^2\sqrt{D} + \frac{1}{2}\kappa^2\tau^2 + \frac{1}{2}(\tau^2-1)D - \ln\tau,$$

which is guaranteed by our condition.

We now turn to the upper tail bound. If $-2\varepsilon - 2\ln\tau$ is below the minimum of the quadratic, we only need to consider $\Pr[Z > z_1^+]$, which is less than $\Pr[Z < z_1^-] \leq \delta/2$ since $\kappa \geq 0$ implies $z_1^+ \geq -z_1^-$, and so we can say the probability of (7) is at least $1-\delta$.

If we instead have the minimum lies below $-2\varepsilon - 2\ln\tau$, we consider the probability Z exceeds z_2^-. We can again show $z_2^- \geq 1$ using the condition on ε and D, so that as above we need to show

$$z_2^+ \geq \sqrt{D},$$

which is true if and only if

$$\varepsilon \geq \tau^2\kappa\sqrt{D} - \frac{1}{2}\tau^2\kappa^2 - \frac{1}{2}D + \ln\tau,$$

which is again guaranteed by our condition.

To finish the proof, we let R be the following set

$$R := \{\alpha \in \mathbb{R} : -2\varepsilon + 2\ln\tau \leq \tau^2(\alpha-\kappa)^2 - \alpha^2 \leq 2\varepsilon + 2\ln\tau\}$$

We have shown that $\Pr[\mathcal{A} \in R] \geq 1-\delta$, and that $\alpha \in R \implies e^{-\varepsilon} \leq \frac{f_{\mathcal{D}}(\alpha)}{f_{\mathcal{D}'}(\alpha)} \leq e^{\varepsilon}$. Therefore,

$$\begin{aligned}
\Pr[\mathcal{A} \in S] &= \Pr[\mathcal{A} \in S \cap R] + \Pr[\mathcal{A} \in S \cap (\mathbb{R}\backslash R)] \\
&\leq \int_{\alpha \in S \cap R} f_{\mathcal{A}}(\alpha) d\alpha + \Pr[\mathcal{A} \notin R] \\
&\leq \int_{\alpha \in S \cap R} e^{\varepsilon} f_{\mathcal{A}'}(\alpha) d\alpha + \delta \\
&\leq e^{\varepsilon} \Pr[\mathcal{A}' \in S] + \delta
\end{aligned}$$

as required.

Remark 1. In the case $T = 1$, we might hope to recover the standard Gaussian Mechanism inequality, since this corresponds to the variance having no message dependence. And indeed, if we let $T = 1$, we have $\varepsilon > K\sqrt{D} + \frac{1}{2}K^2 = O\left(\sqrt{\ln(\frac{1}{\delta})}\delta_f/\rho\right)$, which is asymptotically the same as in the standard case.

In the multidimensional case, we instead prove the following.

Theorem 3. *Suppose we use the following Gaussian mechanism: for a database $\mathcal{D}$, we provide a sample from the distribution $N(\beta_{\mathcal{D}}, \Sigma_{\mathcal{D}})$, where $\beta_{\mathcal{D}} \in \mathbb{R}^d$, and $\Sigma_{\mathcal{D}}$ is a diagonal matrix with diagonal entries $\rho_{\mathcal{D},1}, ..., \rho_{\mathcal{D},d}$ which are dependent on the database $\mathcal{D}$. This mechanism is (ε, δ) differentially private if*

$$\varepsilon > \sqrt{(\frac{1}{2}d(T^2-1) + T^4K^2)D} + \frac{1}{2}T^2K^2 + \frac{1}{2}(T^2-1)(D+d) + d\ln T$$

where $D = 2\ln\frac{1}{\delta}$, $\max\frac{\rho_{\mathcal{D},i}}{\rho_{\mathcal{D}',i}} \leq T$ for all i, and

$$\max\left|\left|\left(\frac{\beta_{\mathcal{D},1} - \beta_{\mathcal{D}',1}}{\rho_{\mathcal{D},1}}, ..., \frac{\beta_{\mathcal{D},d} - \beta'_{\mathcal{D}',d}}{\rho_{\mathcal{D},d}}\right)\right|\right| \leq K,$$

where maximums are taken over adjacent databases $\mathcal{D}$ and $\mathcal{D}'$.

Proof. The proof is given in Appendix B.2.

Remark 2. We again have that the asymptotic behaviour when $T = 1$ is equal to that in the classical case. However, when $d = 1$ the multivariate ε is strictly larger than in the univariate case. This may be an artefact of our proof technique, and in particular that the tail bound we use for a noncentral chi-squared distribution is strictly looser than the tail bound we use for a Gaussian distribution.

The full version of this theorem provides the opportunity to treat different components of the algorithm output differently. This may be valuable if, for example, we want to release the evaluation of different functions which have distinct sensitivities or variance growth. For our case study we will not take advantage of this flexibility, and so give the following corollary for the homogeneous case.

Corollary 2. *The mechanism from Theorem 3 is (ε, δ) differentially private whenever*

$$\varepsilon > \sqrt{(\frac{1}{2}d(T^2-1)+T^4K^2)D} + \frac{1}{2}T^2K^2 + \frac{1}{2}(T^2-1)(D+d) + d\ln T,$$

where $D = 2\ln\frac{1}{\delta}$, $\max\frac{\rho_{\mathcal{D},i}}{\rho_{\mathcal{D}',i}} \leq T$ for all i and $\delta_f/\rho \leq K$, where δ_f is the sensitivity and we have $\rho_{\mathcal{D},i} \geq \rho$ for all i and all databases $\mathcal{D}$.

4 Case Study

Now that we have an analysis of the privacy guarantees of an abstract homomorphic algorithm, we make these ideas concrete by analysing ridge regression, and provide a “recipe" for how to use these ideas for other applications. As we are pursuing a gradient descent approach, we will be investigating the differential privacy guarantees after k iterations.

Looking at Corollary 2 and Theorem 1, we must analyze the following quantities:

- *Sensitivity.* Written δ_f, a bound on $||\beta - \beta'||$ at iteration k where β and β' are produced by neighboring databases. We provide a novel analysis in Section 4.1.
- *Noise Variance.* Written $\rho_{\mathcal{D},i}$, we must understand the variance of the noise on each β_i in the message space. We give an analysis in Appendix C.1, and upper and lower bounds in Appendix C.2. We believe this is the first “average case" analysis of the noise growth across a full CKKS algorithm. For the lower bounds, we introduce a density parameter c which corresponds to a lower bound on relevant message magnitudes.
- *Message Dependence.* Written T, we must bound the ratio of variances produced by neighboring databases. This is done in Appendix C.3.

We make essential use of the heuristics derived in [26], which to the best of our knowledge represents the state of the art for average case analysis of CKKS, and argue that deriving additional average case heuristics is beyond the scope of this work. As such, our noise analysis is specific to textbook CKKS, as this is what is analyzed in [26], and does not use SIMD packing, SIMD techiques, or rotations, as these are not treated by the authors of [26].

4.1 Sensitivity

To bound the difference $||\beta - \beta'||$, we will use a corollary to the following lemma.

Lemma 3. *Fix an arbitrary sequence of updates $G_1, ..., G_T : \Omega \to \Omega$ and another sequence $G'_1, ..., G'_T : \Omega \to \Omega$. Let $\beta_0 = \beta'_0$ be a fixed starting point in Ω and define $\delta_t = ||\beta_t - \beta'_t||$ where β_t, β'_t are defined recursively through*

$$\beta_t = G_t(\beta_{t-1}), \quad \beta'_k = G'_t(\beta'_{t-1})$$

Let $\mathcal{B} = \{\beta_0, ..., \beta_T, \beta'_0, ..., \beta'_T\}$ be the "update set". Then, if $\sup_{x \in \mathcal{B}} ||G_t(x) - G'_t(x)|| \leq \eta_t$ and either G_t or G'_t is L_t-Lipschitz, we have $\delta_t \leq \eta_t + L_t\delta_{t-1}$.

Proof. Consider update t. Without loss of generality, let G'_t be Lipschitz. Then

$$\begin{aligned}
\delta_t &= \left|\left|G_t(\beta_{t-1}) - G'_t(\beta'_{t-1})\right|\right| \\
&\leq ||G_t(\beta_{t-1}) - G'_t(\beta_{t-1})|| + \left|\left|G'_t(\beta_{t-1}) - G'_t(\beta'_{t-1})\right|\right| \\
&\leq \eta_t + L_t \left|\left|\beta_{t-1} - \beta'_{t-1}\right|\right| \\
&\leq \eta_t + L_t\delta_{t-1}
\end{aligned}$$

as required.

Corollary 3. *Let δ_k be the sensitivity of the k^{th} ridge regression update as described in Section 2.5. Then we have the recurrence relation*

$$\delta_0 = 0, \quad \delta_1 = \frac{2\alpha\sqrt{d}}{n}, \quad \delta_k \leq \frac{2\alpha}{n}\left(\sqrt{d} + \frac{d}{\sqrt{\lambda}}\right) + (|1-\lambda| + \alpha d)\delta_{k-1}.$$

Proof. Without loss of generality, say $\mathcal{D}$ and $\mathcal{D}'$ differ on the nth training sample. We let G_k and G'_k be the k^{th} gradient update with databases $\mathcal{D}$ and $\mathcal{D}'$, so that, using Eq. (3),

$$\begin{aligned}
G_k(\beta) &= (1-\lambda\alpha)\beta + \frac{\alpha}{n}\sum_{i=1}^{n} x_i(y_i - \beta \cdot x_i) + \frac{\alpha}{n}x_n(y_n - \beta \cdot x_n), \\
G'_k(\beta) &= (1-\lambda\alpha)\beta + \frac{\alpha}{n}\sum_{i=1}^{n} x_i(y_i - \beta \cdot x_i) + \frac{\alpha}{n}x'_n(y'_n - \beta \cdot x'_n).
\end{aligned}$$

Let us bound δ_1 directly.

$$\delta_1 = ||G_1(0) - G'_1(0)|| = \frac{\alpha}{n}\,||y_i x_i - y'_i x'_i|| \leq \frac{2\alpha\sqrt{d}}{n},$$

since $|y_i| \leq 1$ and $||x|| \leq \sqrt{d}$.

We will now show that $\sup_\beta ||G_k(\beta) - G'_k(\beta)|| \leq \frac{2\alpha}{n}(1 + \frac{1}{\sqrt{\lambda}})$ and G_k is $(|1-\lambda| + \alpha d)$-Lipschitz, so that the claim follows from Lemma 3.

For the first part, we have

$$\begin{aligned}
||G_k(\beta) - G'_k(\beta)|| &= \frac{\alpha}{n}\,||x_n(y_n - \beta \cdot x_n) - x'_n(y'_n - \beta \cdot x'_n)|| \\
&\leq \frac{\alpha}{n}\left(||x_n||\,(|y_n| + |\beta \cdot x_n|) + ||x'_n||\,(|y'_n| + |\beta \cdot x'_n|)\right) \\
&\leq \frac{2\alpha}{n}(\sqrt{d} + d\,||\beta||),
\end{aligned}$$

where the last inequality follows by Cauchy-Schwartz and that $||x_i|| \leq \sqrt{d}$. For the β in the update set, we have $||\beta|| \leq 1/\sqrt{\lambda}$ due to Heuristic 2.51 and Lemma 2.

For the Lipschitz condition,

$$\begin{aligned}
||G_k(\beta) - G_k(\beta')|| &= \left|\left|(1-\lambda)(\beta - \beta') + \frac{\alpha}{n}\sum_{i=1}^{n} x_i(x_i \cdot (\beta - \beta'))\right|\right| \\
&\leq |1-\lambda| \, ||\beta - \beta'|| + \frac{\alpha}{n}\sum_{i=1}^{n} ||x_i|| \, |x_i \cdot (\beta - \beta')| \\
&\leq (|1-\lambda| + \alpha d) \, ||\beta - \beta'||
\end{aligned}$$

as required.

4.2 Experiments

We now explore the Differential Privacy of our case study experimentally by simulating the noise growth with our heuristics. Since we see in [66] that this algorithm exhibits very slow convergence, we do not report on the accuracy of derived models – we will be exclusively interested in the privacy properties. Our primary goal in this section is to use this case study to explore the theory developed in Section 3.

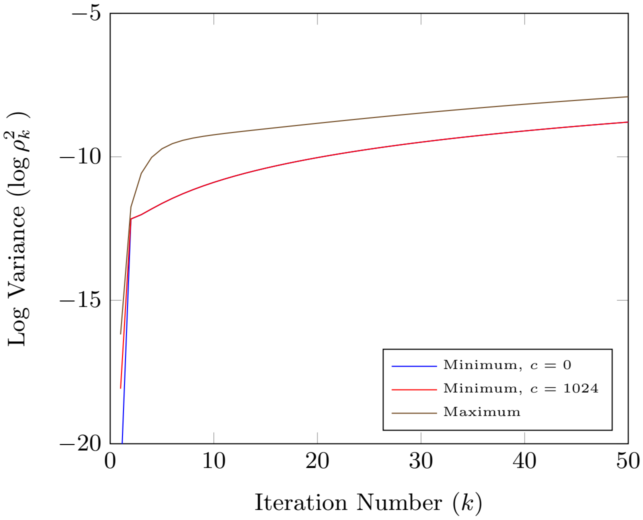

Fig. 1. Variance Growth with Iteration. $\log \varDelta = 20$.

Between the database, algorithm, HE, and DP parameters, we will not be able to explore the impact of all choices on Differential Privacy, and fix many throughout. For the database, we set $n = 4096$ and $d = 10$, while for the regression we let $\lambda = 1$, and use a dynamic learning rate $\alpha = 1/i$ at iteration i. On the privacy side, we fix the failure tolerance as $\delta = 1/n = 1/4096$. For the homomorphic encryption, we will fix all but the precision parameter Δ – we detail additional parameters in Appendix A.1.

We first examine how variance grows as we compute more and more iterations for a fixed precision. This is done in Fig. 1, where we plot the variance lower bound against the iteration number for two different density parameters c, as well as the upper bound. From this graph, we observe that, as expected, the variance grows with iteration number in both "best" and "worst" case for this value of Δ. We also observe that for these parameters the maximum and minimum values of the variance have a difference of around 1 bit by iteration 50. Surprisingly, we find that the density parameter does not impact how the variance lower bound grows. For this reason, we do not vary this parameter in the rest of our experiments, instead using $c = 0$ (the least restrictive value).

We additionally show how sensitivity changes with iteration in Fig. 2. Due to our parameter choices, in particular the decay of the learning rate α, we have that sensitivity increases at first, and then decreases. This shape is echoed in many of the following figures, as many quantities are dependent on sensitivity.

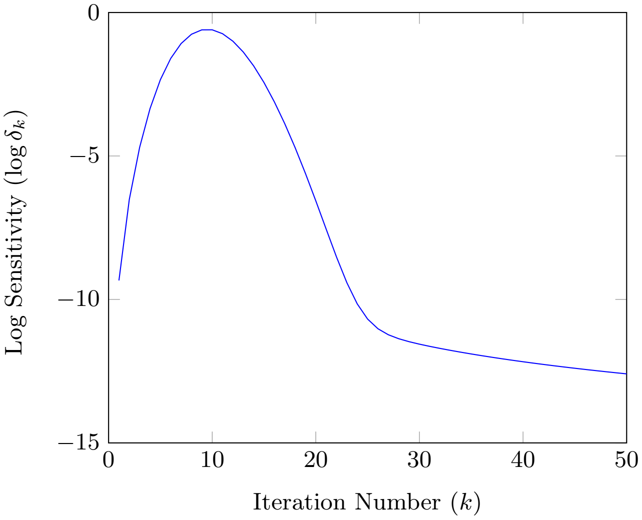

Fig. 2. Sensitivity Growth with Iteration.

"Variance Only" Approach Let's now investigate how much Differential Privacy we might expect if we only look at the variance, i.e., if we treat this as a standard Gaussian Mechanism. In more detail, we calculate a lower bound on the variance ρ^2, and use Theorem 1 to calculate the ε for which we have (ε, δ) privacy. We results are displayed in Fig. 3. These values for the precision parameter may appear quite low, with $\log \Delta$ in the range 30 to 40 being more common. However, this graph does suggest that a small parameter relaxation can give Differential Privacy, with all parameter sets reaching a privacy leakage less than 1 by iteration 30. As we might expect, we also find that, for this approach, the privacy leakage increases with $\log \Delta$. This corresponds to higher precision Δ giving smaller noise.

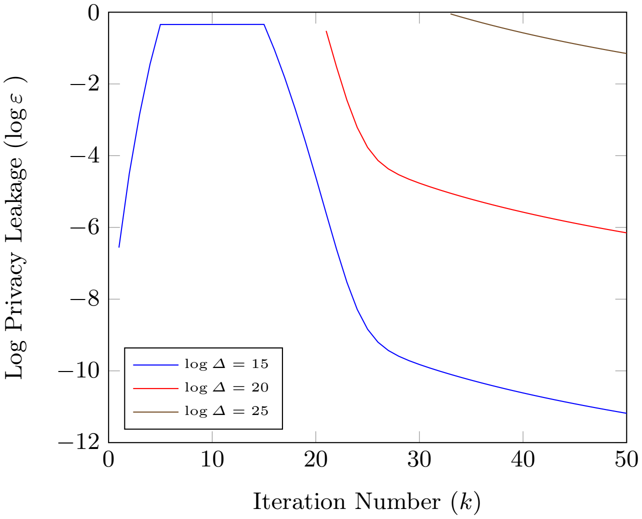

Fig. 3. ε Growth with Iteration for Varying Δ.

To give an additional insight into the relationship between Δ and privacy in this model, we plot $\log \Delta$ against the number of iterations required to achieve (ε, δ) privacy for a variety of privacy budgets ε in Fig. 4. Lines cutting off correspond to not being able to stay below the privacy budget for a given $\log \Delta$ within 100 iterations.

Full Approach Modelling Homomorphic Encryption noise as a simple Gaussian Mechanism suggests we can be cautiously optimistic about achieving Differential Privacy with only a small relaxation of the precision parameter. However, to accurately capture the situation we must consider the impact of message dependence. We plot how this varies over many iterations in Fig. 5. We first note that perhaps surprisingly, message dependence for this case study decreases with iteration, getting very close to the “no message dependence" by iteration 50 for $\log \Delta = 20$ and 25. We also have that the lines for $\log \Delta = 20$ and 25 almost coin-

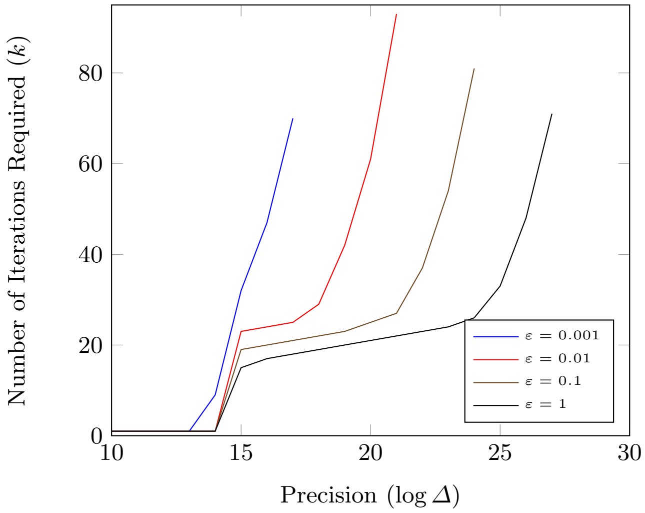

Fig. 4. Number of Iterations for (ε, δ) Privacy as a Function of Δ and ε.

cide completely. Another key observation is that message dependence decreases as $\log \Delta$ increases.

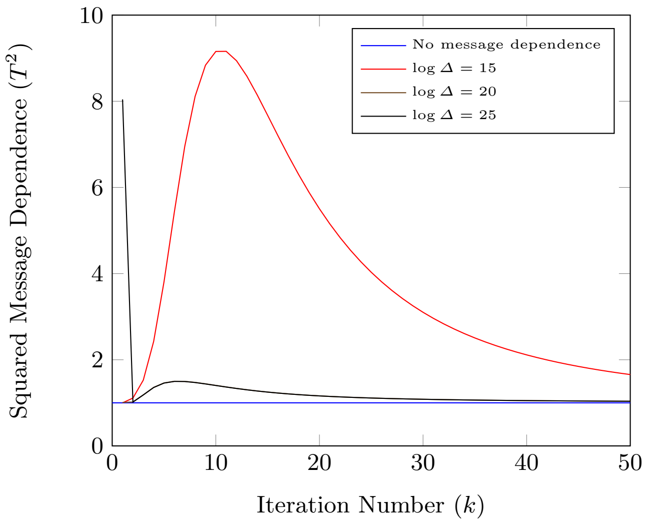

Fig. 5. Message Dependence Change with Iteration.

Lastly, we look at the full privacy leakage of this case study. We plot the privacy budget of each iteration, for a range of $\log \Delta$, following Corollary 2. We find the privacy leakage is much worse than we might expect, given the small leakage in the “variance only" approach (Fig. 3) and the low message dependence for later iterations (Fig. 5). In fact, for these parameters, ε never falls below 2.

Interestingly, we find the opposite relationship between Δ and privacy than was observed when only considering the variance – namely, past a certain number of iterations, higher Δ gives better privacy. This is because message dependence (T) dominates the sensitivity to variance ratio ($\delta_f/\rho \leq K$) in the calculation of ε.

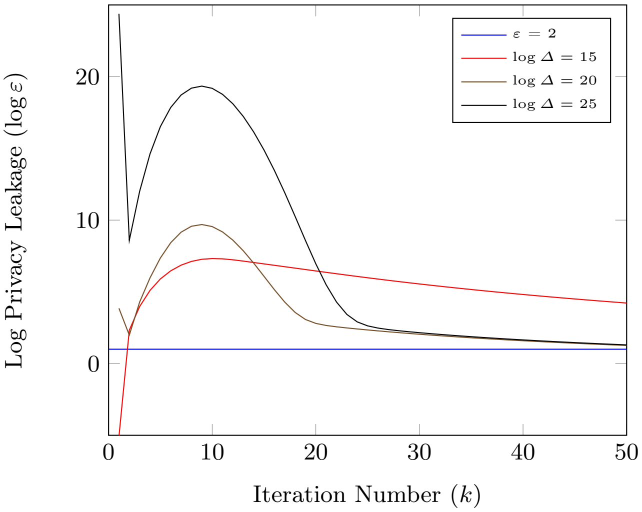

Fig. 6. Change in Log Privacy Leakage with Iteration $\log \varepsilon$.

We find that for this case study, privacy leakage falls as we perform more iterations. This observation contrasts with findings in the differential privacy literature, where early stopping is used to reduce privacy leakage [85,81].

We also probed our privacy budget formula ε to determine exactly which term(s) dominate, at least for this case study. We found that the $\frac{1}{2}dD(T^2 - 1)$ inside the square root has a large impact on the privacy leakage in later iterations. Interestingly, this term is not present in our result for the one dimensional case (Theorem 2), implying algorithms with one dimensional outputs may be able to achieve stronger privacy guarantees.

To understand the privacy-accuracy trade off, we can use the $6\sqrt{V}$ tail bound which is standard in HE noise analysis to give a worst case error on each coefficient of the output. We compute that for $\log \Delta = 25$, our heuristics give an error of at most -6 bits.

The number of iterations we suggest to give good privacy guarantees is much higher than that of the existing implementation. Indeed, the authors of [66] perform only 9 iterations at 40 bit precision, and 13 iterations at 30 bit precision.

However, our simplified algorithm actually has lower depth consumption[7]. We could therefore evaluate k iterations of this algorithm consuming a bit length of $(k+1) \cdot \log \Delta$.

5 Conclusion

In this work, we investigated the extent to which the noise growth in homomorphic encryption can provide differential privacy to the output. We identify that the major challenge is that the noise growth is dependent on the input messages, and so forms an additional privacy leakage. To this end, we derive new results on the differential privacy guarantees when adding message dependent noise. Using a case study, we find that a small relaxation of the precision parameter is enough to give reasonable privacy guarantees when we do not consider message dependence, achieving privacy budgets of $\varepsilon < 0.5$ within 50 iterations. However, when we properly account for message dependence, the privacy leakage is much higher, and we find that message dependence dominates noise growth, leading to noise budgets of $\varepsilon \approx 2$ after 50 iterations.

6 Further Work

Further Noise Analysis. We were limited in our case study by the state of the art in average case, or variance tracking, analysis for CKKS. In particular, to investigate the privacy guarantees of more practical algorithms, we would need to develop heuristics for degrees beyond squaring, and polynomial evaluations more generally. We would also need to understand the impact of packing and packing techniques on slotwise variance growth. Lastly, the heuristics we use apply only to the textbook version of CKKS, whereas most implementations will use an RNS variant.

From Heuristic to Guarantee. While heuristic results may be sufficient when developing functionality, the burden of proof is higher when we want to use these results to argue for privacy. Independence assumptions are commonplace; in this paper for example, we assumed independence between the noise of certain ciphertexts and implicitly between keyswitching keys. More generally in the RLWE setting, it is common to assume independence between the noise on each coefficient, and discrete distributions are approximated with continuous ones. Without understanding either the validity of these assumptions, or their impact on privacy, it is premature to make claims on the Differential Privacy of applications “in the wild".

[7] Indeed, high precision constants are used, requiring an additional rescale, as well as multiplying by 1-hot masks to compensate for the feature by feature encoding. By contrast, our method uses 1 level in precomputation of M_{jk}, Y_j, and then one multiplication per iteration.

Alternative Applications and Schemes. As a major challenge identified in our work is message dependent noise growth, future work may seek encodings and algorithms which minimize this factor, in order to take advantage of Differential Privacy “for free". In particular, we note that message dependence is not present until the first multiplication – if an algorithm begins with many additions, it may be possible to argue that differential privacy has already been achieved before message dependence becomes an issue, and then argue privacy via post processing. Alternatively, schemes such as TFHE do not appear to suffer from the same message dependence, so that relaxing the correctness requirements may result in differentially private outputs.

Beyond Output Perturbation. In this work, we only looked into whether the final noise was sufficient to give differential privacy. In short, we modelled the homomorphic noise as behaving like an output perturbation. However, noise is added to the data during encryption, and so could alternatively be characterized as an input perturbation. Indeed, in some contexts which have very sensitive message spaces, it may be possible that the noise added during encryption means that differential privacy is already achieved at this stage, and then argue by post processing that any output must also be differentially private. It may also be possible to draw from known results on objective and gradient perturbation, since any noise added to the data will perturb the objective as well as any gradient – however, it would appear that message dependence would play a role in these contexts as well.

DP “At A Discount". We have explored Differential Privacy “for free" – i.e., without making modifications to the algorithm or scheme. Future work could investigate whether hybrid solutions can achieve better Differential Privacy guarantees, while still harnessing the noise in approximate HE. For example, it may be possible to introduce independent noise which is less than would be required in a pure DP solution, but compensates for the message dependence present in a pure HE solution. It may also be possible to argue that the noise flooding recommended in [60] to secure CKKS is of sufficient width to guarantee differential privacy.

Acknowledgements

We thank Fernando Virdia for his invaluable suggestions and discussions in the development of this work, including detailed comments on an early draft of this paper.

References

1. Intel he-toolkit. Online: https://github.com/intel/he-toolkit (Feb 2023)

2. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
3. Aharoni, E., Adir, A., Baruch, M., Drucker, N., Ezov, G., Farkash, A., Greenberg, L., Masalha, R., Moshkowich, G., Murik, D., et al.: Helayers: A tile tensors framework for large neural networks on encrypted data
4. Akavia, A., Leibovich, M., Resheff, Y.S., Ron, R., Shahar, M., Vald, M.: Privacy-preserving decision trees training and prediction. ACM Transactions on Privacy and Security **25**(3), 1–30 (2022)
5. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., et al.: Homomorphic encryption standard. Protecting privacy through homomorphic encryption pp. 31–62 (2021)
6. Badawi, A.A., Bates, J., Bergamaschi, F., Cousins, D.B., Erabelli, S., Genise, N., Halevi, S., Hunt, H., Kim, A., Lee, Y., Liu, Z., Micciancio, D., Quah, I., Polyakov, Y., R.V., S., Rohloff, K., Saylor, J., Suponitsky, D., Triplett, M., Vaikuntanathan, V., Zucca, V.: Openfhe: Open-source fully homomorphic encryption library. Cryptology ePrint Archive, Paper 2022/915 (2022), https://eprint.iacr.org/2022/915, https://eprint.iacr.org/2022/915
7. Boemer, F., Cammarota, R., Demmler, D., Schneider, T., Yalame, H.: Mp2ml: A mixed-protocol machine learning framework for private inference. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. pp. 1–10 (2020)
8. Boemer, F., Costache, A., Cammarota, R., Wierzynski, C.: ngraph-he2: A high-throughput framework for neural network inference on encrypted data. In: Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography. pp. 45–56 (2019)
9. Boemer, F., Kim, S., Seifu, G., D.M. de Souza, F., Gopal, V.: Intel hexl: Accelerating homomorphic encryption with intel avx512-ifma52. In: Proceedings of the 9th on Workshop on Encrypted Computing and Applied Homomorphic Cryptography. p. 57–62. WAHC '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3474366.3486926, https://doi.org/10.1145/3474366.3486926
10. Bossuat, J.P., Troncoso-Pastoriza, J., Hubaux, J.P.: Bootstrapping for approximate homomorphic encryption with negligible failure-probability by using sparse-secret encapsulation. In: Applied Cryptography and Network Security: 20th International Conference, ACNS 2022, Rome, Italy, June 20–23, 2022, Proceedings. pp. 521–541. Springer (2022)
11. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)
12. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) lwe. SIAM Journal on computing **43**(2), 831–871 (2014)
13. de Castro, L., Agrawal, R., Yazicigil, R., Chandrakasan, A., Vaikuntanathan, V., Juvekar, C., Joshi, A.: Does fully homomorphic encryption need compute acceleration? arXiv preprint arXiv:2112.06396 (2021)
14. Castryck, W., Iliashenko, I., Vercauteren, F.: On error distributions in ring-based lwe. LMS Journal of Computation and Mathematics **19**(A), 130–145 (2016). https://doi.org/10.1112/S1461157016000280

15. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research **12**(29), 1069–1109 (2011), http://jmlr.org/papers/v12/chaudhuri11a.html
16. Chen, H., Dai, W., Kim, M., Song, Y.: Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 395–412 (2019)
17. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: Bootstrapping for approximate homomorphic encryption. In: Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37. pp. 360–384. Springer (2018)
18. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y.: A full rns variant of approximate homomorphic encryption. In: Selected Areas in Cryptography–SAC 2018: 25th International Conference, Calgary, AB, Canada, August 15–17, 2018, Revised Selected Papers 25. pp. 347–368. Springer (2019)
19. Cheon, J.H., Hong, S., Kim, D.: Remark on the security of ckks scheme in practice. Cryptology ePrint Archive (2020)
20. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. pp. 409–437. Springer (2017)
21. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. pp. 377–408. Springer (2017)
22. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Tfhe: fast fully homomorphic encryption over the torus. Journal of Cryptology **33**(1), 34–91 (2020)
23. Chillotti, I., Joye, M., Ligier, D., Orfila, J.B., Tap, S.: Concrete: Concrete operates on ciphertexts rapidly by extending tfhe. In: WAHC 2020–8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. vol. 15 (2020)
24. Chillotti, I., Joye, M., Ligier, D., Orfila, J.B., Tap, S.: Concrete: Concrete operates on ciphertexts rapidly by extending tfhe. In: WAHC 2020-8th Workshop on Encrypted Computing & Applied Homomorphic Cryptography (2020)
25. Chillotti, I., Ligier, D., Orfila, J.B., Tap, S.: Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for tfhe. In: Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part III 27. pp. 670–699. Springer (2021)
26. Costache, A., Curtis, B.R., Hales, E., Murphy, S., Ogilvie, T., Player, R.: On the precision loss in approximate homomorphic encryption. Cryptology ePrint Archive (2022)
27. Costache, A., Laine, K., Player, R.: Evaluating the effectiveness of heuristic worst-case noise analysis in fhe. In: Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part II 25. pp. 546–565. Springer (2020)
28. Costache, A., Nürnberger, L., Player, R.: Optimizations and trade-offs for helib. Cryptology ePrint Archive (2023)

29. Crockett, E., Peikert, C., Sharp, C.: Alchemy: A language and compiler for homomorphic encryption made easy. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 1020–1037 (2018)
30. Dathathri, R., Kostova, B., Saarikivi, O., Dai, W., Laine, K., Musuvathi, M.: EVA: an encrypted vector arithmetic language and compiler for efficient homomorphic computation. In: Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2020, London, UK, June 15-20, 2020. pp. 546–561 (2020). https://doi.org/10.1145/3385412.3386023, https://doi.org/10.1145/3385412.3386023
31. Dathathri, R., Saarikivi, O., Chen, H., Laine, K., Lauter, K., Maleki, S., Musuvathi, M., Mytkowicz, T.: Chet: an optimizing compiler for fully-homomorphic neural-network inferencing. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation. pp. 142–156 (2019)
32. Ding, J., Zhang, X., Li, X., Wang, J., Yu, R., Pan, M.: Differentially private and fair classification via calibrated functional mechanism. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 622–629 (2020)
33. Drucker, N., Moshkowich, G., Pelleg, T., Shaul, H.: Bleach: Cleaning errors in discrete computations over ckks. Cryptology ePrint Archive (2022)
34. Ducas, L., Micciancio, D.: Fhew: bootstrapping homomorphic encryption in less than a second. In: Advances in Cryptology–EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I 34. pp. 617–640. Springer (2015)
35. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
36. Dwork, C., Rothblum, G.N., Vadhan, S.: Boosting and differential privacy. In: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science. pp. 51–60. IEEE (2010)
37. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012)
38. Fukuchi, K., Tran, Q.K., Sakuma, J.: Differentially private empirical risk minimization with input perturbation. In: Discovery Science: 20th International Conference, DS 2017, Kyoto, Japan, October 15–17, 2017, Proceedings 20. pp. 82–90. Springer (2017)
39. Gaboardi, M., Lim, H., Rogers, R., Vadhan, S.: Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In: International conference on machine learning. pp. 2111–2120. PMLR (2016)
40. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. p. 169–178. STOC '09, Association for Computing Machinery, New York, NY, USA (2009). https://doi.org/10.1145/1536414.1536440, https://doi.org/10.1145/1536414.1536440
41. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 75–92. Springer (2013)
42. Gorantala, S., Springer, R., Purser-Haskell, S., Lam, W., Wilson, R., Ali, A., Astor, E.P., Zukerman, I., Ruth, S., Dibak, C., et al.: A general purpose transpiler for fully homomorphic encryption. arXiv preprint arXiv:2106.07893 (2021)

43. Halevi, S., Shoup, V.: Design and implementation of helib: a homomorphic encryption library. Cryptology ePrint Archive (2020)
44. Hardt, M., Recht, B., Singer, Y.: Train faster, generalize better: Stability of stochastic gradient descent. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. p. 1225–1234. ICML'16, JMLR.org (2016)
45. Heaan v1.0. Online: https://github.com/snucrypto/HEAAN/releases/tag/1.0 (September 2018)
46. Jain, P., Thakurta, A.: Differentially private learning with kernels. In: Dasgupta, S., McAllester, D. (eds.) Proceedings of the 30th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 28, pp. 118–126. PMLR, Atlanta, Georgia, USA (17–19 Jun 2013), https://proceedings.mlr.press/v28/jain13.html
47. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 1895–1912. USENIX Association, Santa Clara, CA (Aug 2019), https://www.usenix.org/conference/usenixsecurity19/presentation/jayaraman
48. Jayaraman, B., Wang, L., Evans, D., Gu, Q.: Distributed learning without distress: Privacy-preserving empirical risk minimization. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018), https://proceedings.neurips.cc/paper/2018/file/7221e5c8ec6b08ef6d3f9ff3ce6eb1d1-Paper.pdf
49. Jiang, X., Kim, M., Lauter, K., Song, Y.: Secure outsourced matrix computation and application to neural networks. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 1209–1222 (2018)
50. Kang, Y., Liu, Y., Niu, B., Tong, X., Zhang, L., Wang, W.: Input perturbation: A new paradigm between central and local differential privacy (2020). https://doi.org/10.48550/ARXIV.2002.08570, https://arxiv.org/abs/2002.08570
51. Kifer, D., Smith, A., Thakurta, A.: Private convex empirical risk minimization and high-dimensional regression. In: Mannor, S., Srebro, N., Williamson, R.C. (eds.) Proceedings of the 25th Annual Conference on Learning Theory. Proceedings of Machine Learning Research, vol. 23, pp. 25.1–25.40. PMLR, Edinburgh, Scotland (25–27 Jun 2012), https://proceedings.mlr.press/v23/kifer12.html
52. Kim, A., Papadimitriou, A., Polyakov, Y.: Approximate homomorphic encryption with reduced approximation error. In: Galbraith, S.D. (ed.) Topics in Cryptology – CT-RSA 2022. pp. 120–144. Springer International Publishing, Cham (2022)
53. Kim, A., Song, Y., Kim, M., Lee, K., Cheon, J.H.: Logistic regression model training based on the approximate homomorphic encryption. BMC medical genomics **11**(4), 23–31 (2018)
54. Kim, M., Song, Y., Wang, S., Xia, Y., Jiang, X., et al.: Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR medical informatics **6**(2), e8805 (2018)
55. Kim, S., Park, M., Kim, J., Kim, T., Min, C.: Evalround algorithm in ckks bootstrapping. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part II. pp. 161–187. Springer (2023)
56. Klemsa, J.: Setting up efficient tfhe parameters for multivalue plaintexts and multiple additions. Cryptology ePrint Archive (2021)

57. Kolar, M., Liu, H.: Marginal regression for multitask learning. In: Lawrence, N.D., Girolami, M. (eds.) Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 22, pp. 647–655. PMLR, La Palma, Canary Islands (21–23 Apr 2012), https://proceedings.mlr.press/v22/kolar12.html
58. Lattigo v2.2.0. Online: http://github.com/ldsec/lattigo (Feb 2023), ePFL-LDS
59. Li, B., Micciancio, D.: On the security of homomorphic encryption on approximate numbers. In: Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40. pp. 648–677. Springer (2021)
60. Li, B., Micciancio, D., Schultz, M., Sorrell, J.: Securing approximate homomorphic encryption using differential privacy. In: Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I. pp. 560–589. Springer (2022)
61. Ligett, K., Neel, S., Roth, A., Waggoner, B., Wu, Z.S.: Accuracy first: Selecting a differential privacy level for accuracy-constrained erm. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 2563–2573. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
62. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. Journal of the ACM (JACM) **60**(6), 1–35 (2013)
63. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In: Advances in Cryptology–EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings 32. pp. 35–54. Springer (2013)
64. Ma, J., Naas, S.A., Sigg, S., Lyu, X.: Privacy-preserving federated learning based on multi-key homomorphic encryption. International Journal of Intelligent Systems **37**(9), 5880–5901 (2022)
65. Murphy, S., Player, R.: A central limit framework for ring-lwe decryption. Cryptology ePrint Archive (2019)
66. Ogilvie, T., Player, R., Rowell, J.: Improved privacy-preserving training using fixed-hessian minimisation. In Michael Brenner, Tancrède Lepoint (Eds.), proceedings of the 8th Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC '20) (2020), https://doi.org/10.25835/0072999
67. PALISADE Lattice Cryptography Library (release 1.11.5). https://palisade-crypto.org/ (Feb 2023)
68. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K.: Semi-supervised knowledge transfer for deep learning from private training data. arXiv preprint arXiv:1610.05755 (2016)
69. Phong, L.T., Aono, Y., Hayashi, T., Wang, L., Moriai, S.: Privacy-preserving deep learning via additively homomorphic encryption. IEEE Transactions on Information Forensics and Security **13**(5), 1333–1345 (2018). https://doi.org/10.1109/TIFS.2017.2787987
70. Polyakov, Y., Rohloff, K., Ryan, G.W.: Palisade lattice cryptography library user manual (2017)
71. Raisaro, J.L., Choi, G., Pradervand, S., Colsenet, R., Jacquemont, N., Rosat, N., Mooser, V., Hubaux, J.P.: Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy. IEEE/ACM Transactions on Computational Biology and Bioinformatics **15**(5), 1413–1426 (2018). https://doi.org/10.1109/TCBB.2018.2854782

72. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6) (sep 2009). https://doi.org/10.1145/1568318.1568324, https://doi.org/10.1145/1568318.1568324
73. Riazi, M.S., Laine, K., Pelton, B., Dai, W.: Heax: An architecture for computing on encrypted data. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 1295–1309 (2020)
74. Rohloff, K., Cousins, D.B.: A scalable implementation of fully homomorphic encryption built on ntru. In: Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers 18. pp. 221–234. Springer (2014)
75. Samardzic, N., Feldmann, A., Krastev, A., Manohar, N., Genise, N., Devadas, S., Eldefrawy, K., Peikert, C., Sanchez, D.: Craterlake: a hardware accelerator for efficient unbounded computation on encrypted data. In: Proceedings of the 49th Annual International Symposium on Computer Architecture. pp. 173–187 (2022)
76. Microsoft SEAL (release 4.1). https://github.com/Microsoft/SEAL (Jan 2023), microsoft Research, Redmond, WA.
77. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1310–1321 (2015)
78. Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. In: 2013 IEEE global conference on signal and information processing. pp. 245–248. IEEE (2013)
79. Tang, P., Wang, W., Gu, X., Lou, J., Xiong, L., Li, M.: Two birds, one stone: Achieving both differential privacy and certified robustness for pre-trained classifiers via input perturbation (2021)
80. Tang, X., Zhu, L., Shen, M., Du, X.: When homomorphic cryptosystem meets differential privacy: training machine learning classifier with privacy protection. arXiv preprint arXiv:1812.02292 (2018)
81. Triastcyn, A., Faltings, B.: Federated learning with bayesian differential privacy. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 2587–2596. IEEE (2019)
82. Wu, X., Li, F., Kumar, A., Chaudhuri, K., Jha, S., Naughton, J.: Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In: Proceedings of the 2017 ACM International Conference on Management of Data. p. 1307–1322. SIGMOD '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3035918.3064047, https://doi.org/10.1145/3035918.3064047
83. Zhang, J., Zheng, K., Mou, W., Wang, L.: Efficient private erm for smooth objectives. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. p. 3922–3928. IJCAI'17, AAAI Press (2017)
84. Zhang, J., Zhang, Z., Xiao, X., Yang, Y., Winslett, M.: Functional mechanism: Regression analysis under differential privacy. Proc. VLDB Endow. **5**(11), 1364–1375 (jul 2012). https://doi.org/10.14778/2350229.2350253, https://doi.org/10.14778/2350229.2350253
85. Zhang, T., Zhu, T., Gao, K., Zhou, W., Philip, S.Y.: Balancing learning model privacy, fairness, and accuracy with early stopping criteria. IEEE Transactions on Neural Networks and Learning Systems (2021)

A CKKS Scheme and Noise Growth

A.1 CKKS Scheme

The CKKS, or HEAAN, scheme [20] is a cryptosystem based on the Ring Learning with Errors (RLWE) problem, and is parametrised by the power of 2 polynomial modulus N, power-of-two top level and auxiliary moduli Q_L and P, power-of-two precision parameter $\varDelta$, and secret key and error distributions S and χ. The relationship between PQ_L, N, S, and χ determines the security of the RLWE instance. How to set these securely is beyond the scope of this paper, but we refer to [5] for an overview.

For plaintexts $m \in \mathbb{Z}_Q[X]/(X^N+1)$ the scheme is as follows. We omit the details of homomorphic rotations as we do not use them.

$\texttt{SecretKeyGen}$**:** Sample $s \leftarrow S$ and output $\texttt{sk} = (1, s)$.
$\texttt{PublicKeyGen}(\texttt{sk})$**:** Sample $a \leftarrow R_q$ uniformly at random and $e \leftarrow \chi$. Output $\texttt{pk} = ([-as+e]_q, a)$.
$\texttt{EvaluationKeyGen}(\texttt{sk})$**:** Sample $a' \leftarrow R_{P.Q_L}$ uniformly at random and $e' \leftarrow \chi$. Output $\texttt{evk} = \left([-a's+e'+Ps^2]_{P\cdot q}, a'\right)$.
$\texttt{Encrypt}(\texttt{pk}, m \in R_{q_L})$**:** Let $\texttt{pk} = (p_0, p_1)$, sample $v \leftarrow S$ and $e_1, e_2 \leftarrow \chi$. Output

$$\texttt{ct} = ([m + p_0 v + e_1]_{Q_L}, [p_1 v + e_2]_{Q_L})$$

.
$\texttt{Decrypt}(\texttt{sk}, \texttt{ct})$**:** Let $\texttt{ct} = (c_0, c_1)$. Output $m' = [c_0 + c_1 s]_{q_l}$.
$\texttt{Add}(\texttt{ct}_0, \texttt{ct}_1)$**:** Output $\texttt{ct} = ([\texttt{ct}_0[0] + \texttt{ct}_1[0]]_q, [\texttt{ct}_0[1] + \texttt{ct}_1[1]]_q)$.
$\texttt{Pre-Multiply}(\texttt{ct}_0, \texttt{ct}_1)$**:** Set $d_0 = \left[\texttt{ct}_0[0]\texttt{ct}_1[0]\right]_q$,
$d_1 = \left[\texttt{ct}_0[0]\texttt{ct}_1[1] + \texttt{ct}_0[1]\texttt{ct}_1[0]\right]_q$, and
$d_2 = \left[\texttt{ct}_0[1]\texttt{ct}_1[1]\right]_q$. Output $\texttt{ct} = (d_0, d_1, d_2)$.
$\texttt{KeySwitch}(\texttt{ct} = (d_0, d_1, d_2), \texttt{evk})$**:** Output $(d_0, d_1) + \lfloor P^{-1} d_2 \cdot \texttt{evk} \rfloor \mod q$.
$\texttt{Rescale}(\texttt{ct}, \varDelta):$ Output $\texttt{ct} = (\lfloor \frac{1}{\varDelta} c'_0 \rfloor, \lfloor \frac{1}{\varDelta} c'_1 \rfloor)$.
$\texttt{Multiply}(\texttt{ct}_0, \texttt{ct}_1, \texttt{evk}, \varDelta)$**:** Output the result of $\texttt{Pre-Multiply}$, followed by $\texttt{KeySwitch}$, followed by $\texttt{Rescale}$.

To translate messages from $\mathbb{C}^{N/2}$ to plaintexts in $\mathbb{Z}[X]/(X^N+1)$, we use the canonical embedding. In more detail, if $\zeta_1, ..., \zeta_{N/2}$ are a set of primitive $2N$th roots of unity without conjugates, define the canonical embedding $\pi : \mathbb{Z}[X]/(X^N+1) \rightarrow \mathbb{C}^{N/2}$ as follows

$$\pi(m(X)) = \left(m(\zeta_1), m(\zeta_2), ..., m(\zeta_{N/2})\right).$$

We in fact have that π is an isomorphism, and we can use this as an encoding of our messages $z \in \mathbb{C}^{N/2}$ into plaintexts as follows:

$$\texttt{Encode}(z, \varDelta) = \lfloor \varDelta \pi^{-1}(z) \rceil.$$

When we have finished computing, we return to the message space via

$$\texttt{Decode}(m, \varDelta) = \frac{1}{\varDelta}\pi(m).$$

Remark 3. In our work, we will only use constant real scalars, rather than vectors, and so our encoding is much simpler – for $z \in \mathbb{R}$ we set the constant coefficient of the plaintext equal to $\lfloor \Delta z \rceil$. When we decode, we take only the real part of the first slot.

For our case study, we let $N = 2^{16}$, set χ as the discrete Gaussian with standard deviation 3.2, and let S be the uniform ternary distribution, so that the hamming weights of both s and v are well approximated by $\frac{2}{3}N$. For the keyswitching, we assume the auxiliary modulus has the same size as the top level modulus, or $Q_L = P$, as was the case in the original CKKS implementation [45]. We also implicitly assume the final modulus is large enough that neither plaintext nor noise have a wrap around. For a more comprehensive treatment of how to set parameters for Homomorphic Encryption, and the security of Ring Learning with Errors more generally, we refer to the HE Standard [5].

A.2 Noise Growth

We recap the heuristics from [26]. We describe the plaintext space, or ring, variance, and write this σ^2_{op}. We have that the variance on each slot in the real message space after decoding is then given by $\rho^2_{\text{op}} = \frac{N}{2\Delta^2}\sigma^2_{\text{op}}$.

Fresh The fresh noise polynomial is given by $ve + se_2 + e_1$. Approximating the discrete Gaussian with the continuous one, we have that the coefficients of this polynomial are distributed $N(0, (||v||_2^2 + ||s||_2^2 + 1)\sigma^2 I_N)$. Since for our case study we have $||v||^2, ||s||^2 \approx \frac{2N}{3}$, we can let $\sigma^2_{\text{fresh}} = \left(\frac{4N}{3} + 1\right)\sigma^2$.

Round Suppose we have a ciphertext with coefficients in the reals, and we wish to round it back to the integers. This introduces an additional error. If the rounding polynomials for ciphertext components 0 and 1 respectively are τ_0 and τ_1, the error from this procedure after decrypting will be $\tau_0 + \tau_1 s$, so that, modelling τ_i as having independent coefficients uniform in $[-\frac{1}{2}, \frac{1}{2}]$, the Central Limit Theorem gives that as a polynomial, this error distribution is well approximated by a $N(0, (\frac{1}{12} + \frac{1}{12}||s||_2^2)I_N)$, so that the polynomial noise variance is well approximated by

$$\sigma^2_{\text{round}} = \left(\frac{1}{12} + \frac{1}{18}N\right).$$

Keyswitch Error here has variance given by

$$\sigma^2_{\text{ks}} = \left(\frac{1}{12}P^{-2}Nq_l^2\sigma^2\right) + \mathbb{1}_{P\nmid q_l}\sigma^2_{\text{round}}$$

where P is the large keyswitching modulus – for our use case, this is equal to the top level modulus. For our heuristics therefore, we will only need to know the ratio between q_l and P.

Add If we have two independent encryptions, with noise variances σ_1^2 and σ_2^2 respectively, adding the two ciphertexts will give a ciphertext encrypting the sum of the underlying messages with noise variance given by $\sigma_1^2 + \sigma_2^2$.

Multiplication Suppose we have independent encryptions of the polynomials m_1 and m_2 with polynomial noise variances σ_1^2 and σ_2^2 respectively. Then after the tensor, the noise polynomial coefficients have distribution well approximated by $N(0, (N\sigma_1^2\sigma_2^2 + \sigma_1 \|m_2\|_2^2 + \sigma_2 \|m_1\|_2^2)I_N)$ so that, after the tensor but before the keyswitch and rescale, the error variance is given by

$$\sigma_{\text{mult}}^2 = N\sigma_1^2\sigma_2^2 + \sigma_1^2 \|m_2\|_2^2 + \sigma_2^2 \|m_1\|_2^2 . \tag{10}$$

To recover polynomial bounds (e.g., $\|m_1\|_2^2$), we recall that $\|z_i\|_2^2 = \frac{N}{2\Delta^2} \|m_i\|_2^2$. In particular, for vectors with constant entry $z \in \mathcal{R}$, as we will consider here, $\|m\|_2^2 = \Delta^2 z^2$.

Squaring If we instead square an encryption of a polynomial m, the noise random variables are clearly not independent. In this case, we have the noise has variance

$$\sigma_{\text{square}}^2 = 2N\sigma^4 + 4\sigma^2 \|m\|_2^2 . \tag{11}$$

Rescale Rescale consists of taking an encryption of the polynomial Δm and dividing both ciphertext components by Δ and rounding, resulting in a an approximate ciphertext encrypting m with noise divided by Δ, plus an additional rounding error. Therefore the noise variance after rescale is given by

$$\sigma_{\text{rs}}^2 = \frac{1}{\Delta^2}\sigma^2 + \sigma_{\text{round}}^2.$$

Plaintext Multiplication When multiplying an encryption by a polynomial $p(X)$, we should use the multiplication heuristic, but model the polynomial $p(X)$ as an encryption whose only error is a rounding error resulting from encoding (an encoding error).

When instead multiplying by a constant $c \in \mathbb{Z}$ in the ring, we can instead simply scale the noise variance up by c^2. We will extend this heuristically to $c \in \mathbb{R}$ as well.

B Differential Privacy Proofs

B.1 Proof of Theorem 1

Let us first restate the theorem:

Let $\varepsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism is (ε, δ)-differentially private whenever $\rho \geq c\delta_f/\varepsilon$, where δ_f is the sensitivity.

The following proof is adapted from [35,51].

Proof. Fix adjacent databases $\mathcal{D}$ and $\mathcal{D}'$, and let $\mathcal{A}$ be the algorithm output when we use $\mathcal{D}$ as input database, and $\mathcal{A}'$ when we use $\mathcal{D}'$. Our strategy will be to show the ratio of probability density functions $f_{\mathcal{A}}(\alpha)/f_{\mathcal{A}'}(\alpha)$ is less than e^{ε}, except with probability at most δ as α follows the distribution of $\mathcal{A}$. From here we will be able to conclude. So let

$$\mathcal{A} = \beta + N(0, \rho^2 I_d), \quad \mathcal{A}' = \beta' + N(0, \rho^2 I_d).$$

where $\beta, \beta' \in \mathbb{R}^d$ are the true outputs of the algorithm, without noise, for databases $\mathcal{D}$ and $\mathcal{D}'$ respectively. Let $\kappa = \beta - \beta'$. By definition of sensitivity, $||\kappa|| \leq \delta_f$.

The pdf ratio at a point $\alpha \in \mathbb{R}^d$ therefore satisfies

$$\begin{aligned}
\frac{f_A(\alpha)}{f_{A'}(\alpha)} &= \exp\left(\frac{||\alpha - \beta'||^2}{2\rho^2} - \frac{||\alpha - \beta||^2}{2\rho^2}\right) && (12)\\
&= \exp\left(\frac{1}{2\rho^2}\left(||\alpha - \beta + \kappa||^2 - ||\alpha - \beta||^2\right)\right) && (13)\\
&= \exp\left(\frac{1}{2\rho^2}\left(2(\alpha - \beta) \cdot \kappa + ||\kappa||^2\right)\right) && (14)\\
&\leq \exp\left(\frac{1}{\rho^2}(\alpha - \beta) \cdot \kappa + \frac{\delta_f^2}{2\rho^2}\right). && (15)
\end{aligned}$$

We want to show this is less than e^{ε} with probability at least $1 - \delta$ as $\alpha \leftarrow \mathcal{A}$. So let

$$R := \left\{\alpha \in \mathbb{R}^d : \left|\frac{1}{\rho^2}(\alpha - \beta) \cdot \kappa + \delta_f^2/2\rho^2\right| \leq \varepsilon\right\}$$

Let $t = \varepsilon\rho^2 - \delta_f^2$, so that $\alpha \in R$ whenever $|(\alpha - \beta) \cdot \kappa| < t$. We want to show $\Pr[\mathcal{A} - \beta) \cdot \kappa > t] \leq \delta/2$. Since $(\mathcal{A} - \beta) \cdot \kappa \sim N(0, ||\kappa||^2 \rho^2)$, so that, using Corollary 1, we must show

$$\ln\left(\frac{t}{||\kappa||\,\rho}\right) + \frac{1}{2}\left(\frac{t}{||\kappa||\,\rho}\right)^2 > \ln\left(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}\right) \tag{16}$$

Let $\rho = c\delta_f/\varepsilon$. We will show that if $c^2 > 2\ln(1.25/\delta)$ then Eq. (16) holds. First observe

$$\frac{t}{||\kappa||\,\rho} \geq \frac{t}{\delta_f \rho} = c - \frac{\varepsilon}{2c} > 1,$$

whenever $\varepsilon \leq 1$ and $c > 3/2$, so that $\ln\left(\frac{t}{||\kappa||\rho}\right) > 0$ and we can ignore the first term. Now, since $\varepsilon \leq 1$ and we have already bounded $c > 3/2$,

$$\frac{1}{2}\left(\frac{t}{||\kappa||\,\rho}\right)^2 \geq \frac{1}{2}\left(c - \frac{\varepsilon}{2c}\right)^2 > \frac{1}{2}\left(c^2 - \frac{8}{9}\right) > \ln\left(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}\right),$$

with the last inequality following as $c^2 > 2\ln(1.25/\delta) > \frac{8}{9} + 2\ln\left(\sqrt{\frac{2}{\pi}}\right) + 2\ln(1/\delta)$. We have therefore shown $\Pr[\mathcal{A} \notin R] \leq 1 - \delta$.

To conclude, we have, for all $S \subset \mathbb{R}^d$,

$$\begin{aligned}\Pr[\mathcal{A} \in S] &= \Pr[\mathcal{A} \in S \cap R] + \Pr[\mathcal{A} \in S \cap (\mathbb{R}^d \backslash R)] && (17)\\ &\leq \int_{\alpha \in S \cap R} f_{\mathcal{A}}(\alpha) d\alpha + \Pr[\mathcal{A} \notin R] && (18)\\ &\leq \int_{\alpha \in S \cap R} e^{\varepsilon} f_{\mathcal{A}'}(\alpha) d\alpha + \delta && (19)\\ &\leq e^{\varepsilon} \Pr[\mathcal{A}' \in S] + \delta, && (20)\end{aligned}$$

so that the mechanism $\mathcal{A}$ is (ε, δ)-differentially private by definition.

B.2 Proof of Theorem 3

We will use the following tail bound for non-central chi-squared distributions.

Lemma 4. *[57] Let $Y \sim \chi^2_d(\nu)$ be a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter ν. Then for $x \geq 0$ we have*

$$\Pr\left[Y > d + \nu + 2\sqrt{(d + 2\nu)x} + 2x\right] \leq \exp(-x)$$

Let us restate Theorem 3.

Suppose we use the following Gaussian mechanism: for a database $\mathcal{D}$, we provide a sample from the distribution $N(\beta_{\mathcal{D}}, \Sigma_{\mathcal{D}})$, where $\beta_{\mathcal{D}} \in \mathbb{R}^d$, and $\Sigma_{\mathcal{D}}$ is a diagonal matrix with diagonal entries $\rho_{\mathcal{D},1}, ..., \rho_{\mathcal{D},d}$ which are dependent on the database $\mathcal{D}$. This mechanism is (ε, δ) differentially private whenever

$$\varepsilon > \sqrt{(\frac{1}{2}d(T^2 - 1) + T^4K^2)D} + \frac{1}{2}T^2K^2 + \frac{1}{2}(T^2 - 1)(D + d) + d\ln T \quad (21)$$

where $D = 2\ln\frac{1}{\delta}$, $\max\frac{\rho_{\mathcal{D},i}}{\rho_{\mathcal{D}',i}} \leq T$ for all i, and

$$\max\left|\left|\left(\frac{\beta_{\mathcal{D},1} - \beta_{\mathcal{D}',1}}{\rho_{\mathcal{D},1}}, ..., \frac{\beta_{\mathcal{D},d} - \beta'_{\mathcal{D}',d}}{\rho_{\mathcal{D},d}}\right)\right|\right| \leq K,$$

where maximums are taken over adjacent databases $\mathcal{D}$ and $\mathcal{D}'$.

Proof. We recycle the proof strategy from Theorem 2, namely, we fix a database $\mathcal{D}$ and identify a set $R \subset \mathbb{R}^d$ such that the mechanism $\mathcal{A} \in R$ with probability at least $1 - \delta$, and whenever $\alpha \in R$, the ratio $\frac{f_{\mathcal{A}}(\alpha)}{f_{\mathcal{A}'}(\alpha)} \leq e^{\varepsilon}$ for all adjacent database mechanisms, $\mathcal{A}'$.

So fix a database with mechanism $\mathcal{A} \sim N(\beta, \Sigma)$, and an adjacent database with mechanism $\mathcal{A}' \sim N(\beta', \Sigma')$, where Σ and Σ' are diagonal with diagonal entries $(\rho_1, ..., \rho_d)$ and $(\rho'_1, ..., \rho'_d)$ respectively. Then the ratio of pdfs at a point $\alpha \in \mathbb{R}^d$ is given by

$$\begin{aligned}\frac{f_{\mathcal{A}}(\alpha)}{f_{\mathcal{A}'}(\alpha)} &= \left(\prod_{i=1}^{d} \frac{\rho'_i}{\rho_i}\right) \exp\left(\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\alpha_i - \beta'_i}{\rho'_i}\right)^2 - \left(\frac{\alpha_i - \beta_i}{\rho_i}\right)^2\right) \\ &= \prod_{i=1}^{d} \frac{1}{\tau_i} \exp\left(\frac{1}{2}\sum_{i=1}^{d} \tau_i^2 \left(\frac{\alpha_i - \beta_i}{\rho_i} - \kappa_i\right)^2 - \left(\frac{\alpha_i - \beta_i}{\rho_i}\right)^2\right)\end{aligned}$$

where $\tau_i = \frac{\rho_i}{\rho'_i}$ and $\kappa_i = \frac{\beta'_i - \beta_i}{\rho_i}$. We are interested in bounding the probability this exceeds e^ε as $\alpha \leftarrow \mathcal{A}$. Letting $Z \sim N(0, I_d)$ be a standard normal, this probability can be written

$$\begin{aligned}&\Pr\left[\sum_{i=1}^{d}\left(\ln \tau_i + \frac{1}{2}\tau_i^2 (Z_i - \kappa_i)^2 - \frac{1}{2}Z_i^2\right) > \varepsilon\right] && (22) \\ &\leq \Pr\left[d \ln T + \frac{1}{2}\sum_{i=1}^{d}\left(T^2 (Z_i - \kappa_i)^2 - Z_i^2\right) > \varepsilon\right] \\ &= \Pr\left[(T^2 - 1)\left|\left|Z - \frac{T^2}{T^2 - 1}\kappa\right|\right|^2 - \frac{T^2}{T^2 - 1}\left|\left|\kappa\right|\right|^2 > 2\varepsilon - 2d \ln T\right] \\ &= \Pr\left[X > \frac{2\varepsilon - 2d \ln T}{T^2 - 1} + \frac{T^2}{(T^2 - 1)^2}\left|\left|\kappa\right|\right|^2\right], && (23)\end{aligned}$$

where $X = \left|\left|Z - \frac{T^2}{T^2-1}\kappa\right|\right|^2 \sim \chi_d^2\left(\left|\left|\frac{T^2}{T^2-1}\kappa\right|\right|^2\right)$ is a non central chi-squared distribution with d degrees of freedom, and non centrality parameter $\left|\left|\frac{T^2}{T^2-1}\kappa\right|\right|^2$. Using Lemma 4 with $x = \frac{1}{2}D$ we have that (23) is less than δ so long as

$$\begin{aligned}\frac{2\varepsilon - 2d \ln T}{T^2 - 1} &+ \frac{T^2}{(T^2 - 1)^2}\left|\left|\kappa\right|\right|^2 \\ &\geq d + \left|\left|\frac{T^2}{T^2 - 1}\kappa\right|\right|^2 + 2\sqrt{\left(\frac{1}{2}d + \left|\left|\frac{T^2}{T^2 - 1}\kappa\right|\right|^2\right) D} + D.\end{aligned}$$

Rearranging for ε, this is precisely what is guaranteed by the condition (21).

To conclude the proof, let

$$R = \left\{\alpha \in \mathbb{R}^d : \frac{f_{\mathcal{A}}(\alpha))}{f_{\mathcal{A}'}(\alpha)} \leq e^\varepsilon\right\}.$$

We have shown that $\Pr[\mathcal{A} \in R] \geq 1 - \delta$. Now we have

$$\begin{aligned}
\Pr[\mathcal{A} \in S] &= \Pr[\mathcal{A} \in S \cap R] + \Pr[\mathcal{A} \in S \cap (\mathbb{R}^d \backslash R)] \\
&\leq \int_{\alpha \in S \cap R} f_{\mathcal{A}}(\alpha) d\alpha + \Pr[\mathcal{A} \notin R] \\
&\leq \int_{\alpha \in S \cap R} e^{\varepsilon} f_{\mathcal{A}'}(\alpha) d\alpha + \delta \\
&\leq e^{\varepsilon} \Pr[\mathcal{A}' \in S] + \delta
\end{aligned}$$

So that this mechanism is differentially private by definition.

C Case Study Details

C.1 Noise Analysis

We will calculate how the ring variance changes over the course of an iteration, so that we can iteratively calculate the ring variance $\sigma^2_{\beta_j}$ from the fixed starting point. From here, we can calculate the real variance $\rho^2_{\beta_j}$ via $\rho^2_{\beta_j} = \frac{N}{2\Delta^2}\sigma^2_{\beta_j}$.

Let us first follow [66] and rearrange the update circuit for the j^{th} weight as follows.

$$\begin{aligned}
\beta_j &\leftarrow \beta_j - \alpha \frac{\partial J}{\partial \beta_j} \\
&= (1 - \lambda\alpha)\beta_j + \alpha \underbrace{\frac{1}{n}\sum_{i=1}^{n} y_i x_{ij}}_{Y_j} - \alpha \sum_{k=1}^{d} \beta_k \underbrace{\frac{1}{n}\sum_{i=1}^{n} x_{ij} x_{ik}}_{M_{jk}}
\end{aligned}$$

In the original homomorphic implementation of ridge regression training using gradient descent [66], SIMD, or packing, was used for efficiency. We will however assume that each value is packed into a separate ciphertext, and call them $\mathsf{ct}.x_{ij}$, $\mathsf{ct}.y_j$, etc.. The ciphertexts $\mathsf{ct}.Y_j$ and $\mathsf{ct}.M_{jk}$ are therefore calculated homomorphically via

$$\mathsf{ct}.Y_j \leftarrow \frac{1}{n}\sum_{i=1}^{n} \mathsf{Mult}(\mathsf{ct}.y_i, \mathsf{ct}.x_{ij}), \tag{24}$$

$$\mathsf{ct}.M_{jk} \leftarrow \frac{1}{n}\sum_{i=1}^{n} \mathsf{Mult}(\mathsf{ct}.x_{ij}, \mathsf{ct}.x_{ik}), \tag{25}$$

and gradient descent updates are then evaluated via

$$\begin{aligned}
\mathsf{ct}.\beta_j \leftarrow \underbrace{\mathsf{Mult}\left(1 - \lambda\alpha - \alpha\mathsf{ct}.M_{jj}, \mathsf{ct}.\beta_j\right)}_{(1)} + \underbrace{\alpha\mathsf{ct}.Y_j}_{(2)} \\
\underbrace{- \alpha \sum_{k \neq j} \mathsf{Mult}\left(\mathsf{ct}.\beta_k, \mathsf{ct}.M_{jk}\right)}_{(3)}
\end{aligned}$$

We highlight that, since M_{jj} corresponds to summing the squares of ciphertexts, we calculate this variance using the squaring heuristic (see (Eq. (11)) rather than the multiplication heuristic (see Eq. (10)).

We model the noise of the three terms (1), (2) and (3) as independent, as well as assuming independence within these terms: e.g. noise on $\mathsf{ct}.\beta_k$ is independent of the noise associated with $\mathsf{ct}.M_{jk}$, the noise associated with their product is independent of the noise in the product of $\mathsf{ct}.\beta_{k'}$ and $\mathsf{ct}.M_{jk'}$ etc.

The variance of the noise in the polynomial ring for the ciphertexts $\mathsf{ct}.Y_j$ and $\mathsf{ct}.M_{jk}$ for $j \neq k$ are then given by

$$\sigma_{Y_j}^2 = \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + \sigma_{\text{fresh}}^2\sum_{i=1}^{n}(x_{ij}^2 + y_i^2) + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{26}$$

$$\sigma_{M_{jk}}^2 = \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + \sigma_{\text{fresh}}^2\sum_{i=1}^{n}(x_{ij}^2 + x_{ik}^2) + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{27}$$

while for $j = k$, we have

$$\sigma_{M_{jj}}^2 = \frac{1}{n^2}\left(\frac{2nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 4\sigma_{\text{fresh}}^2\sum_{i=1}^{n}x_{ij}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right) \tag{28}$$

where we have assumed a lazy keyswitching/rescale paradigm has been employed [55,52]. As the weights are initialized to zero, we have that the variances after the first iteration are given $\sigma_{\beta_j}^2 \leftarrow \alpha^2\sigma_{Y_j}^2$.

For a fixed j, the three variances are therefore given by

$$\begin{aligned}
\sigma_1^2 &= \frac{N\alpha^2}{\Delta^2}\sigma_{M_{jj}}^2\sigma_{\beta_j}^2 + \alpha^2\sigma_{M_{jj}}^2\beta_j^2 + \sigma_{\beta_j}^2(1-\lambda\alpha-\alpha M_{jj})^2,\\
\sigma_2^2 &= \alpha^2\sigma_{Y_j}^2,\\
\sigma_3^2 &= \sum_{k\neq j}\left(\frac{N\alpha^2}{\Delta^2}\sigma_{M_{jk}}^2\sigma_{\beta_k}^2 + \alpha^2\sigma_{M_{jk}}^2\beta_k^2 + \alpha^2\sigma_{\beta_k}^2M_{jk}^2\right).
\end{aligned}$$

After the update, we can update $\sigma_{\beta_j}^2 \leftarrow \sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2$.

C.2 Variance Bounds

Lower Bounds After a multiplication and rescale of two ciphertexts $\mathsf{ct}.m_1$ and $\mathsf{ct}.m_2$ with ring noise variances σ_1^2 and σ_2^2, our noise variance is of the form

$$\frac{N}{\Delta^2}\sigma_1^2\sigma_2^2 + \sigma_1^2m_2^2 + \sigma_2^2m_1^2 \tag{29}$$

As Δ^2 is very large, we would expect noise growth to come from the $\sigma_i^2m_j^2$ terms. In standard noise analysis, we only want to bound noise above, and so it is sufficient to upper bound message magnitude. However, for our purposes, we will want to bound messages away from zero to accurately reflect noise growth. To this end, we introduce a new parameter c and claim bounds of the form

$$\text{for all } j, \sum_{i=1}^{n} x_{ij}^2 \geq c, \text{as well as} \sum_{i=1}^{n} y_i^2 \geq c \tag{30}$$

In other words, the 2-norms of the columns are bounded below. As we already assume $|y_i|, |x_{ij}| \leq 1$, we have $0 \leq c \leq n$; c can be thought of as reflecting the density of databases under consideration. We explore the impact of c on privacy in our experiments.

Remark 4. As with Heuristic 2.51, for a fixed c, we will only be able to ensure Differential Privacy over the databases where Eq. (30) holds.

The parameter c enables us to more realistically lower bound $\sigma_{Y_j}^2, \sigma_{M_{jk}}^2, \sigma_{M_{jj}}^2$. We therefore can bound as follows.

$$\sigma_{Y_j}^2 \geq \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 2c\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{31}$$

$$\sigma_{M_{jk}}^2 \geq \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 2c\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{32}$$

$$\sigma_{M_{jj}}^2 \geq \frac{1}{n^2}\left(\frac{2nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 4c\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right). \tag{33}$$

As the first iteration is given by $\beta_j \leftarrow \alpha Y_j$, we can initialize our lower bound on $\sigma_{\beta_j}^2$ with α^2 multiplied by the lower bound on $\sigma_{Y_j}^2$.

If $\alpha \leq 1$, as it will be for our experiments, since $M_{jj} \leq 1$, we can bound $(1 - \alpha M_{jj} - \lambda\alpha)^2 \geq (1 - \alpha - \lambda\alpha)^2$. Without additional assumptions, we cannot bound any of the β_k^2, M_{jk}^2 away from zero. We therefore lower bound $\sigma_{\beta_j}^2$ after the update as follows:

$$\sigma_{\beta_j}^2 \geq \frac{N\alpha^2}{\Delta^2}\sum_{k=1}^{d}\sigma_{\beta_k}^2\sigma_{M_{jk}}^2 + \alpha^2\sigma_{Y_j}^2 + \sigma_{\beta_j}^2\left(\alpha M_{jj} + \lambda\alpha - 1\right)^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2,$$

where we replace each term with its lower bound.

Upper Bounds From the assumptions $x_{ij}^2 \leq 1$ and $y_i^2 \leq 1$, we have that $\sum_{i=1}^{n}(x_{ij}^2 + y_i^2) \leq 2n$ and $\sum_{i=1}^{n} x_{ij}^2 \leq n$. We can therefore upper bound the relevant variances as follows:

$$\sigma_{Y_j}^2 \leq \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 2n\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{34}$$

$$\sigma_{M_{jk}}^2 \leq \frac{1}{n^2}\left(\frac{nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 2n\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right), \tag{35}$$

$$\sigma_{M_{jj}}^2 \leq \frac{1}{n^2}\left(\frac{2nN}{\Delta^2}\sigma_{\text{fresh}}^4 + 4n\sigma_{\text{fresh}}^2 + \sigma_{\text{ks}}^2 + \sigma_{\text{round}}^2\right). \tag{36}$$

We can additionally upper bound $\beta_j^2 \leq \frac{1}{\lambda}, M_{jk}^2 \leq 1$, and if $\alpha \leq 1$, $(\alpha M_{jj} + \lambda\alpha - 1)^2 \leq \max((\lambda\alpha - 1)^2, (\lambda\alpha)^2)$.

As the first iteration is given by $\beta_j \leftarrow \alpha Y_j$, we can initialize our upper bound on $\sigma_{\beta_j}^2$ with α^2 multiplied by the upper bound on $\sigma_{Y_j}^2$.

Therefore, if we have that each $\sigma_{\beta_j}^2 \leq \sigma_\beta^2$, after performing an additional iteration we have

$$\begin{aligned}\sigma_{\beta_j}^2 \leq \frac{N\alpha^2}{\Delta^2}\sum_{k=1}^{d}\sigma_{M_{jk}}^2\sigma_\beta^2 + \frac{\alpha^2}{\lambda}\sum_{k=1}^{d}\sigma_{M_{jk}}^2 + \alpha^2\sigma_{Y_j}^2 + \\ (d-1)\alpha^2\sigma_\beta^2 + \sigma_\beta^2(\alpha M_{jj} + \lambda\alpha - 1)^2,\end{aligned} \tag{37}$$

where terms are replaced with their upper bounds.it'

C.3 Message Dependence

To apply our theorem, we need to bound the ratio of message space variances $\frac{\rho^2}{\rho'^2}$ for adjacent databases at iteration k. We will approach this by bounding $\left|\sigma^2 - \sigma'^2\right|$, where σ^2 and σ'^2 are in the ring and associated with adjacent databases, and then invoking:

$$\frac{\rho^2}{\rho'^2} = \frac{\sigma^2}{\sigma'^2} \leq 1 + \frac{|\sigma^2 - \sigma'^2|}{\sigma_{\min}^2}$$

where $\sigma_{\min}^2$ is the minimum possible value for σ^2.

Remark 5. This methodology may lead to weak bounds. In particular, it may be that the databases which maximize $\left|\sigma^2 - \sigma'^2\right|$ do not coincide with the databases which minimize σ^2. In our context, we believe this method produces least upper bounds, particularly for continuous input data.

Our analysis makes use of the following identities:

- if $x^2, x'^2 \leq 1$, then $\left|x^2 - x'^2\right| \leq 1$.
- $|\sigma_1^2\sigma_2^2 - \sigma_1'^2\sigma_2'^2| \leq \sigma_1^2|\sigma_2^2 - \sigma_2'^2| + \sigma_2'^2|\sigma_1^2 - \sigma_1^2|$,
- If $||x||^2, ||x'||^2 \leq \frac{1}{\lambda}$, then for each j we have $|x_j^2 - x_j'^2| \leq \frac{2}{\sqrt{\lambda}}\,||x - x'||$.

Recall our approach is to bound $|\sigma^2 - \sigma'^2|$ at a fixed iteration. In this section, as well as in the code, we will use the notation d_X to denote the difference in variance when calculating X with adjacent databases, or

$$\left|\sigma_X^2 - \sigma_{X'}^2\right| \leq d_X,$$

so that the aim of this section is to upper bound d_{β_j} at iteration k.

Assume without loss of generality the two databases differ on the n^{th} row. For our three primary variances, we have

$$d_{Y_j} = \frac{\sigma^2_{\text{fresh}}}{n^2}\left|x^2_{nj} + y^2_n - x'^2_{nj} - y'^2_n\right| \leq \frac{2\sigma^2_{\text{fresh}}}{n^2}, \tag{38}$$

$$d_{M_{jk}} = \frac{\sigma^2_{\text{fresh}}}{n^2}\left|x^2_{nj} + x^2_{nk} - x'^2_{nj} - x'^2_{nk}\right| \leq \frac{2\sigma^2_{\text{fresh}}}{n^2}, \tag{39}$$

$$d_{M_{jj}} = \frac{4\sigma^2_{\text{fresh}}}{n^2}\left|x^2_{nj} - x'^2_{nj}\right| \leq \frac{4\sigma^2_{\text{fresh}}}{n^2}. \tag{40}$$

As before, we can initialize d_{β_j} with α^2 multiplied with the upper bound on d_{Y_j}.

In addition, we can bound the message differences as follows:

$$\left|M^2_{jk} - M^2_{jk}\right| = \left|\left(M_{jk} - M'_{jk}\right)\left(M_{jk} + M'_{jk}\right)\right| \leq \left|\frac{2}{n}\cdot 2\right| = \frac{4}{n}, \tag{41}$$

$$\left|\beta^2_k - \beta'^2_k\right| \leq \frac{2}{\sqrt{\lambda}}\left|\left|\beta_k - \beta_{k'}\right|\right|, \tag{42}$$

while

$$\begin{aligned} &\left|(\alpha M_{jj} + \lambda\alpha - 1)^2 - (\alpha M'_{jj} + \lambda\alpha - 1)^2\right| \\ &\qquad = |\alpha(M_{jj} - M_{jj'})(\alpha M_{jj} + \alpha M_{jj'} + 2\lambda\alpha - 2)| \\ &\qquad \leq \frac{2\alpha}{n}\max(2, 2\alpha(\lambda + 1)). \end{aligned} \tag{43}$$

We can now explain how to update the bound on $|\sigma^2_{\beta_j} - \sigma'^2_{\beta_j}|$. Using the triangle inequality, we must sum bounds on the following

$$\begin{aligned} d_1 &= \frac{N\alpha^2}{\Delta^2}\sum_{k=1}^{d}|\sigma^2_{M_{jk}}\sigma^2_{\beta_j} - \sigma^2_{M'_{jk}}\sigma^2_{\beta'_j}|, \\ d_2 &= \alpha^2\sum_{k=1}^{d}|\sigma^2_{M_{jk}}\beta^2_k - \sigma^2_{M'_{jk}}\beta'^2_k|, \\ d_3 &= \alpha^2\sum_{k\neq j}|\sigma^2_{\beta_k}M^2_{jk} - \sigma^2_{\beta'_k}M'^2_{jk}|, \\ d_4 &= \alpha^2|\sigma^2_{Y_j} - \sigma^2_{Y'_j}|, \\ d_5 &= |\sigma^2_{\beta_j}(1 - \lambda\alpha - \alpha M_{jj})^2 - \sigma^2_{\beta'_j}(1 - \lambda\alpha - \alpha M'_{jj})^2|. \end{aligned}$$

Via repeated application of $|x^2y^2 - x'^2y'^2| \leq x^2|y^2 - y'^2| + y'^2|x^2 - x'^2|$, these terms can be bounded as follows:

$$\begin{aligned}
d_1 &\leq \frac{N\alpha^2}{\Delta^2}\sum_{k=1}^{d}\left(\sigma^2_{M_{jk}}\left|\sigma^2_{\beta_k} - \sigma^2_{\beta'_k}\right| + \sigma^2_{\beta'_k}\left|\sigma^2_{M_{jk}} - \sigma^2_{M'_{jk}}\right|\right),\\
d_2 &\leq \alpha^2\sum_{k=1}^{d}\left(\sigma^2_{M_{jk}}\left|\beta^2_k - \beta'^2_k\right| + \sigma^2_{\beta'_k}\left|\sigma^2_{M_{jk}} - \sigma^2_{M'_{jk}}\right|\right),\\
d_3 &\leq \alpha^2\sum_{k\neq j}\left(\sigma^2_{\beta_k}\left|M^2_{jk} - M'^2_{jk}\right| + M'^2_{jk}\left|\sigma^2_{\beta_k} - \sigma^2_{\beta'_k}\right|\right),\\
d_5 &= \alpha^2|\sigma^2_{Y_j} - \sigma^2_{Y'_j}|,
\end{aligned}$$

while

$$\begin{aligned}
d_4 \leq \sigma^2_{\beta_j}&\left|(\alpha M_{jj} + \lambda\alpha - 1)^2 - (\alpha M'_{jj} + \lambda\alpha - 1)^2\right|\\
&+ (\alpha M'_{jj} + \lambda\alpha - 1)^2\left|\sigma^2_{\beta_j} - \sigma^2_{\beta'_j}\right|. \qquad (44)
\end{aligned}$$

We then replace each term with it's upper bound, either using the sensitivity analysis, the bounds in this section, or the variance upper bounds.