High-Precision Arithmetic in Homomorphic Encryption

Hao Chen[1], Kim Laine[2], Rachel Player[3], and Yuhou Xia[4]

[1] Microsoft Research, USA haoche@microsoft.com
[2] Microsoft Research, USA kim.laine@microsoft.com
[3] Royal Holloway, University of London, UK
rachel.player.2013@live.rhul.ac.uk
[4] Princeton University yuhoux@math.princeton.edu

Abstract. In most RLWE-based homomorphic encryption schemes the native plaintext elements are polynomials in a rting $\mathbb{Z}_t[x]/(x^n+1)$, where n is a power of 2, and t an integer modulus. For performing integer or rational number arithmetic, one typically uses an encoding scheme which converts the inputs to polynomials, and allows the result of the homomorphic computation to be decoded to recover the result as an integer or rational number, respectively. The problem is that the modulus t often needs to be extremely large to prevent the plaintext polynomial coefficients from being reduced modulo t during the computation, which is a requirement for the decoding operation to work correctly. This results in larger noise growth, and prevents the evaluation of deep circuits, unless the encryption parameters are significantly increased.
We combine a trick of Hoffstein and Silverman, where the modulus t is replaced by a polynomial $x-b$, with the Fan-Vercauteren homomorphic encryption scheme. This yields a new scheme with a very convenient plaintext space $\mathbb{Z}/(b^n+1)\mathbb{Z}$. We then show how rational numbers can be encoded as elements of this plaintext space, enabling homomorphic evaluation of deep circuits with high-precision rational number inputs. We perform a fair and detailed comparison to the Fan-Vercauteren scheme with the Non-Adjacent Form encoder, and find that the new scheme significantly outperforms this approach. For example, when the new scheme allows us to evaluate circuits of depth 9 with 32-bit integer inputs, in the same parameter setting the Fan-Vercauteren scheme only allows us to go up to depth 2. We conclude by discussing how known applications can benefit from the new scheme.

Keywords: homomorphic encryption, encoding, encrypted arithmetic

1 Introduction

1.1 Background

Fully homomorphic encryption enables Boolean or arithmetic circuits to be evaluated on encrypted data, without requiring access to the secret key. While the

idea is old [40], the existence of such encryption schemes was an open problem for decades, and was solved only in 2009 by Craig Gentry [24], with an explicit construction based on ideal lattices. While the scheme of [24] was impractical, a long list of vastly more efficient schemes have since emerged [12, 11, 22, 9, 26]. Several lines of research have focused on improving the efficiency of homomorphic encryption for practical tasks, e.g. by improving the data representations [38, 25, 41, 21, 16], and by providing clever optimization tricks to improve the performance of existing schemes both from a theoretical [25, 30] and a software engineering [37, 30] point of view.

All of the schemes mentioned above have several features in common. For example, their security is based on the hardness of either the Learning With Errors (LWE) [39] or the Ring Learning With Errors (RLWE) [36] problem, which makes the plaintext and ciphertext spaces to be very similar in all of the schemes. Another commonality is that in each scheme every ciphertext comes with an inherent attribute called *noise*, which accumulates in homomorphic operations—in particular in multiplications—and corrupts the ciphertext once it reaches a certain maximum value. Once a ciphertext is corrupted, it can no longer be decrypted, even with the correct secret key. Gentry [24] used a clever *bootstrapping* procedure to re-encrypt a homomorphically encrypted ciphertext under a second layer of encryption, by evaluating the decryption circuit homomorphically using the encryptions of the bits of the secret key. While there has been a lot of work recently towards making bootstrapping more practical [18, 6], and improving it further is certainly an interesting direction for future work, typically a more efficient solution is to simply increase the parameters of the encryption scheme to allow deep enough circuits to be evaluated before the noise ceiling is reached. This approach—called *leveled (fully) homomorphic encryption* [5]—has been remarkably successful: most implementations of homomorphic encryption do not implement bootstrapping, and most papers discussing applications do not use it. In this paper we focus on the leveled approach.

In most schemes based on the RLWE assumption, the natural plaintext elements are polynomials in a ring $R_t = \mathbb{Z}_t[x]/\Phi_m(x)$, where Φ_m denotes the m-th cyclotomic polynomial. For security and performance reasons it is common to restrict m to be a power of 2, in which case $\Phi_{2n}(x)$ is of the form $x^n + 1$. Thus, homomorphic operations performed on ciphertexts reflect on the plaintext side as additions and multiplications in the ring R_t. This is extremely unnatural for nearly all naturally occurring applications, as in practice we often want to perform operations on encrypted integers and rational numbers. For this reason, an *encoding* of elements of $\mathbb{Z}$ or $\mathbb{Q}$ into polynomials in R_t is needed. Such an encoding needs to respect both additions and multiplications, and also be injective in a large domain (subset of $\mathbb{Z}$ or $\mathbb{Q}$), so that the results of the computation can be decoded after decryption. Several encoding methods for integers and rational numbers have been proposed in the literature [38, 10, 32, 21, 20, 16], but all of these have a common limitation: the decoding operation will work correctly only as long as the homomorphic operations do not cause the underlying plaintext polynomial coefficients to be reduced modulo the integer t. In other words, in

order for the result to be correct as an integer or as a rational number, t needs to be set sufficiently large. This issue is brought up and closely studied in [20], where for a certain family of "regular circuits", and bit-length of the inputs, the authors analyze a lower bound for t that ensures a correct decoding. Therefore, when selecting encryption parameters for applications, one typically needs to not only make sure that the noise does not overflow, but also that the plaintext polynomial coefficients do not grow too large. This results in a subtle optimization problem: in order to have no plaintext coefficient wrap-around, we need to choose a large t, which unfortunately implies faster noise growth (see Section 3.2). We may need to choose larger parameters overall for the encryption scheme to increase the noise ceiling and to preserve the security level. The consequence of this is worse performance.

1.2 Our Contributions

In this work we tackle the issue of the plaintext polynomial coefficient growth using a trick that Hoffstein and Silverman suggested in [29] to be used in the context of the NTRU encryption scheme [28]. Namely, they suggested replacing the modulus t with a small *polynomial* $x - b$, for some positive integer b (e.g. $b = 2$), turning the plaintext space into the integer quotient ring $\mathbb{Z}/(b^n + 1)\mathbb{Z}$. In typical parameter settings suitable for homomorphic encryption, n has size several thousands, yielding a plaintext space large enough to contain the results of many naturally occurring computations, without modular reduction ever taking place. We combine this method with the Fan-Vercauteren (FV) scheme [22], which is one of the most successful homomorphic encryption schemes to date.

In Section 3 we review the FV scheme, and present heuristic upper bounds for its noise growth in homomorphic operations. In the process, we use a new and more convenient definition for noise, which results in simpler analysis, and more uniform growth properties.

In Section 4 we describe the new (leveled) homomorphic encryption scheme, prove its correctness, and study its noise growth properties both in terms of strict and heuristic upper bounds.

In Section 6 we show how to encode rational numbers as integers in the plaintext space $\mathbb{Z}/(b^n + 1)\mathbb{Z}$, allowing the new scheme to be used to perform high-precision rational number arithmetic.

In Section 7 we discuss and the performance of the new scheme. In particular, we describe a fair and reasonable methodology for comparing it to the FV scheme. We choose to use the *Non-Adjacent Form (NAF) encoder* [16] to enable integer arithmetic in the FV scheme, as it yields some of the best performance results. We find that the new scheme significantly outperforms this FV-NAF approach when deep circuits on integers or rational numbers need to be evaluated.

In Section 8 we discuss how certain known applications of homomorphic encryption can benefit from the new scheme. In many cases, the new scheme allows much smaller parameters to be used, yielding performance, message expansion, and security level improvements.

1.3 Related Work

The idea of using the trick of Hoffstein and Silverman [29] in homomorphic encryption is by no means new: Geihs and Cabarcas [23] applied it in the context of the Brakerski-Vaikuntanathan (BV) scheme [12]. However, we note that this is much more straightforward than using it with modern schemes. For convenience, they used $b = 2$ in the modulus polynomial $x - b$, and noted that other choices might produce useful properties, such as the message space being isomorphic to a finite field, or isomorphic to a product ring in which one can use the Chinese Remainder Theorem to encode multiple plaintext integers at once. The same ideas apply in our setting, and indeed we observed that choosing b appropriately is critical for achieving the best results with the new scheme.

Lauter et al. [32] apply the idea to YASHE, but only focus on specific applications. They cite an unpublished work of López-Alt and Naehrig [35] for more details. In contrast, we present a detailed construction, noise growth analysis, performance evaluation, and comparison to the FV scheme. While [32] only encrypts integers, we describe also how to efficiently encrypt rational numbers with high precision.

There has recently been a lot of interest in the homomorphic encryption community in encrypting rational numbers more efficiently [4, 17, 7, 21]. Some researchers have even proposed homomorphic encryption schemes that encrypt true floating point numbers, while others have proposed technical improvements to existing schemes, or to previously known encoding methods, to enable more efficient fixed-precision rational number arithmetic. As encrypted floating point arithmetic is very unnatural from the point of view of the schemes, it is not surprising that the latter approaches yield substantially more efficient constructions; indeed, our solution falls into the same category, and can be thought of as a technical modification to the FV scheme.

Some approaches, such as the work of Cheon et al. [17], have substantially different properties, which makes a direct comparison less meaningful. For example, their scheme allows *batching* to be used, which results in good *amortized* performance in cases where the SIMD capabilities of the scheme can be fully utilized. However, the latency is much worse than in our scheme. This work also becomes extremely costly as the desired bit-precision increases, as do others with similar capabilities (e.g. [4]). In comparison, our scheme can more conveniently support deep circuits on high-precision inputs without any precision loss, and with much better computational performance.

Finally, it is worth noting that many of the approaches mentioned above for homomorphic encryption of integers and rational numbers are difficult to use in an optimal way, even for experts in the field, due to the large number of parameters involved in both encrypting and encoding. On the other hand, our approach has fewer parameters, making it easier to use and to optimize.

2 Notation

For n a power of 2, we denote $R = \mathbb{Z}[x]/(x^n + 1)$—the $2n$-th cyclotomic ring of integers. For an integer a, we denote $R_a = R/aR = \mathbb{Z}_a[x]/(x^n + 1)$, and $R^{\mathbb{Q}} = R \otimes \mathbb{Q} = \mathbb{Q}[x]/(x^n + 1)$.

For any polynomial in $\mathbb{Z}[x]$ (or $\mathbb{Q}[x]$) we denote the infinity norm by $\|\cdot\|$. For any polynomial in R (or R_a, $R^{\mathbb{Q}}$), we always consider the representative with lowest possible degree. We also encounter the infinity norm in the so-called canonical embedding [25, 19], and for an polynomial in R (or $R^{\mathbb{Q}}$) denote it by $\|\cdot\|^{\text{can}}$. For integers modulo $a \in \mathbb{Z}_{>0}$, we always use representatives in the symmetric interval $[-\lceil (a-1)/2 \rceil, \lfloor (a-1)/2 \rfloor]$. For any polynomial in $\mathbb{Z}[x]$, $[\cdot]_a$ denotes the coefficient-wise reduction modulo a. For any polynomial in $\mathbb{Q}[x]$ we denote rounding of the coefficients to the nearest integer by $\lfloor \cdot \rceil$.

For any polynomial $p \in \mathbb{Z}[x]$, and an integer base w, we denote the polynomials in its coefficient-wise base-w decomposition by $p^{(i)}$, where $i = 0, \ldots, \lfloor \log_w \|p\| \rfloor$.

We denote by χ a discrete Gaussian distribution having standard deviation σ, truncated at some large bound B (e.g. $B \approx 6\sigma$). The computational security parameter is denoted λ. By log we always mean $\log_2$.

Ciphertext elements considered in this work are always pairs of polynomials, e.g. $\mathtt{ct} = (c_0, c_1)$. For such a pair, and a third polynomial s, we denote $\mathtt{ct}(s) = c_0 + c_1 s$.

3 Preliminaries

As the new scheme can be thought of as a variant of the Fan-Vercauteren scheme [22], for the convenience of the reader, we include the definition and some preliminaries of the FV scheme in the full version [15].

3.1 Noise Fundamentals

As we briefly explained in Section 1.1, every ciphertext in FV carries with itself a noise component, which grows in homomorphic operations. When using leveled fully homomorphic encryption schemes, it becomes particularly important to be able to estimate the noise growth as accurately as possible. This is because only the party holding the secret key can compute the exact value of the noise, and the party performing the homomorphic evaluations must estimate the noise growth to ensure that the ciphertexts will not become corrupted. For the FV scheme, [22] presents upper bound estimates for noise growth, but these estimates are not very tight, and cannot be used for determining accurately whether specific parameters work for a specific computation. Costache and Smart [19] instead study heuristic upper bounds for the noise growth for a number of schemes, including FV. Such a heuristic analysis proves to be a powerful tool, yielding much tighter and more realistic noise growth estimates, and yields reasonable results when used for determining parameters in the leveled setting.

In Section 3.2 we will present heuristic noise growth results for the FV scheme, and in Section 5 both strict and heuristic noise growth bounds *à la* Costache-Smart for the new scheme. In Section 7 we use these heuristic results as a component in our comparison of the two schemes.

3.2 Noise in FV

In this section we present (without proof) heuristic upper bounds for noise growth in the FV scheme. For much more details on the methodology, we refer the reader to [19, 25].

The definition of noise (*invariant noise*) that we employ here is the same that is used in [31], and different from those used in e.g. [22, 33, 19].

Definition 1 (FV invariant noise). *Let* $\mathtt{ct} = (c_0, c_1)$ *be an FV ciphertext encrypting the message* $m \in R_t$*. Its invariant noise* $v \in R^{\mathbb{Q}}$ *is the polynomial with the smallest infinity norm such that*

$$\frac{t}{q}\,\mathtt{ct}(s) = \frac{t}{q}\,(c_0 + c_1 s) = m + v + at \in R^{\mathbb{Q}}\,,$$

for some polynomial $a \in R$*.*

Intuitively, Definition 1 captures the notion that the noise v being rounded incorrectly is what causes decryption failures in the FV scheme. We see this in the following lemma, which bounds the coefficients of v.

Lemma 1. *An FV ciphertext* $\mathtt{ct}$ *encrypting a message* m *decrypts correctly, as long as the invariant noise* v *satisfies* $\|v\| < 1/2$*.*

Proof. Let $\mathtt{ct} = (c_0, c_1)$. Using the formula for decryption, we have for some polynomial A:

$$m' = \left[\left\lfloor \frac{t}{q}\,[c_0 + c_1 s]_q \right\rceil\right]_t = \left[\left\lfloor \frac{t}{q}\,(c_0 + c_1 s) + At \right\rceil\right]_t = \left[\left\lfloor \frac{t}{q}\,(c_0 + c_1 s) \right\rceil\right]_t .$$

By the definition of v, $m' = [\lfloor m + v + at \rceil]_t = m + \lfloor v \rceil \pmod t$. Hence decryption is successful as long as v is removed by the rounding, i.e. if $\|v\| < 1/2$. □

The key to obtaining the heuristics is to use the infinity norm in the canonical embedding, which we call the *canonical norm* and denote $\|\cdot\|^{\text{can}}$, instead of the usual infinity norm. Discussing the canonical norm in detail is beyond the scope of this paper. The canonical norm is useful due to the following facts.

Lemma 2 ([19, 25]). *For any polynomials* $a, b \in R^{\mathbb{Q}}$*,*

$$\|a\| \le \|a\|^{can} \le \|a\|_1\,, \quad \|ab\|^{can} \le \|a\|^{can}\,\|b\|^{can}\,.$$

If $a \in R^{\mathbb{Q}}$ *has its coefficients sampled independently from a distribution with standard deviation* σ_{coeff}*, then* $\|a\|^{can} \le 6\sigma_{coeff}\sqrt{n}$*, with very high probability.*

Since the usual infinity norm is always bounded from above by the canonical norm, it suffices to ensure for correctness that the canonical norm never reaches 1/2, and therefore in the heuristic estimates all bounds are presented for the canonical norm of the noise.

The following lemmas can easily be obtained from standard noise growth arguments for FV [22], combined with Lemma 2. For more details on exactly how this is done, we refer the reader to [19].

Lemma 3 (FV initial noise heuristic). *Let* $\mathtt{ct}$ *be a fresh FV encryption of a message* $m \in R_t$*. Let* N_m *be an upper bound on the number of non-zero terms in the polynomial* m*. Let* $r_t(q)$ *denote* $q - \lfloor q/t \rfloor t$*, which is a non-negative integer less than* t*. The noise* v *in* $\mathtt{ct}$ *satisfies*

$$\|v\|^{can} \leq \frac{r_t(q)}{q}\|m\|N_m + \frac{6\sigma t}{q}\left(4\sqrt{3n} + \sqrt{n}\right),$$

with very high probability.

Lemma 4 (FV addition heuristic). *Let* $\mathtt{ct}_1$ *and* $\mathtt{ct}_2$ *be two ciphertexts encrypting* $m_1, m_2 \in R_t$*, and having noises* v_1*,* v_2*, respectively. Then the noise* v_{add} *in their sum* $\mathtt{ct}_{add}$ *satisfies* $\|v_{add}\|^{can} \leq \|v_1\|^{can} + \|v_2\|^{can}$*.*

Lemma 5 (FV multiplication heuristic). *Let* $\mathtt{ct}_1$ *be a ciphertext encrypting* m_1 *with noise* v_1*, and let* $\mathtt{ct}_2$ *be a ciphertext encrypting* m_2 *with noise* v_2*. Let* N_{m_1} *and* N_{m_2} *be upper bounds on the number of non-zero terms in the polynomials* m_1 *and* m_2*, respectively. Then with very high probability, the noise* v_{mult} *in the product* $\mathtt{ct}_{mult}$ *satisfies the following bound:*

$$\begin{aligned}\|v_{mult}\|^{can} \leq{}& \left(2\|m_1\|N_{m_1} + 6tn + t\sqrt{3n}\right)\|v_2\|^{can} \\ &+ \left(2\|m_2\|N_{m_2} + 6tn + t\sqrt{3n}\right)\|v_1\|^{can} \\ &+ 3\|v_1\|^{can}\|v_2\|^{can} + \frac{t\sqrt{3n}}{q}\cdot\frac{(12n)^{3/2}-1}{\sqrt{12n}-1} + \frac{6\sqrt{3}t}{q}n\sigma(\ell+1)w\,.\end{aligned}$$

Of the five summands appearing this formula, the first two are by far the most significant ones. The parameter w only affects the running time, so when that is not a concern we can assume it to be small. This makes the last term small compared to the first two. Since $\|m_i\| \leq t/2$, and $N_{m_i} \leq n$, we find the following simple estimate:

$$\|v_{\text{mult}}\|^{\text{can}} \lesssim 14tn \max\left\{\|v_1\|^{\text{can}}, \|v_2\|^{\text{can}}\right\}. \tag{1}$$

In this paper we are restricting our considerations to a situation where the native SIMD functionality (batching) of the scheme [41] is not used, in which case it is possible to choose the parameters so that $r_t(q) = 1$. Furthermore, in practice $\|m\| \ll t/2$ when encoding integers or rational numbers using the encoders described in [21, 14, 16, 7]. This implies that the first term in the initial noise estimate of Lemma 3 is small, yielding the following simpler estimate:

$$\|v_{\text{initial}}\|^{\text{can}} \lesssim \frac{42\sigma tn}{q}. \tag{2}$$

4 The New Scheme

4.1 Hat Encoder

Before describing the new scheme, we need to introduce a variant of the integer encoder of [14].

Let $m \in \mathcal{M}$ be a plaintext element, considered in the symmetric interval $[-\lceil b^n/2 \rceil, \lfloor b^n/2 \rfloor]$. When $b > 2$, denote by $\widehat{m}$ a polynomial whose coefficients are the (symmetric representatives of) the base-b digits of m. When $b = 2$, we use the binary digits of m, but augmented with the (repeating) sign. Note that this is exactly the integer encoding discussed in [14]. Unfortunately, only b^n consecutive integers can be represented in such a way as polynomials of degree at most $n-1$, and we are left with one plaintext integer without an obvious encoding. However, it suffices to allow the coefficients (in fact, at most one coefficient) in the encodings to have absolute value up to $(b+1)/2$. This gives more room to encode all elements of $\mathcal{M}$, but also introduces non-uniqueness in the encodings. This is not a problem, however, as evaluating any such encoding at $x = b$ yields the correct result modulo $b^n + 1$. Furthermore, will only need the fact that every element of $\mathcal{M}$ has such an encoding of length at most n, with coefficients at most $(b+1)/2$. For example, when $b = 3$ and $n = 2$, we can encode -5 as $-x-2$, but also as $-2x + 1$. For definiteness, we fix once and for all one such encoding per each element of $\mathcal{M}$.

Definition 2. *Let $m \in \mathcal{M}$. For each $m \in \mathcal{M}$ choose a shortest polynomial with $\|\widehat{m}\| \leq (b+1)/2$, such that $\widehat{m}(b) = m$ modulo $b^n + 1$, and denote it $\widehat{m}$. As was explained above, such a polynomial $\widehat{m}$ always exists, and has degree at most $n-1$.*

4.2 New (Leveled) Scheme

Let $b \geq 2$ be an integer, and define the new plaintext space $\mathcal{M} = \mathbb{Z}/(b^n + 1)\mathbb{Z}$. The parameters n, q, σ, w, ℓ, and the ring R_q are as in the FV scheme (defined in the full version [15]). The ciphertext space is the same as in FV, namely $R_q \times R_q$. We define

$$\Delta_b = \left\lfloor -\frac{q}{b^n + 1}(x^{n-1} + bx^{n-2} + \ldots + b^{n-1}) \right\rceil .$$

The polynomial Δ_b is analogous to the number Δ appearing in the FV scheme.

The following set of algorithms describes our new leveled fully homomorphic encryption scheme.

- `SecretKeyGen` : Output `sk` = `FV.SecretKeyGen`.
- `PublicKeyGen(sk)`: Output `pk` = `FV.PublicKeyGen(sk)`.
- `EvaluationKeyGen(sk)`: Output `evk` = `FV.EvaluationKeyGen(sk)`.
- `Encrypt(pk`, $m \in \mathcal{M}$`)`: Let `pk` $= (p_0, p_1)$. Sample u with coefficients uniform in $\{-1, 0, 1\}$, and $e_0, e_1 \leftarrow \chi$. Let $\widehat{m}$ be an encoding of m, as described above. Output `ct` $= ([\Delta_b \widehat{m} + p_0 u + e_0]_q, [p_1 u + e_1]_q) \in R_q \times R_q$.

- $\texttt{Decrypt}(\texttt{sk}, \texttt{ct})$: Let $s = \texttt{sk}$ and $(c_0, c_1) = (\texttt{ct}[0], \texttt{ct}[1])$. Compute $\widehat{M} = \left\lfloor \frac{x-b}{q}[c_0 + c_1 s]_q \right\rceil$. Output $m' = \widehat{M}(b) \in \mathcal{M}$.

We prove correctness of the above public-key encryption scheme in Section 4.3. Security follows from exactly the same argument as for the FV scheme [22], and is commented on in the full version [15].

For the new scheme, homomorphic addition is exactly the same as for FV:

- $\texttt{Add}(\texttt{ct}_0, \texttt{ct}_1)$: Output $\texttt{FV.Add}(\texttt{ct}_0, \texttt{ct}_1)$.

Multiplication again consists of two parts. The first part ($\texttt{Multiply}'$) forms an intermediate three-component ciphertext $\texttt{ct}'_{\text{mult}}$, just like in FV, which can be converted back to size 2 using $\texttt{FV.Relinearize}$ with $\texttt{evk}$, to form the final two-component output ciphertext $\texttt{ct}_{\text{mult}}$.

- $\texttt{Multiply}'(\texttt{ct}_0, \texttt{ct}_1)$: Denote $(c_0, c_1) = \texttt{ct}_0$ and $(d_0, d_1) = \texttt{ct}_1$. Compute $c'_0 = \left[\left\lfloor \frac{x-b}{q} c_0 d_0 \right\rceil\right]_q$, $c'_1 = \left[\left\lfloor \frac{x-b}{q}(c_0 d_1 + c_1 d_0) \right\rceil\right]_q$, and $c'_2 = \left[\left\lfloor \frac{x-b}{q} c_1 d_1 \right\rceil\right]_q$, and output $\texttt{ct}'_{\text{mult}} = (c'_0, c'_1, c'_2) \in R_q \times R_q \times R_q$.
- $\texttt{Relinearize}(\texttt{ct}', \texttt{evk})$: Output $\texttt{FV.Relinearize}(\texttt{ct}', \texttt{evk})$.
- $\texttt{Multiply}(\texttt{ct}_0, \texttt{ct}_1, \texttt{evk})$: Output $\texttt{Relinearize}(\texttt{Multiply}'(\texttt{ct}_0, \texttt{ct}_1)) \in R_q \times R_q$.

4.3 Correctness

We use the following variant of Definition 1 to analyze the performance and correctness of the public-key encryption scheme.

Definition 3 (Invariant noise). *Let $\texttt{ct} = (c_0, c_1)$ be a ciphertext encrypting the message $m \in \mathcal{M}$. Its invariant noise $v \in R^{\mathbb{Q}}$ is the polynomial with the smallest infinity norm such that*

$$\frac{x-b}{q}\texttt{ct}(s) = \frac{x-b}{q}(c_0 + c_1 s) = \widehat{m} + v + a(x-b) \in R^{\mathbb{Q}},$$

for some polynomial $a \in R$.

We now consider under what conditions decryption works correctly.

Lemma 6. *The function $\texttt{Decrypt}$, as presented in Section 4.2, correctly decrypts a ciphertext $\texttt{ct}$ encrypting a message m, as long as the invariant noise v satisfies $\|v\| < 1/2$.*

Proof. Let $\texttt{ct} = (c_0, c_1)$. Using the formula for decryption, we have for some polynomial A:

$$\begin{aligned}\widehat{M} &= \left\lfloor \frac{x-b}{q}\left[c_0 + c_1 s\right]_q \right\rceil = \left\lfloor \frac{x-b}{q}(c_0 + c_1 s + Aq) \right\rceil \\ &= \lfloor \widehat{m} + v + a(x-b) \rceil + A(x-b) = \widehat{m} + \lfloor v \rceil + (A+a)(x-b)\,.\end{aligned}$$

As long as v is removed by the rounding, i.e. if $\|v\| < 1/2$, $\texttt{Decrypt}$ outputs $m' = \widehat{M}(b) = \widehat{m}(b) = m \in \mathcal{M}$. □

Next, we prove that the noise in a fresh encryption is small enough for correct decryptions. First we need the following lemma. The proof is given in the full version [15].

Lemma 7. *Let Δ_b be as defined above. Then $\Delta_b(x - b) = q + \rho \in R^{\mathbb{Q}}$, and $\|\rho\| \leq (b+1)/2$.*

Lemma 8 (Initial noise). *Let $\mathtt{ct} = (c_0, c_1)$ be a fresh encryption of a message $m \in \mathcal{M}$. Let N_m denote an upper bound on the number of non-zero coefficients in $\widehat{m}$. The noise v in $\mathtt{ct}$ satisfies the bound*

$$\|v\| \leq \frac{1}{q}\left(\frac{b+1}{2}\right)^2 N_m + \frac{b+1}{q} B(2n+1)\,.$$

Proof. See the full version [15]. □

Note that $N_m \leq n$ in any case. We combine Lemma 6 and Lemma 8 to obtain correctness for the public-key encryption scheme.

Theorem 1. *The public-key encryption scheme defined by the algorithms* `SecretKeyGen`*,* `PublicKeyGen`*,* `Encrypt`*, and* `Decrypt`*, is correct as long as the parameters are chosen so that*

$$\frac{1}{q}\left(\frac{b+1}{2}\right)^2 n + \frac{b+1}{q} B(2n+1) < \frac{1}{2}\,.$$

□

In the remaining of this section, we present two lemmas stating the correctness of homomorphic addition and multiplication. For the proofs of the lemmas, we refer the reader to the full version [15].

Lemma 9 (Addition). *Let $\mathtt{ct}_1$ and $\mathtt{ct}_2$ be two ciphertexts encrypting $m_1, m_2 \in \mathcal{M}$, and having noises v_1, v_2, respectively. Then $\mathtt{ct}_{add} = \mathtt{Add}(\mathtt{ct}_1, \mathtt{ct}_2)$ encrypts the sum $m_1 + m_2 \in \mathcal{M}$, and has noise v_{add}, such that $\|v_{add}\| \leq \|v_1\| + \|v_2\|$.*

Lemma 10 (Multiplication). *Let $\mathtt{ct}_1$ and $\mathtt{ct}_2$ be two ciphertexts encrypting $m_1, m_2 \in \mathcal{M}$, and having noises v_1, v_2, respectively. Let N_{m_1} and N_{m_2} be upper bounds on the number of non-zero terms in the polynomials $\widehat{m_1}$ and $\widehat{m_2}$, respectively. Then $\mathtt{ct}_{mult} = \mathtt{Multiply}(\mathtt{ct}_1, \mathtt{ct}_2, \mathtt{evk})$ encrypts the product $m_1 m_2 \in \mathcal{M}$, and has noise v_{mult}, such that*

$$\begin{aligned}\|v_{mult}\| \leq\ & \frac{b+1}{2}(N_{m_1} + n^2 + 2n)\|v_2\| + \frac{b+1}{2}(N_{m_2} + n^2 + 2n)\|v_1\| \\ & + 3n\|v_1\|\|v_2\| + \frac{(b+1)B}{q}(1 + n + n^2) + \frac{b+1}{q} nB(\ell+1)w\,.\end{aligned}$$

5 Homomorphic Operations

In this section we present heuristic noise growth estimates of homomorphic addition and multiplication analogous to those in Section 3.2.

5.1 Heuristic Estimates

In this section we present heuristic upper bounds for the noise growth in the new scheme, just like we did for FV in Section 3.2, and as was motivated in Section 3.1. Again, we use the canonical norm $\|\cdot\|^{\text{can}}$ instead of the usual infinity norm $\|\cdot\|$ for the same reasons as in Section 3.2: essentially, it allows to prove much more accurate heuristic estimates for the noise growth in multiplication. We will present these results, but omit the proofs, as they are simple modifications of the proofs of Lemma 8, Lemma 9, and Lemma 10 combined with Lemma 2.

Lemma 11 (Initial noise heuristic). *Let $\mathtt{ct}$ be a fresh encryption of a message $m \in \mathcal{M}$. Let N_m denote an upper bound on the number of non-zero coefficients in $\widehat{m}$. The noise v in $\mathtt{ct}$ satisfies the bound*

$$\|v\|^{can} \leq \frac{1}{q}\left(\frac{b+1}{2}\right)^2 2\sqrt{3n}\, N_m + \frac{6\sigma(b+1)}{q}\left(4\sqrt{3}n + \sqrt{n}\right),$$

with very high probability.

Lemma 12 (Addition heuristic). *Let $\mathtt{ct}_1$ and $\mathtt{ct}_2$ be two ciphertexts encrypting $m_1, m_2 \in \mathcal{M}$, and having noises v_1, v_2, respectively. Then $\mathtt{ct}_{add} = \mathtt{Add}(\mathtt{ct}_1, \mathtt{ct}_2)$ encrypts the sum $m_1 + m_2 \in \mathcal{M}$, and has noise v_{add}, such that $\|v_{add}\|^{can} \leq \|v_1\|^{can} + \|v_2\|^{can}$.*

Lemma 13 (Multiplication heuristic). *Let $\mathtt{ct}_1$ and $\mathtt{ct}_2$ be two ciphertexts encrypting $m_1, m_2 \in \mathcal{M}$, and having noises v_1, v_2, respectively. Let N_{m_1} and N_{m_2} be upper bounds on the number of non-zero terms in the polynomials $\widehat{m_1}$ and $\widehat{m_2}$, respectively. Then*

$$\mathtt{ct}_{mult} = \mathtt{Multiply}(\mathtt{ct}_1, \mathtt{ct}_2, \mathtt{evk})$$

encrypts the product $m_1 m_2 \in \mathcal{M}$, and has noise v_{mult}, such that

$$\begin{aligned}\|v_{mult}\|^{can} \leq{}& (b+1)\left(N_{m_1} + 6n + \sqrt{3n}\right)\|v_2\|^{can} \\ &+ (b+1)\left(N_{m_2} + 6n + \sqrt{3n}\right)\|v_1\|^{can} \\ &+ 3\|v_1\|^{can}\|v_2\|^{can} + \frac{b+1}{q}\sqrt{3n}\left(1 + \sqrt{12n} + 12n\right) \\ &+ \frac{6\sqrt{3}(b+1)}{q} n\sigma(\ell+1)w\,,\end{aligned}$$

with very high probability.

Of the five summands appearing this formula, the first two are again by far the most significant ones. As before, the parameter w only affects the running time, so when that is not a concern we can assume it to be small. This makes the

last term small compared to the first two. Since $N_{m_i} \leq n$, we find the following simple estimate:

$$\|v_{\text{mult}}\|^{\text{can}} \lesssim 14(b+1)n \max\left\{\|v_1\|^{\text{can}}, \|v_2\|^{\text{can}}\right\} . \tag{3}$$

For the initial noise, we again use $N_m \leq n$ to obtain

$$\|v_{\text{initial}}\|^{\text{can}} \lesssim \frac{(b+1)^2 n^{3/2}}{q} . \tag{4}$$

6 Fractional Encoder

The *fractional encoder* introduced by Dowlin et al. in [21] (see also [14, 20]) is a convenient way of encoding and encrypting fixed-precision rational numbers, and can be used in conjunction with many RLWE-based homomorphic encryption schemes. In this section we construct a fractional encoder based on theirs to be used in conjunction with the new scheme.

6.1 Abstract Fractional Encoder

For the new scheme, and in fact for any homomorphic encryption scheme whose plaintext space is a ring $\mathcal{M}$, we can abstract out the functionality of encoding fractional numbers as a triple $(\mathcal{P}, \texttt{Encode}, \texttt{Decode})$, where $\mathcal{P}$ is a finite subset of $\mathbb{Q}$, and

$$\texttt{Encode} : \mathcal{P} \to \mathcal{M} , \quad \texttt{Decode} : \texttt{Encode}(\mathcal{P}) \to \mathcal{P}$$

are maps satisfying $\texttt{Decode}(\texttt{Encode}(x)) = x$, for all $x \in \mathcal{P}$.

To preserve the homomorphic property, we additionally require that when $x, y, x+y, xy \in \mathcal{P}$, then

$$\begin{aligned} \texttt{Encode}(x+y) &= \texttt{Encode}(x) + \texttt{Encode}(y), \\ \texttt{Encode}(xy) &= \texttt{Encode}(x)\texttt{Encode}(y) . \end{aligned}$$

In our case we have $\mathcal{M} = \mathbb{Z}/(b^n+1)\mathbb{Z}$, so a natural candidate for a fractional encoding map that satisfies the homomorphic properties would be

$$\texttt{Encode} : \mathcal{P} \to \mathcal{M}, \quad \texttt{Encode}\left(\frac{x}{y}\right) = xy^{-1} \mod (b^n+1) . \tag{5}$$

However, $\mathcal{P}$ needs to chosen carefully to make this map both well-defined and injective. For example, it is clearly undefined when $\gcd(y, b^n+1) > 1$. We resolve these issues below, presenting appropriate choices for $\mathcal{P}$.

6.2 Case of Odd b

When b is odd, we prove that

$$\mathcal{P} = \left\{ c + \frac{d}{b^{n/2}} : c, d \in \left[-\frac{b^{n/2}-1}{2}, \frac{b^{n/2}-1}{2} \right] \cap \mathbb{Z} \right\}$$

makes the map `Encode` presented above well-defined and injective, and thus invertible in its range.

Lemma 14. *The map* $\texttt{Encode} : \mathcal{P} \to \mathcal{M}$ *in* (5) *is injective.*

Proof. Suppose $c + d/b^{n/2} = c' + d'/b^{n/2} \mod (b^n + 1)$. Then $(c - c')b^{n/2} + (d - d') = k(b^n + 1)$ for some integer k. However, we have

$$\left| (c - c')b^{n/2} + (d - d') \right| \leq (b^{n/2} - 1)b^{n/2} + (b^{n/2} - 1) = b^n - 1 < b^n + 1 \,.$$

Thus $k = 0$, and $cb^{n/2} + d = c'b^{n/2} + d'$. Dividing both sides by $b^{n/2}$ proves the claim. □

We define `Decode` as the left inverse of `Encode` in its range. We derive a simple description for `Decode` below. As usual, $[y]_a$ denotes reduction of the integer y modulo a in the symmetric interval $[-\lceil (a-1)/2 \rceil, \lfloor (a-1)/2 \rfloor]$.

Lemma 15. *For* $z \in \texttt{Encode}(\mathcal{P})$, $\texttt{Decode}(z) = b^{-n/2}[zb^{n/2}]_{b^n+1}$.

Proof. Assume $z = \texttt{Encode}(y)$, with $y = c + d/b^{n/2}$. By definition of `Encode`, $zb^{n/2} = yb^{n/2} = cb^{n/2} + d \mod (b^n + 1)$. It follows from definition of $\mathcal{P}$, that $|cb^{n/2} + d| \leq (b^n - 1)/2$. Hence $[zb^{n/2}]_{b^n+1} = cb^{n/2} + d$, and dividing both sides by $b^{n/2}$ yields the result. □

6.3 Case of Even b

When b is odd, we can encode fractions with $n/2$ integral base-b digits, and $n/2$ fractional base-b digits. When b is even, due to technical constraints, we need to reduce either the number of fractional digits or the number of integral digits by one. Suppose we reduce the number of fractional digits by one, and set

$$\mathcal{P} = \left\{ c + \frac{d}{b^{n/2-1}} : |c| \leq \frac{(b^{n/2}-1)b}{2(b-1)}, |d| \leq \frac{(b^{n/2-1}-1)b}{2(b-1)}, c, d \in \mathbb{Z} \right\} .$$

We prove that this makes the map `Encode` presented above well-defined and injective, and thus invertible in its range.

Lemma 16. *The map* $\texttt{Encode} : \mathcal{P} \to \mathcal{M}$ *in* (5) *is injective.*

Proof. Suppose $c+d/b^{n/2-1} = c'+d'/b^{n/2-1} \mod (b^n+1)$. Then $(c-c')b^{n/2-1}+(d-d') = k(b^n+1)$ for some integer k. However, we have

$$\begin{aligned}\left|(c-c')b^{n/2-1}+(d-d')\right| &\leq \frac{b}{b-1}\left[(b^{n/2}-1)b^{n/2-1}+b^{n/2-1}-1\right] \\ &= \frac{b}{b-1}\left(b^{n-1}-1\right) \leq b^n - b < b^n+1.\end{aligned}$$

Thus $k=0$, and $cb^{n/2-1}+d = c'b^{n/2-1}+d'$. Dividing both sides by $b^{n/2-1}$ proves the claim. □

Note that if we do not reduce the number of digits by one, then Lemma 16 might fail. Namely, if we have $n/2$ digits for both the integral and fractional parts, then the equation in the proof becomes $(c-c')b^{n/2}+(d-d') = k(b^n+1)$, and the inequality becomes

$$\left|(c-c')b^{n/2}+(d-d')\right| \leq \frac{b}{b-1}(b^n-1)\,,$$

where the right-hand side can now be greater than or equal to b^n+1.

We now derive a simple expression for `Decode`.

Lemma 17. *For $z \in \mathtt{Encode}(\mathcal{P})$, $\mathtt{Decode}(z) = b^{-(n/2-1)}[zb^{n/2-1}]_{b^n+1}$.*

Proof. Assume $z = \mathtt{Encode}(y)$, with $y = c + d/b^{n/2-1}$. By definition of `Encode`, $zb^{n/2-1} = yb^{n/2-1} = cb^{n/2-1}+d \mod (b^n+1)$. It follows from the definition of $\mathcal{P}$, that

$$\left|cb^{n/2-1}+d\right| \leq \frac{b^n-b}{2(b-1)} < \frac{b^n+1}{2}\,.$$

Hence $[zb^{n/2-1}]_{b^n+1} = cb^{n/2-1}+d$, and dividing both sides by $b^{n/2-1}$ yields the result. □

As an example, let $n=8$, $b=10$, and $y=12.55$. Since $100^{-1} = -10^6 \mod (10^8+1)$, $z = \mathtt{Encode}(y) = \left[-1255\cdot 10^6\right]_{10^8+1} = 45000013$. For the purposes of encryption, we need to also compute the polynomial encoding $\widehat{z} = -5x^7-5x^6+x+2$. Decryption evaluates this polynomial (or—more correctly—a polynomial equal to it modulo $x-10$) at $x=10$. Of course, this gives back the number $45000013 \mod (10^8+1)$, which decoding converts to

$$\mathtt{Decode}(z) = \frac{\left[45000013\cdot 10^3\right]_{10^8+1}}{10^3} = 12.55\,.$$

7 Comparison to FV

In this section we present a performance comparison of the new scheme with the FV scheme. Since the schemes have very different properties, how such a comparison should be performed in a fair and realistic way is not immediately obvious. Thus, we start by describing and motivating the methodology, after which we present the comparison, and finally summarize the results.

7.1 Methodology

To make a comparison of FV and the new scheme meaningful, we need to fix on a specific computational task, which both schemes can perform reasonably well. For such a task, we choose the evaluation of a "regular circuit", as described in [20]. Such a regular circuit is parametrized by three integers A, D, and L, and consists of evaluating A levels of additions, followed by one level of multiplication, iterated D times. The inputs to the circuit are integers in the interval $[-L, L]$. Note that such a regular circuit has (multiplicative) depth D. For a fair comparison, and to illustrate the different cases, we consider $A \in \{0, 3, 10\}$, with inputs of size $L \in \{2^8, 2^{16}, 2^{32}, 2^{64}, 2^{128}\}$, and try to find the largest possible D.

Since FV does not natively encrypt integers, we choose to use the NAF encoder [16], which performs better than the integer encoders of [14]. The main challenge with using FV is the plaintext polynomial coefficient growth, which quickly forces a very large t to be used, causing faster noise growth, and subsequently restricting the depth of the circuits. In all settings that we considered, we did not get even close to filling the plaintext polynomial space up to the top coefficient. Since the only advantage of using a higher base (as in [14]) in the encoding process is that the encodings are shorter, we are not losing anything by restricting to the NAF encoder.

Since the security of FV and the new scheme are based on exactly the same parameters, it suffices to fix σ, and settle on a set of pairs (n, q) with desired security properties. We choose to use the parameter sets presented in [14], which are estimated [3] to have a high security level[5]. We also include a set that is one step larger than these, namely $(n = 32768, q \approx 2^{890})$, as such parameter sizes can still be considered practical. For all parameters we use $\sigma = 3.19$, which is a standard choice [34, 14].

Having all of the above settled, the strategy is fairly simple. We use the heuristic upper bound estimates for noise growth, as presented in Section 3.2 for FV, and in Section 5.1 for the new scheme, to find optimal tuples (t, D) for FV, and tuples (b, D) for the new scheme, such that the depth D of the regular circuit is maximized, while ensuring correctness. Next, we discuss the inequalities imposed by these constraints for both schemes.

FV. Using (2), (1), and Lemma 4, we can bound the noise after the evaluation of a regular circuit with parameters A and D by (approximately)

$$\left(14tn\,2^A\right)^D \frac{42\sigma tn}{q}\,.$$

For correctness, this needs to be less than $1/2$, which gives us the heuristic depth estimate

$$D \lesssim \left\lfloor \frac{\log q - \log(84\sigma tn)}{\log(14tn) + A} \right\rfloor . \tag{6}$$

[5] In this paper, all estimates of the security level λ were obtained using commit `cc5f6e8` of the LWE estimator [3] which considers the most recent attacks, e.g. [1, 2].

We use the analysis of [16] (see also [20]) to bound the coefficient growth in the plaintext polynomials. One can show that the length of the NAF encoding of integers of absolute value up to L is bounded by $\lfloor \log L \rfloor + 2$, of which at most $d = \lceil (\lfloor \log L \rfloor + 2)/2 \rceil$ are non-zero. For correct decoding, [16] proves that we need

$$\sqrt{\frac{6}{\pi 2^D d(d+2)}}\,(d+1)^{2^D} 2^{A(2^{D+1}-2)} < t/2\,. \tag{7}$$

We also need to ensure that the plaintext polynomial does not wrap around $x^n + 1$, resulting in the condition $(\lfloor \log L \rfloor + 2) \cdot 2^D \leq n - 1$, but this bound has no effect in any of the experiments we run, as was already pointed out in Section 7.1, and can easily be verified from the results. It therefore suffices to search for a t, that yields a maximum depth D, satisfying only the coefficient growth condition (7), and the noise condition (6).

New scheme. For the new scheme, using (4), (3), and Lemma 12, we can bound the noise after the evaluation of a regular circuit with parameters A and D by (approximately)

$$\left(14(b+1)n\,2^A\right)^D \frac{(b+1)^2 n^{3/2}}{q}\,.$$

For correctness, this needs to be less than $1/2$, which gives us the heuristic depth estimate

$$D \lesssim \left\lfloor \frac{\log q - \log\left(2(b+1)^2 n^{3/2}\right)}{\log(14(b+1)n) + A} \right\rfloor . \tag{8}$$

We also get a restriction from the plaintext wrapping around $b^n + 1$. The output of the regular circuit has absolute value bounded by (see [20]) $V = L^{2^D} 2^{A(2^{D+1}-2)}$, so for correctness it is necessary that $V \leq (b^n - 1)/2$, which yields

$$D \lesssim \left\lfloor \log\left(\frac{\log\left((b^n-1)2^{2A-1}\right)}{\log\left(2^{2A}L\right)} \right) \right\rfloor \approx \left\lfloor \log\left(\frac{n\log b + 2A - 1}{2A + \log L} \right) \right\rfloor . \tag{9}$$

Combining (9) with the noise condition (8) yields, for a fixed b, the overall bound

$$D \lesssim \min\left\{ \left\lfloor \log\left(\frac{n\log b + 2A - 1}{2A + \log L} \right) \right\rfloor , \left\lfloor \frac{\log q - \log\left(2(b+1)^2 n^{3/2}\right)}{\log(14(b+1)n) + A} \right\rfloor \right\} .$$

7.2 Results

Our results for maximizing D are summarized in Fig. 1, and presented in more detail in the full version [15]. These results show that, for performing encrypted arithmetic on both small and large integers, the new scheme significantly outperforms the FV scheme with the NAF encoding. The difference becomes particularly strong when more additions are performed at each level, as FV suffers

from the coefficient growth resulting from these multiplications. For example, when $A = 10$ the FV scheme allows us to evaluate regular circuits of depth at most 3, even with the smallest input size that we considered, whereas with the new scheme we can go up to depth 15; this is a massive increase in performance.

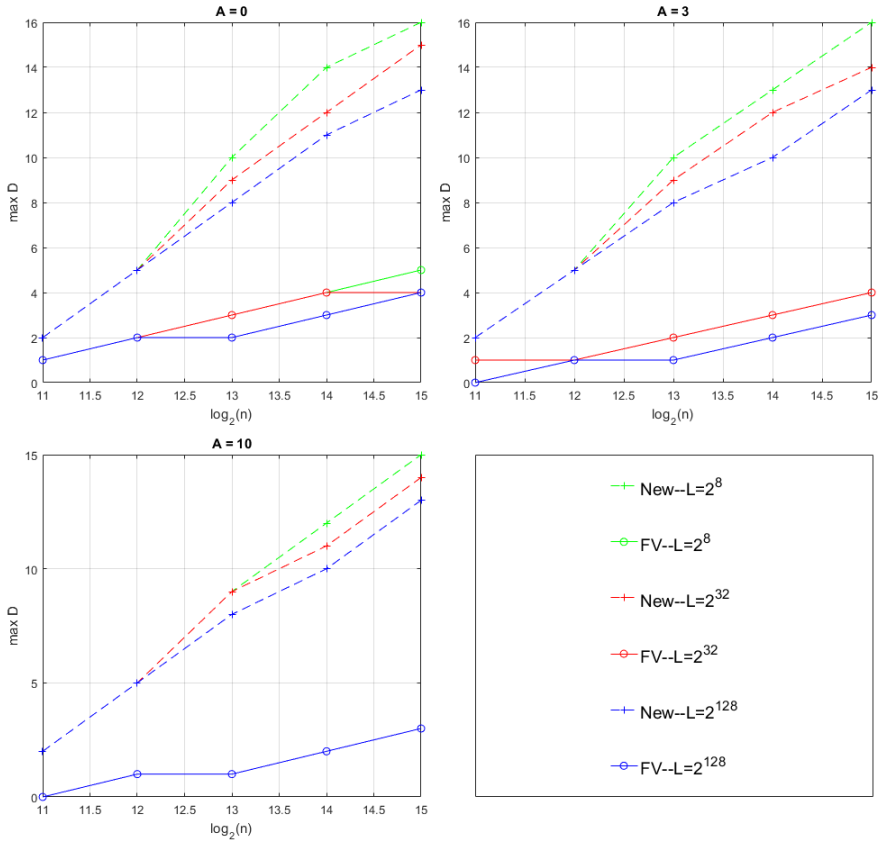

Fig. 1. Comparing maximum depth D between the FV scheme with NAF encoding, and the new scheme; at each level the circuit has 2^A additions followed by a multiplication. Results are given for $A \in \{0, 3, 10\}$, and input sizes $L \in \{2^8, 2^{32}, 2^{128}\}$.

We would also like to point out that the parameters we used in our comparison are estimated [3] to have a very high security level against the most recent attacks. In some sense, the new scheme will perform *better* in comparison to FV when using lower-security parameters: for a fixed n and σ, a lower security level corresponds to using a larger q, which has a smaller initial noise. Thus, there is more room for homomorphic operations noise-wise. This is in many cases great for the new scheme, allowing deeper circuits to be evaluated. In the FV

scheme, increasing the depth requires t to be substantially larger, which directly affects the noise growth in homomorphic multiplications, and quickly makes any increase in the noise ceiling irrelevant.

7.3 Rational Number Arithmetic

Even though the comparison above focused on integer arithmetic, a generalization to rational number inputs, with a generalization of the NAF or other integer encoders being used with the FV scheme, would yield similar results. The reason for this is explained in detail in [20]: integer operations on scaled plaintexts are essentially equivalent to performing computations using the fractional encoders, including the one described in Section 6. The difference between scaling to integers and using fractional encoders is very minor, and is explained in [14]. Instead, the benefit of using fractional encoders is mostly for convenience, as it frees the user from having to keep track of different scaling factors. Thus, the performance of integer arithmetic is exactly the same as the performance of rational number arithmetic. For example, computations on 64-bit integer inputs has the same performance as computations on rational numbers with e.g. 32-bit fractional and 32-bit integral parts.

8 Applications

The applications of homomorphic encryption on integral or rational number data are numerous. Recently, several papers have discussed applications to medical risk prediction [10], genomic analysis [32, 16], evaluating neural networks on encrypted images [27], and performing predictive analysis on power consumption in smart grids [8, 7]. A common challenge in works of this type is the growth of the plaintext polynomial coefficients, which is commonly solved either by increasing all of the parameters, or by using several smaller relatively prime plaintext polynomial coefficient moduli, and performing the computations separately using each of these: the final result can then be obtained using the Chinese Remainder Theorem coefficient-wise in the plaintext space (e.g. [27, 8]). However, with the new scheme, the situation is much better. We illustrate this by discussing the works [32] and [16]. Further examples can be found in the full version [15].

The works [32] and [16] implement medical risk prediction tasks using logistic regression, and the Cox Proportional Hazard model. Both models require non-polynomial functions to be evaluated, which the authors solve by using Taylor [32] and minimax [16] approximations. For example, for evaluating logistic regression models, [16] uses polynomials up to degree 11 evaluated on high-precision rational number inputs. This forces them to use very large parameters: their polynomial modulus has degree 23430, yielding an acceptable estimated security level $\lambda \approx 113$. With the new scheme such computations can be done easily with only $n = 4096$, and an estimated security level of $\lambda \approx 120$.

References

[1] M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In *EUROCRYPT, Part II*, pages 103–129, 2017.

[2] M. R. Albrecht, F. Göpfert, F. Virdia, and T. Wunderer. Revisiting the expected cost of solving uSVP and applications to LWE. In *ASIACRYPT, Part I*, pages 297–322, 2017.

[3] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[4] S. Arita and S. Nakasato. Fully homomorphic encryption for point numbers. In *Inscrypt*, pages 253–270, 2016.

[5] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter, and M. Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192, 2015.

[6] F. Benhamouda, T. Lepoint, C. Mathieu, and H. Zhou. Optimization of bootstrapping in circuits. In *SODA*, pages 2423–2433, 2017.

[7] C. Bonte, C. Bootland, J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren. Faster homomorphic function evaluation using non-integral base encoding. In *CHES*, pages 579–600, 2017.

[8] J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren. Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In *AFRICACRYPT*, pages 184–201, 2017.

[9] J. W. Bos, K. E. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In *IMACC*, pages 45–64, 2013.

[10] J. W. Bos, K. E. Lauter, and M. Naehrig. Private predictive analysis on encrypted medical data. *Journal of biomedical informatics*, 50:234–243, 2014.

[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325, 2012.

[12] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In *CRYPTO*, pages 505–524, 2011.

[13] M. Brenner and K. Rohloff, editors. *Proceedings of WAHC'17 - 5th Workshop on Encrypted Computing and Applied Homomorphic Cryptography*, 2017.

[14] H. Chen, K. Laine, and R. Player. Simple Encrypted Arithmetic Library - SEAL. In Brenner and Rohloff [13].

[15] H. Chen, K. Laine, R. Player, and Y. Xia. High-precision arithmetic in homomorphic encryption. Cryptology ePrint Archive, Report 2017/809, 2017.

[16] J. H. Cheon, J. Jeong, J. Lee, and K. Lee. Privacy-preserving computations of predictive medical models with minimax approximation and Non-Adjacent Form. In Brenner and Rohloff [13].

[17] J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT, Part I*, pages 409–437, 2017.

[18] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *ASIACRYPT, Part I*, pages 3–33, 2016.

[19] A. Costache and N. P. Smart. Which ring based somewhat homomorphic encryption scheme is best? In *CT-RSA*, pages 325–340, 2016.

[20] A. Costache, N.P. Smart, S. Vivek, and A. Waller. Fixed point arithmetic in SHE scheme. In *SAC*, pages 401–422, 2016.

[21] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, 105(3):552–567, 2017.

[22] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012.

[23] M. Geihs and D. Cabarcas. Efficient integer encoding for homomorphic encryption via ring isomorphisms. In *LATINCRYPT*, pages 48–63, 2014.

[24] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[25] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the AES circuit. In *CRYPTO*, pages 850–867, 2012.

[26] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO, Part I*, pages 75–92, 2013.

[27] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*, pages 201–210, 2016.

[28] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *Proceedings of Algorithmic Number Theory, Third International Symposium*, pages 267–288, 1998.

[29] J. Hoffstein and J. Silverman. Optimizations for NTRU. Public-Key Cryptography and Computational Number Theory (Proceedings of the International Conference), 2001. Available at: `https://assets.securityinnovation.com/static/downloads/NTRU/resources/TECH_ARTICLE_OPT.pdf`.

[30] A. Khedr, G. Gulak, and V. Vaikuntanathan. SHIELD: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Transactions on Computers*, 65(9):2848–2858, 2016.

[31] K. Laine, H. Chen, and R. Player. Simple Encrypted Arithmetic Library - SEAL v2.2. Technical report, 2017.

[32] K. E. Lauter, A. López-Alt, and M. Naehrig. Private computation on encrypted genomic data. In *LATINCRYPT*, pages 3–27, 2014.

[33] T. Lepoint and M. Naehrig. A comparison of the homomorphic encryption schemes FV and YASHE. In *AFRICACRYPT*, pages 318–335, 2014.

[34] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, pages 319–339, 2011.

[35] A. López-Alt and M. Naehrig. Large integer plaintexts in ring-based fully homomorphic encryption, 2014. Unpublished.

[36] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013.

[37] C. Aguilar Melchor, J. Barrier, S. Guelton, A. Guinet, M.-O. Killijian, and T. Lepoint. NFLlib: NTT-Based Fast Lattice Library. In *CT-RSA*, pages 341–356, 2016.

[38] M. Naehrig, K. E. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In *CCSW*, pages 113–124, 2011.

[39] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[40] R. L. Rivest, L. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[41] N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *Designs, codes and cryptography*, 71(1):57–81, 2014.