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 ring $\mathbb{Z}_t[x]/(x^n+1)$, 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 [51], the existence of such encryption schemes was an open problem for decades, and was solved only in 2009 by Craig Gentry [31], with an explicit construction based on ideal lattices. While the scheme of [31] was impractical, a long list of vastly more efficient schemes have since emerged [16, 17, 13, 14, 29, 45, 12, 33]. Several lines of research have focused on improving the efficiency of homomorphic encryption for practical tasks, e.g. by improving the data representations [49, 32, 54, 27, 22], and by providing clever optimization tricks to improve the performance of existing schemes both from a theoretical [32, 7, 38] and a software engineering [48, 38] 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) [50] or the Ring Learning With Errors (RLWE) [46] 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 [31] 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 [28, 24, 20, 8], 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* [6]—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 [49, 11, 42, 27, 25, 22], 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 [25], 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 3.3). 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 [36] to be used in the context of the NTRU encryption scheme [35]. 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 [29], 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* [22] 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. Our results are presented in Table 2 (for FV) and Table 3 (for the new scheme), and summarized in Figure 1.

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 [36] in homomorphic encryption is by no means new: Geihs and Cabarcas [30] applied it in the context of the Brakerski-Vaikuntanathan (BV) scheme [17]. 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 we show that choosing b appropriately is critical for achieving the best results with the new scheme (see Table 3).

Lauter *et al.* [42], citing an unpublished work of López-Alt and Naehrig [44], use a similar variant of the YASHE scheme [12], which relies on non-standard assumptions that were recently attacked in [2, 21, 40]. We apply [36] to the FV scheme, which relies only on the RLWE assumption [46]. Moreover, in contrast to [42] which mostly focuses on specific applications, we present a detailed construction, noise growth analysis, performance evaluation, and comparison to the FV scheme. While [42] 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 [5, 37, 23, 9, 27]. 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.* [23], 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. [5]). 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 [32, 26], 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

3.1 Fan-Vercauteren Scheme

As the new scheme can be thought of as a variant of the Fan-Vercauteren scheme [29], for the reader to understand and appreciate the differences, we recall the definition of the FV scheme here.

In the FV scheme the plaintext space is the R_t, and the ciphertext space is the product ring $R_q \times R_q$. The reader should assume $t \ll q$, which is the case for nearly all useful parameter choices. The degree n in the polynomial modulus $x^n + 1$ is a power of 2—typically at least 1024. The standard deviation σ of χ is often in practice chosen rather small; $\sigma \approx 3.19$ is a common choice [43]. We denote $\Delta = \lfloor q/t \rfloor$, so that $q = \Delta t + r_t(q)$ for some $r_t(q) < t$. We take $w \geq 2$ an integer—typically a power of 2 for performance reasons—which is used for coefficient-wise base-w decompositions of polynomials, and denote $\ell = \lfloor \log_w q \rfloor$.

The security of the FV scheme is based on the hardness of the decisional RLWE problem [46, 29], which is by now a standard building block of homomorphic encryption schemes, and states essentially that given a fixed $s \leftarrow \chi$, the following two distributions are computationally indistinguishable: the distribution of pairs $(a, b = as + e) \in R_q \times R_q$, where $a \leftarrow R_q$, and $e \leftarrow \chi$, and the

distribution of uniformly sampled pairs $(a, b) \leftarrow R_q \times R_q$. In practice, for performance and noise growth reasons, many implementations instead use a "small secret" variant and sample the coefficients of s from a narrow distribution, e.g. uniformly from $\{-1, 0, 1\}$. This was suggested as an optimization in [29], and we will use it also in this work. Although more attacks apply in this setting (e.g. [1]), there are theoretical results showing that certain small secret RLWE variants are as hard as those with $s \leftarrow \chi$, if the dimension n is increased sufficiently [15]. For a fixed σ, the security level λ is determined mainly by n and q (for fixed n, smaller q means higher security), and can be estimated using for example the methods described in [3].

The following set of algorithms describes the leveled fully homomorphic variant of the FV scheme.

- `FV.SecretKeyGen` : Sample $s \in R$ with coefficients uniform in $\{-1, 0, 1\}$. Output
$$\mathtt{sk} = s\,.$$

- `FV.PublicKeyGen(sk)`: Let $s = \mathtt{sk}$. Sample $a \leftarrow R_q$, and $e \leftarrow \chi$. Output
$$\mathtt{pk} = ([-(as + e)]_q, a) \in R_q \times R_q\,.$$

- `FV.EvaluationKeyGen(sk)`: For $i = 0, \ldots, \ell$, sample $a_i \leftarrow R_q$, and $e_i \leftarrow \chi$. Output the vector of pairs
$$\mathtt{evk} = [([-(a_i s + e_i) + w^i s^2]_q, a_i) \in R_q \times R_q : i = 0, \ldots, \ell]\,.$$

- `FV.Encrypt(pk, m` $\in R_t$`)`: Let $\mathtt{pk} = (p_0, p_1)$. Sample u with coefficients uniform in $\{-1, 0, 1\}$, and $e_0, e_1 \leftarrow \chi$. Output
$$\mathtt{ct} = ([\Delta m + p_0 u + e_0]_q, [p_1 u + e_1]_q) \in R_q \times R_q\,.$$

- `FV.Decrypt(sk, ct)`: Let $s = \mathtt{sk}$, $c_0 = \mathtt{ct}[0]$, and $c_1 = \mathtt{ct}[1]$. Output
$$\left[\left\lfloor \frac{t}{q}[c_0 + c_1 s]_q \right\rceil\right]_t \in R_t\,.$$

The correctness of the above public-key encryption scheme is proved in [29], and its security follows from a simple indistinguishability argument, relying on the hardness of the decision-RLWE problem [29, 46].

We next describe the homomorphic operations. Addition is easy:

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

$$(\texttt{ct}_0[0] + \texttt{ct}_1[0], \texttt{ct}_0[1] + \texttt{ct}_1[1]) \in R_q \times R_q \,.$$

Multiplication is more complicated, and consists of two parts. The first part ($\texttt{FV.Multiply}'$) forms an intermediate three-component ciphertext $\texttt{ct}'_{\text{mult}}$. While in fact the three-component ciphertext can be easily decrypted with an extension of the $\texttt{FV.Decrypt}$ method described above, it is standard to instead employ a *key switching* method to reduce the size of the ciphertext back to 2. Thus, the second part ($\texttt{FV.Relinearize}$) converts $\texttt{ct}'_{\text{mult}}$ to the final two-component output ciphertext $\texttt{ct}_{\text{mult}}$ using the evaluation key $\texttt{evk}$.

- $\texttt{FV.Multiply}'(\texttt{ct}_0, \texttt{ct}_1)$: Denote $(c_0, c_1) = \texttt{ct}_0$ and $(d_0, d_1) = \texttt{ct}_1$. Compute

$$c'_0 = \left[\left\lfloor \frac{t}{q} c_0 d_0 \right\rceil\right]_q, \quad c'_1 = \left[\left\lfloor \frac{t}{q} (c_0 d_1 + c_1 d_0) \right\rceil\right]_q, \quad c'_2 = \left[\left\lfloor \frac{t}{q} c_1 d_1 \right\rceil\right]_q,$$

and output

$$\texttt{ct}'_{\text{mult}} = (c'_0, c'_1, c'_2) \in R_q \times R_q \times R_q \,.$$

- $\texttt{FV.Relinearize}(\texttt{ct}', \texttt{evk})$: Denote $(c'_0, c'_1, c'_2) = \texttt{ct}'$. Express c'_2 in base w, so that $c'_2 = \sum_{i=0}^{\ell} {c'_2}^{(i)} w^i$. Set

$$c_0 = c'_0 + \sum_{i=0}^{\ell} \texttt{evk}[i][0] {c'_2}^{(i)}, \quad c_1 = c'_1 + \sum_{i=0}^{\ell} \texttt{evk}[i][1] {c'_2}^{(i)},$$

and output

$$(c_0, c_1) \in R_q \times R_q \,.$$

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

$$\texttt{FV.Relinearize}(\texttt{FV.Multiply}'(\texttt{ct}_0, \texttt{ct}_1), \texttt{evk}) \in R_q \times R_q \,.$$

3.2 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, [29] 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 [26] 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 3.3 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.

3.3 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 [26, 32].

The definition of noise (*invariant noise*) that we employ here is the same that is used in [41], and different from those used in e.g. [29, 26].

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 $\|v\| < 1/2$. □

Strict upper bound estimates for the noise growth in homomorphic operations are presented in [41], and here we will only present the heuristics. 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 ([26, 32]). *For any polynomials $a, b \in R^{\mathbb{Q}}$,*

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

If $a \in R^{\mathbb{Q}}$ has its coefficients sampled independently from a distribution with standard deviation σ_{coeff}, then $\|a\|^{can} \leq 6\sigma_{coeff}\sqrt{n}$, 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 [29], combined with Lemma 2. For more details on exactly how this is done, we refer the reader to [26].

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. 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 the noise v_{mult} in the product $\mathtt{ct}_{mult}$ satisfies the following bound:*

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

with very high probability.

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 [54] 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 [27, 19, 22, 9]. 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 [19].

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 [19]. 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 (Section 3.1).

The ciphertext space is the same as in FV, namely $R_q \times R_q$. We define

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

which is analogous to Δ in Section 3.1, as we will explain in Section 4.3.

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

- `SecretKeyGen` : Output

$$\texttt{sk} = \texttt{FV.SecretKeyGen}\,.$$

- `PublicKeyGen`(`sk`): Output

$$\texttt{pk} = \texttt{FV.PublicKeyGen}(\texttt{sk})\,.$$

- `EvaluationKeyGen`(`sk`): Output

$$\texttt{evk} = \texttt{FV.EvaluationKeyGen}(\texttt{sk})\,.$$

- `Encrypt`($\texttt{pk}, m \in \mathcal{M}$): Let $\texttt{pk} = (p_0, p_1)$. 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

$$\texttt{ct} = ([\Delta_b \widehat{m} + p_0 u + e_0]_q, [p_1 u + e_1]_q) \in R_q \times R_q\,.$$

- `Decrypt`(`sk`, `ct`): Let $s = \texttt{sk}$, $c_0 = \texttt{ct}[0]$, and $c_1 = \texttt{ct}[1]$. Let

$$\widehat{M} = \left\lfloor \frac{x-b}{q}[c_0 + c_1 s]_q \right\rceil .$$

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, and will be commented on in Section 4.3.

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

- `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 `FV.Relinearize` with `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

$$c'_0 = \left[\left\lfloor \frac{x-b}{q} c_0 d_0 \right\rceil\right]_q, \quad c'_1 = \left[\left\lfloor \frac{x-b}{q} (c_0 d_1 + c_1 d_0) \right\rceil\right]_q,$$

$$c'_2 = \left[\left\lfloor \frac{x-b}{q} c_1 d_1 \right\rceil\right]_q,$$

  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 `Decrypt`, as presented in Section 4.2, correctly decrypts a ciphertext `ct` encrypting a message m, 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:

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

As long as v is removed by the rounding, i.e. if $\|v\| < 1/2$, `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. To this end, we recall the definition of Δ_b, and prove the following analogue of $q = \Delta t + r_t(q)$:

Lemma 7. *With*

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

$\Delta_b(x-b) = q + \rho \in R^{\mathbb{Q}}$*, and* $\|\rho\| \leq (b+1)/2$.

Proof. The proof is a straightforward computation. For some polynomial ϵ, with $\|\epsilon\| \leq 1/2$,

$$\begin{aligned}\Delta_b(x-b) &= -\frac{q}{b^n+1}(x-b)(x^{n-1} + bx^{n-2} + \ldots + b^{n-1}) + \epsilon(x-b) \\ &= -\frac{q}{b^n+1}(x^n - b^n) + \epsilon(x-b) \\ &= \frac{q}{b^n+1}\left(b^n + 1 - (x^n+1)\right) + \epsilon(x-b) \\ &= q + \epsilon(x-b) - \frac{q}{b^n+1}(x^n+1)\,.\end{aligned}$$

Thus, $\Delta_b(x-b) = q + \rho$ in $R^{\mathbb{Q}}$, where $\rho = \epsilon(x-b)$. The bound $\|\rho\| \leq (b+1)/2$ is clear. □

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* `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. Let $\mathtt{ct} = (c_0, c_1)$ be an encryption of m under the public key $\mathtt{pk} = (p_0, p_1) = ([-(as+e)]_q, a)$. Then, for some polynomials k_0, k_1, k,

$$\begin{aligned}
\frac{x-b}{q}(c_0 + c_1 s) &= \frac{x-b}{q}\left(\Delta_b \widehat{m} + p_0 u + e_0 + k_0 q + p_1 u s + e_1 s + k_1 q s\right) \\
&= \widehat{m} + \frac{\rho\widehat{m}}{q} + \frac{x-b}{q}\left(p_0 u + e_0 + p_1 u s + e_1 s\right) \\
&\quad + (x-b)(k_0 + k_1 s) \\
&= \widehat{m} + \frac{\rho\widehat{m}}{q} + \frac{x-b}{q}\left((-as - e + kq)u + e_0 + a u s + e_1 s\right) \\
&\quad + (x-b)(k_0 + k_1 s) \\
&= \widehat{m} + \frac{\rho\widehat{m}}{q} + \frac{x-b}{q}\left(-eu + e_1 + e_2 s\right) \\
&\quad + (x-b)(k_0 + k_1 s + ku)\,,
\end{aligned}$$

so the noise is

$$v = \frac{\rho\widehat{m}}{q} + \frac{x-b}{q}\left(-eu + e_1 + e_2 s\right)\,.$$

To bound $\|v\|$, we use Lemma 7, that the error polynomials sampled from χ have coefficients bounded by B, and that $\|s\| = \|u\| = 1$:

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

□

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*

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

□

4.4 Security

The security argument for the new scheme is exactly the same as for the FV scheme. Namely, the public key is indistinguishable from uniform due to the decision-RLWE assumption [46]. Ciphertexts are indistinguishable from uniform due to a two-layered decision-RLWE assumption, where the uniformity of the public key is used together with the decision-RLWE assumption to hide the

message. Thus, one can prove that the scheme is secure if the 2-sample (small-secret) decision-RLWE problem is hard. The evaluation key does introduce a standard circular security assumption, as is discussed in more detail in [29]. For much more details on the security argument, we refer the reader to [47].

5 Homomorphic Operations

In this section we prove the correctness of homomorphic addition and multiplication, and describe the noise growth bounds for the new scheme. We also present heuristic noise growth estimates analogous to those in Section 3.3.

5.1 Addition

Let $\mathtt{ct}_1$ be a ciphertext encrypting m_1, and $\mathtt{ct}_2$ a ciphertext encrypting m_2. Recall (Definition 2), that the messages m_1 and m_2 can always be encoded as polynomials $\widehat{m_1}$ and $\widehat{m_2}$ of degree at most $n-1$, where $\|\widehat{m_1}\|, \|\widehat{m_2}\| \leq (b+1)/2$. The output $\mathtt{ct}_{\text{add}} = \mathtt{Add}(\mathtt{ct}_1, \mathtt{ct}_2)$ of a homomorphic addition is supposed to encrypt the sum of the underlying plaintexts, $m_1 + m_2 \in \mathcal{M}$, as long as $\mathtt{ct}_{\text{add}}$ has noise less than $1/2$.

In the following proof, we will want to replace the sum of the encodings $\widehat{m_1}$ and $\widehat{m_2}$ with $\widehat{m_1 + m_2}$. Luckily, these are not too different: $(\widehat{m_1} + \widehat{m_2} - \widehat{m_1 + m_2})(b) = 0 \mod (b^n + 1)$, which means that in R (i.e. modulo $x^n + 1$) we can always write $\widehat{m_1} + \widehat{m_2} - \widehat{m_1 + m_2} = a(x - b)$, for some integer-coefficient polynomial a.

Lemma 9. *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 $\|v_{add}\| \leq \|v_1\| + \|v_2\|$.*

Proof. According to Definition 3, we can write

$$\frac{x-b}{q}\mathtt{ct}_1(s) = \widehat{m_1} + v_1 + a_1(x-b)\,, \quad \frac{x-b}{q}\mathtt{ct}_2(s) = \widehat{m_2} + v_2 + a_2(x-b)\,,$$

for some integer-coefficient polynomials a_1, a_2. It follow from the definition of $\mathtt{Add}$, that

$$\begin{aligned}\frac{x-b}{q}\mathtt{ct}_{\text{add}}(s) &= \frac{x-b}{q}\mathtt{ct}_1(s) + \frac{x-b}{q}\mathtt{ct}_2(s) \\ &= \widehat{m_1} + \widehat{m_2} + v_1 + v_2 + (a_1 + a_2)(x-b) \\ &= \widehat{m_1 + m_2} + v_1 + v_2 + (a_1 + a_2 + a)(x-b)\,.\end{aligned}$$

Therefore, $\mathtt{ct}_{\text{add}}$ indeed encrypts the sum $m_1 + m_2$, and has noise $v_{\text{add}} = v_1 + v_2$. Obviously $\|v_{\text{add}}\| = \|v_1 + v_2\| \leq \|v_1\| + \|v_2\|$. □

5.2 Multiplication

Recall that homomorphic multiplication ($\texttt{Multiply}$) consists of two steps: the first step ($\texttt{Multiply}'$ outputs an intermediate three-component ciphertext, and the second step ($\texttt{Relinearize}$) changes it back to size 2.

The output $\texttt{ct}_{\text{mult}} = \texttt{Multiply}(\texttt{ct}_1, \texttt{ct}_2, \texttt{evk})$ of a homomorphic multiplication is supposed to encrypt the product of the underlying plaintexts, $m_1 m_2 \in \mathcal{M}$, as long as $\texttt{ct}_{\text{mult}}$ has noise less than $1/2$.

Just like in Lemma 9, in the following proof, we will want to replace the product of the encodings $\widehat{m_1}$ and $\widehat{m_2}$ with $\widehat{m_1 m_2}$. Again, these are not too different: $(\widehat{m_1}\widehat{m_2} - \widehat{m_1 m_2})(b) = 0 \mod (b^n + 1)$, which means that in R (i.e. modulo $x^n + 1$) we can always write $\widehat{m_1}\widehat{m_2} - \widehat{m_1 m_2} = a(x - b)$, for some integer-coefficient polynomial a.

Lemma 10. *Let $\texttt{ct}_1$ and $\texttt{ct}_2$ 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 $\texttt{ct}_{mult} = \texttt{Multiply}(\texttt{ct}_1, \texttt{ct}_2, \texttt{evk})$ encrypts the product $m_1 m_2 \in \mathcal{M}$, and has noise v_{mult}, such that*

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

Proof. Denote $\texttt{ct}_1 = (x_0, x_1)$ and $\texttt{ct}_2 = (y_0, y_1)$. Consider the three-component ciphertext $\texttt{ct}'_{\text{mult}} = (c'_0, c'_1, c'_2)$ output by $\texttt{Multiply}'(\texttt{ct}_1, \texttt{ct}_2)$. By definition,

$$\begin{aligned} c'_0 &= \frac{x-b}{q} x_0 y_0 + \epsilon_0 + A_0 q\,, \\ c'_1 &= \frac{x-b}{q}(x_0 y_1 + x_1 y_0) + \epsilon_1 + A_1 q\,, \\ c'_2 &= \frac{x-b}{q} x_1 y_1 + \epsilon_2 + A_2 q\,, \end{aligned}$$

for some polynomials $\epsilon_0, \epsilon_1, \epsilon_2$ with coefficients in $(-\frac{1}{2}, \frac{1}{2}]$, and for some polynomials A_0, A_1, A_2 with integer coefficients.

First we prove that

$$\frac{x-b}{q}\texttt{ct}'_{\text{mult}}(s) = \frac{x-b}{q}(c'_0 + c'_1 s + c'_2 s^2) = \widehat{m_1 m_2} + v'_{\text{mult}} + a'(x-b)\,,$$

for some small polynomial v'_{mult}, and for some integer coefficient polynomial a'. According to Definition 3, we can write

$$\frac{x-b}{q}\texttt{ct}_1(s) = \widehat{m_1} + v_1 + a_1(x-b)\,, \quad \frac{x-b}{q}\texttt{ct}_2(s) = \widehat{m_2} + v_2 + a_2(x-b)\,,$$

for some integer-coefficient polynomials a_1, a_2. It follows from the definition of $\texttt{Multiply}'$, that

$$\begin{aligned}
\frac{x-b}{q}\texttt{ct}'_{\text{mult}}(s) &= \frac{x-b}{q}(c'_0 + c'_1 s + c'_2 s^2) \\
&= \frac{x-b}{q}\left[\left(\frac{x-b}{q}(x_0y_0) + \epsilon_0 + A_0 q\right)\right. \\
&\quad + \left(\frac{x-b}{q}(x_0y_1 + x_1y_0) + \epsilon_1 + A_1 q\right) s \\
&\quad \left. + \left(\frac{x-b}{q}(x_1y_1) + \epsilon_2 + A_2 q\right) s^2\right] \\
&= \frac{x-b}{q}\texttt{ct}_1(s) \cdot \frac{x-b}{q}\texttt{ct}_2(s) \\
&\quad + \frac{x-b}{q}(\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2) \\
&\quad + \left(A_0 + A_1 s + A_2 s^2\right)(x-b) \\
&= (\widehat{m_1} + v_1 + a_1(x-b))(\widehat{m_2} + v_2 + a_2(x-b)) \\
&\quad + \frac{x-b}{q}(\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2) \\
&\quad + (A_0 + A_1 s + A_2 s^2)(x-b) \\
&= \widehat{m_1 m_2} + \widehat{m_1} v_2 + \widehat{m_2} v_1 + v_1 v_2 \\
&\quad + v_1 a_2(x-b) + v_2 a_1(x-b) \\
&\quad + \frac{x-b}{q}(\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2) \mod (x-b)\,.
\end{aligned}$$

Thus,

$$\begin{aligned}
v'_{\text{mult}} &= \widehat{m_1} v_2 + \widehat{m_2} v_1 + v_1 v_2 \\
&\quad + v_1 a_2(x-b) + v_2 a_1(x-b) \\
&\quad + \frac{x-b}{q}(\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2)\,.
\end{aligned}$$

To establish a bound for v'_{mult}, we first note that

$$\left\| \frac{x-b}{q}(\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2) \right\| \leq \frac{(b+1)B}{q}(1 + n + n^2)\,. \tag{3}$$

Next, note that

$$\begin{aligned}
\|a_i(x-b)\| &= \left\| \frac{x-b}{q}\texttt{ct}_i(s) - \widehat{m_i} - v_i \right\| \\
&\leq \frac{b+1}{2}(1+n) + \frac{b+1}{2} + \|v_i\| \\
&= \frac{b+1}{2}(2+n) + \|v_i\|\,.
\end{aligned} \tag{4}$$

Now use (3) and (4) to bound v'_{mult}:

$$\begin{aligned}
\|v'_{\mathrm{mult}}\| &\le \|\widehat{m_1} v_2\| + \|\widehat{m_2} v_1\| + \|v_1 v_2\| \\
&\quad + \|v_1 a_2 (x-b)\| + \|v_2 a_1 (x-b)\| \\
&\quad + \left\| \frac{x-b}{q} (\epsilon_0 + \epsilon_1 s + \epsilon_2 s^2) \right\| \\
&\le \frac{b+1}{2} N_{m_1} \|v_2\| + \frac{b+1}{2} N_{m_2} \|v_1\| + n \|v_1\| \|v_2\| \\
&\quad + n\|v_1\| \left(\frac{b+1}{2}(2+n) + \|v_2\| \right) \\
&\quad + n\|v_2\| \left(\frac{b+1}{2}(2+n) + \|v_1\| \right) \\
&\quad + \frac{(b+1)B}{q}(1+n+n^2) \\
&= \frac{b+1}{2}(N_{m_1} + n^2 + 2n)\|v_2\| \\
&\quad + \frac{b+1}{2}(N_{m_2} + n^2 + 2n)\|v_1\| \\
&\quad + 3n\|v_1\|\|v_2\| + \frac{(b+1)B}{q}(1+n+n^2)\,.
\end{aligned}$$

It remains to analyze what happens when `Relinearize` is applied to $\mathtt{ct}'_{\mathrm{mult}}$. Recall that, given an evaluation key

$$\mathtt{evk} = [([-(a_i s + e_i) + w^i s^2]_q, a_i) : i = 0, \ldots, \ell]\,,$$

$\mathtt{Relinearize}(\mathtt{ct}'_{\mathrm{mult}}, \mathtt{evk})$ outputs $\mathtt{ct}_{\mathrm{mult}} = (c_0, c_1)$, where

$$c_0 = c'_0 + \sum_{i=0}^{\ell} \mathtt{evk}[i][0] {c'_2}^{(i)}, \quad c_1 = c'_1 + \sum_{i=0}^{\ell} \mathtt{evk}[i][1] {c'_2}^{(i)}\,,$$

and ${c'_2}^{(i)}$ denotes the i-th component in the base-w expansion of c'_2. Then, for some integer-coefficient polynomials a_i, $0 \le i \le \ell + 1$,

$$\begin{aligned}
\frac{x-b}{q} \mathtt{ct}_{\mathrm{mult}}(s) &= \frac{x-b}{q}(c_0 + c_1 s) \\
&= \frac{x-b}{q} \left(\sum_{i=0}^{\ell} \mathtt{evk}[i][0] {c'_2}^{(i)} + s \sum_{i=0}^{\ell} \mathtt{evk}[i][1] {c'_2}^{(i)} \right) \\
&\quad + \frac{x-b}{q}(c'_0 + c'_1 s) \\
&= \frac{x-b}{q} \left(-\sum_{i=0}^{\ell} e_i {c'_2}^{(i)} + \sum_{i=0}^{\ell} a_i q {c'_2}^{(i)} + s^2 \sum_{i=0}^{\ell} w^i {c'_2}^{(i)} \right) \\
&\quad + \frac{x-b}{q}(c'_0 + c'_1 s)\,.
\end{aligned}$$

Since $\sum_i w^i {c_2'}^{(i)} = c_2'$, this becomes

$$\begin{aligned}
&\frac{x-b}{q}\left(-\sum_{i=0}^{\ell} e_i {c_2'}^{(i)} + \sum_{i=0}^{\ell} a_i q {c_2'}^{(i)}\right) + \frac{x-b}{q}\left(c_0' + c_1' s + c_2'\right) \\
&= -\frac{x-b}{q}\sum_{i=0}^{\ell} e_i {c_2'}^{(i)} + \frac{x-b}{q}\left(c_0' + c_1' s + c_2' s^2\right) + (x-b)\sum_{i=0}^{\ell} a_i {c_2'}^{(i)} \\
&= \widehat{m_1 m_2} + v'_{\text{mult}} - \frac{x-b}{q}\sum_{i=0}^{\ell} e_i {c_2'}^{(i)} + \left(a_{\ell+1} + \sum_{i=0}^{\ell} a_i {c_2'}^{(i)}\right)(x-b)\,.
\end{aligned}$$

Hence, the noise further grows by an additive factor in the relinearization process. Bounding the new term is easy:

$$\left\| -\frac{x-b}{q}\sum_{i=0}^{\ell} e_i {c_2'}^{(i)} \right\| \leq \frac{b+1}{q}\sum_{i=0}^{\ell} \left\| e_i {c_2'}^{(i)} \right\| \leq \frac{b+1}{q} nB(\ell+1)w\,.$$

Putting everything together, we finally find

$$\begin{aligned}
\|v_{\text{mult}}\| &\leq \|v'_{\text{mult}}\| + \left\| -\frac{x-b}{q}\sum_{i=0}^{\ell} e_i {c_2'}^{(i)} \right\| \\
&\leq \frac{b+1}{2}(N_{m_1} + n^2 + 2n)\|v_2\| \\
&\quad + \frac{b+1}{2}(N_{m_2} + n^2 + 2n)\|v_1\| \\
&\quad + 3n\|v_1\|\|v_2\| + \frac{(b+1)B}{q}(1+n+n^2) \\
&\quad + \frac{b+1}{q} nB(\ell+1)w\,.
\end{aligned}$$

□

5.3 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 3.3, and as was motivated in Section 3.2. Again, we use the canonical norm $\|\cdot\|^{\text{can}}$ instead of the usual infinity norm $\|\cdot\|$ for the same reasons as in Section 3.3: 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* `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* `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:

$$\|v_{\text{mult}}\|^{\text{can}} \lesssim 14(b+1)n \max\left\{\|v_1\|^{\text{can}}, \|v_2\|^{\text{can}}\right\}. \tag{5}$$

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

$$\|v_{\text{initial}}\|^{\text{can}} \lesssim \frac{(b+1)^2 n^{3/2}}{q}. \tag{6}$$

6 Fractional Encoder

The *fractional encoder* introduced by Dowlin et al. in [27] (see also [19, 25]) 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

$$\texttt{Encode} : \mathcal{P} \to \mathcal{M}, \quad \texttt{Encode}\left(\frac{x}{y}\right) = xy^{-1} \mod (b^n+1)\,. \tag{7}$$

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 $\texttt{Encode}$ presented above well-defined and injective, and thus invertible in its range.

Lemma 14. *The map $\texttt{Encode} : \mathcal{P} \to \mathcal{M}$ in (7) 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

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

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 $z \in \mathtt{Encode}(\mathcal{P})$, we have*

$$\mathtt{Decode}(z) = \frac{[zb^{n/2}]_{b^n+1}}{b^{n/2}} .$$

Proof. Assume $z = \mathtt{Encode}(y)$, 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 (see Remark 1 below), 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 $\mathtt{Encode} : \mathcal{P} \to \mathcal{M}$ in (7) 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. □

Remark 1. 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 $z \in \mathtt{Encode}(\mathcal{P})$, we have*

$$\mathtt{Decode}(z) = \frac{[zb^{n/2-1}]_{b^n+1}}{b^{n/2-1}}.$$

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 [25]. 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 [22], which performs better than the integer encoders of [19]. 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 [19]) 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 [19], which are estimated [3] to have a high security level even considering the new attack of Albrecht [1]. For convenience, we present these pairs in Table 1. 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 [43, 19].

n	2048	4096	8192	16384
q	$2^{60} - 2^{14} + 1$	$2^{116} - 2^{18} + 1$	$2^{226} - 2^{26} + 1$	$2^{435} - 2^{33} + 1$

Table 1. Parameters (n, q).

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 3.3 for FV, and in Section 5.3 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

$$D \lesssim \left\lfloor \frac{\log q - \log(84\sigma tn)}{\log(14tn) + A} \right\rfloor . \tag{8}$$

We use the analysis of [22] (see also [25]) 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, [22] proves that we need

$$\sqrt{\frac{6}{\pi 2^D d(d+2)}} \, (d+1)^{2^D} 2^{A(2^{D+1}-2)} < t/2 \,. \tag{9}$$

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 (9), and the noise condition (8). The results are presented in Table 2.

		$A = 0$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$
2048	60	4	1	5	1	6	1	7	1	8	1
4096	116	9	2	11	2	13	2	16	2	19	2
8192	226	19	3	24	3	30	3	36	3	19	2
16384	435	39	4	50	4	63	4	36	3	43	3
32768	890	80	5	102	5	63	4	76	4	91	4

		$A = 3$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$
2048	60	10	1	11	1	12	1	13	1	—	0
4096	116	10	1	11	1	12	1	13	1	14	1
8192	226	27	2	29	2	31	2	34	2	37	2
16384	435	61	3	66	3	72	3	78	3	85	3
32768	890	129	4	140	4	153	4	78	3	85	3

		$A = 10$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$	$\log t$	$\max D$
2048	60	—	0	—	0	—	0	—	0	—	0
4096	116	24	1	25	1	26	1	27	1	28	1
8192	226	24	1	25	1	26	1	27	1	28	1
16384	435	69	2	71	2	73	2	76	2	79	2
32768	890	159	3	164	3	170	3	176	3	183	3

Table 2. Allowed maximum depth D for the FV scheme with NAF encoding; at each level the circuit has 2^A additions followed by a multiplication. Results are given for $A \in \{0, 3, 10\}$, and several input sizes $L \in \{2^8, 2^{16}, 2^{32}, 2^{64}, 2^{128}\}$.

New scheme. For the new scheme, using (6), (5), 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

$$D \lesssim \left\lfloor \frac{\log q - \log\left(2(b+1)^2 n^{3/2}\right)}{\log(14(b+1)n) + A} \right\rfloor . \tag{10}$$

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 [25]) $V = L^{2^D} 2^{A(2^{D+1}-2)}$, so for correctness it is necessary that $V \leq (b^n - 1)/2$, which yields

$$D \lesssim \left\lfloor \log\left(\frac{\log\left((b^n-1)2^{2A-1}\right)}{\log\left(2^{2A}L\right)} \right) \right\rfloor \approx \left\lfloor \log\left(\frac{n\log b + 2A - 1}{2A + \log L} \right) \right\rfloor . \tag{11}$$

Combining (11) with the noise condition (10) yields, for a fixed b, the overall bound

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

The results for maximizing D are presented in Table 3. The largest parameters illustrate how the size of the integers quickly becomes the main bottleneck in the new scheme, and demands the use of extremely large values for b.

7.2 Results

Comparing Table 2 and Table 3 shows 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. For convenience, we summarize the performance results in Figure 1 in a more intuitive way.

We would also like to point out that the parameters we used in our comparison are estimated to have a very high security level against the most recent attacks [3, 1]. 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

		$A = 0$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$
2048	60	2	2	2	2	2	2	2	2	2	2
4096	116	2	5	2	5	2	5	2	5	3	5
8192	226	3	10	5	10	5	9	17	9	17	8
16384	435	257	14	257	13	257	12	257	11	65539	11
32768	890	$\approx 2^{16}$	16	$\approx 2^{16}$	15	$\approx 2^{32}$	15	$\approx 2^{32}$	14	$\approx 2^{32}$	13

		$A = 3$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$
2048	60	2	2	2	2	2	2	2	2	2	2
4096	116	2	5	2	5	2	5	2	5	3	5
8192	226	4	10	7	10	6	9	21	9	19	8
16384	435	128	13	2048	13	724	12	431	11	332	10
32768	890	$\approx 2^{28}$	16	$\approx 2^{22}$	15	$\approx 2^{19}$	14	$\approx 2^{35}$	14	$\approx 2^{33.5}$	13

		$A = 10$									
		$L = 2^8$		$L = 2^{16}$		$L = 2^{32}$		$L = 2^{64}$		$L = 2^{128}$	
n	$\log q$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$	b	$\max D$
2048	60	2	2	2	2	2	2	2	2	2	2
4096	116	2	5	2	5	2	5	2	5	3	5
8192	226	4	9	5	9	10	9	7	8	25	8
16384	435	128	12	512	12	91	11	1447	11	609	10
32768	890	$\approx 2^{28}$	15	$\approx 2^{18}$	14	$\approx 2^{26}$	14	$\approx 2^{21}$	13	$\approx 2^{37}$	13

Table 3. Allowed maximum depth D for 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 several input sizes $L \in \{2^8, 2^{16}, 2^{32}, 2^{64}, 2^{128}\}$.

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 [25]: 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 [19]. Instead, the benefit of using fractional encoders is mostly for convenience, as it frees the

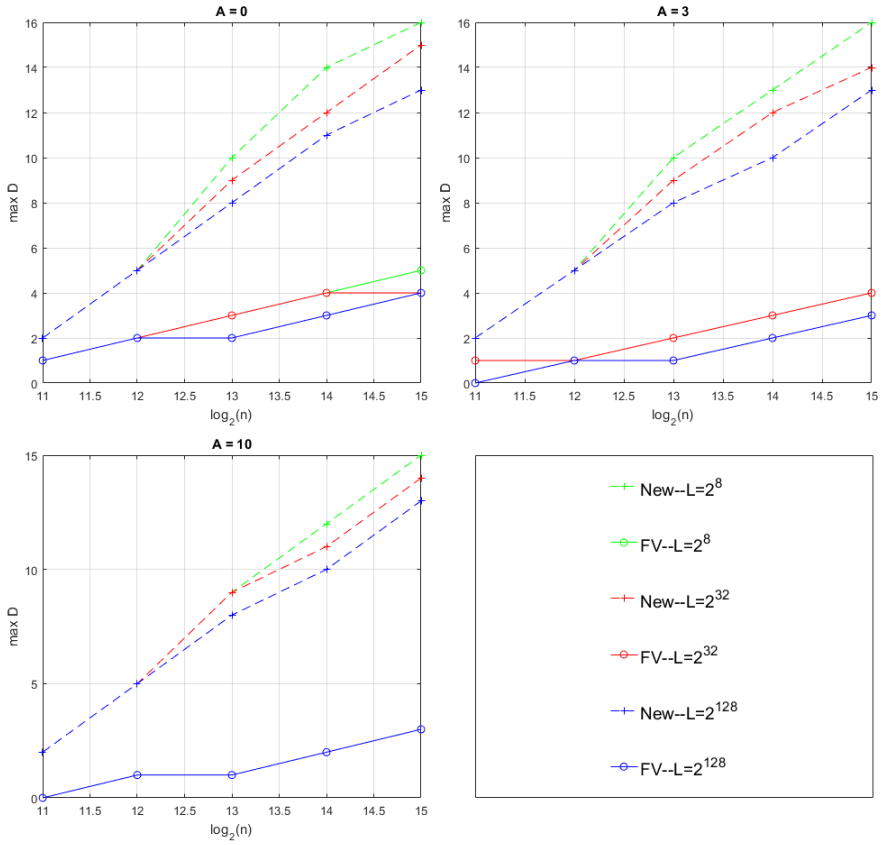

Fig. 1. Comparing maximum depth D between the FV scheme with NAF encoding, and the new scheme; at each level the circuit has 2^A additions followed by a multiplication. Results are given for $A \in \{0, 3, 10\}$, and input sizes $L \in \{2^8, 2^{32}, 2^{128}\}$.

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 (Table 2 and Table 3) has the same performance as computations on rational numbers with e.g. 32-bit fractional and 32-bit integral parts.

7.4 Computational Overhead

The new scheme is algebraically slightly more complicated than the FV scheme, and one can reasonably ask what kind of an impact these changes have on its performance in implementations.

First, we note that the hat encoding of Section 4.1 is computationally the same as the usual integer encoders, and same is true for the fractional encoder

of Section 6. In general, the cost of encoding is typically negligible compared to other operations.

Encryption incurs a slight overhead due to multiplication by the constant Δ in FV encryption being replaced by the polynomial Δ_b. However, Δ_b is typically very sparse, containing fewer than $\log q / \log b$ non-zero terms, so the product $\Delta_b \widehat{m}$ can be very efficiently computed. This is negligible compared to the cost of the dense polynomial multiplications that dominate the cost of encryption both in FV and in the new scheme.

The cost of homomorphic multiplications is nearly unchanged. Instead of multiplying ciphertext polynomials by an integer t in the FV scheme, we multiply then by a polynomial $x - b$. This is almost equally fast, and negligible compared to the cost of the dense polynomial multiplications that dominate the cost of multiplication both in FV and in the new scheme.

Decryption is most affected by the changes, as one has to evaluate the polynomial $\widehat{M}$ (see Section 4) at $x = b$, and in the process reduce it modulo $b^n + 1$. In particular, $\widehat{M}$ can be dense even if $\widehat{M}(b)$ evaluates to something very small modulo $b^n + 1$. This is normally not the case in the FV scheme with the NAF encoder: the decrypted plaintext polynomial tends to be proportional in size to the number it decodes to. While this is not ideal, we would like to point out that the new scheme allows smaller parameters to be used, which will instantly make up for the cost of evaluating $\widehat{M}(b)$, and that decryption is very rarely a bottleneck in applications of homomorphic encryption anyway.

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 [11], genomic analysis [42, 39, 22], evaluating neural networks on encrypted images [34], and performing predictive analysis on power consumption in smart grids [10, 9]. 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. [34, 10]). However, with the new scheme, the situation is much better.

First, we would like to comment on [42], and [22]. These works 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 [42] and minimax [22] approximations. For example, for evaluating logistic regression models, [22] 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[5] security level $\lambda \approx 110$ against the attack of [1]. With the new scheme such computations can be done easily with only $n = 4096$, and an estimated security level of $\lambda \approx 120$.

In [34] the authors discuss evaluating neural networks on encrypted images. To achieve good performance, they use a network specifically designed to be easy to evaluate on homomorphically encrypted data. Namely, they use square activation functions, instead of the more common sigmoid or rectified linear functions, which are hard to approximate well with low-degree polynomials. The authors choose to use $(n = 8192, q \approx 2^{383})$, and $\sigma = 3.19$, resulting in a low estimated security level of $\lambda \approx 72$ against the attack of [1]. The computation consists of three linear layers, and two non-linear (square) layers, which can easily be performed using the new scheme with $n = 4096$ size parameters, and an estimated security level of $\lambda \approx 120$. This also will make a big difference in terms of computational performance, and in terms of message expansion. To be fair, one of the key aspects of [34] is to use batching to improve the amortized performance by allowing thousands of small image predictions to be done simultaneously, which might also be the preferred scenario in applications to medical image prediction. Nevertheless, in cases where only one instance needs to be evaluated, our approach will be much more efficient. The authors also point out that the square function might not be suitable for deeper networks due to the instability it creates in the training process. This problem could be solved to some extent using the new scheme, as the enhanced homomorphic capabilities would allow the use of higher degree activation functions, with better growth properties near zero.

Remark 2. It is possible to use batching also with the new scheme, although typically to a lesser extent than in schemes like FV. The idea is simply to choose b such that $b^n + 1$ factors into a product of relatively prime factors $\prod e_i$. Then, by the Chinese Remainder Theorem, there is a canonical ring isomorphism

$$\mathbb{Z}/(b^n + 1)\mathbb{Z} \cong \prod \mathbb{Z}/e_i\mathbb{Z}\,,$$

allowing us to operate on integers in each factor separately. Of course, now the integers in each factor separately must not wrap around the corresponding modulus e_i, which limits the usefulness of this technique. On the other hand, with a small number of factors (such as 2–8) this might not be an issue.

Finally, we would like to comment on the work of [9], which is a follow-up paper to [10]. In [10], the authors discuss an application of homomorphic encryption to consumption prediction in *smart grids* using a technique called *Group Method of Data Handling (GMDH)*, and in [9] improve the results using a new *Non-Integral Base Non-Adjacent Form* encoder for rational numbers: essentially, this technique allows the use of a smaller plaintext coefficient modulus t, which results in an overall performance improvement compared to simpler rational

[5] In this section, all estimates of the security level λ were obtained using the LWE estimator [3] of Albrecht *et al.* (commit `ee94f7e`).

number encoding methods. The GMDH "network" that they use has three hidden layers, resulting in a circuit of depth 4. They choose to use parameters $n = 4096$, $q \approx 2^{186}$, $\sigma = 102$, which are estimated to have $\lambda \approx 75$ bits of security against the attack of [1]. With the new scheme, we can comfortably evaluate such circuits with $n = 4096$ and $q \approx 2^{116}$. This parameter set implies an estimated security level of $\lambda \approx 120$, which would likely be required in realistic implementations of the protocol. Due to the smaller q we have, our ciphertexts are also smaller by nearly 40%, which will result in an overall run-time improvement.

Acknowledgements

We thank the anonymous reviewers for helpful comments on a previous version of this work.

References

[1] Martin R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HElib and SEAL. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part II*, volume 10211 of *Lecture Notes in Computer Science*, pages 103–129, 2017.

[2] Martin R. Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, volume 9814 of *Lecture Notes in Computer Science*, pages 153–178. Springer, 2016.

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

[4] Diego F. Aranha and Alfred Menezes, editors. *Progress in Cryptology - LATINCRYPT 2014 - Third International Conference on Cryptology and Information Security in Latin America, Florianópolis, Brazil, September 17-19, 2014, Revised Selected Papers*, volume 8895 of *Lecture Notes in Computer Science*. Springer, 2015.

[5] Seiko Arita and Shota Nakasato. Fully homomorphic encryption for point numbers. Cryptology ePrint Archive, Report 2016/402, 2016. `http://eprint.iacr.org/2016/402`.

[6] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192, 2015. `http://eprint.iacr.org/2015/1192`.

[7] Jean-Claude Bajard, Julien Eynard, Anwar Hasan, and Vincent Zucca. A full rns variant of fv like somewhat homomorphic encryption schemes. In *Selected Areas in Cryptography-SAC*, 2016.

[8] Fabrice Benhamouda, Tancrède Lepoint, Claire Mathieu, and Hang Zhou. Optimization of bootstrapping in circuits. In Philip N. Klein, editor, *Proceedings*

of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19, pages 2423–2433. SIAM, 2017.

[9] Charlotte Bonte, Carl Bootland, Joppe W. Bos, Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Faster homomorphic function evaluation using non-integral base encoding. Cryptology ePrint Archive, Report 2017/333, 2017. `http://eprint.iacr.org/2017/333`.

[10] Joppe W. Bos, Wouter Castryck, Ilia Iliashenko, and Frederik Vercauteren. Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In Marc Joye and Abderrahmane Nitaj, editors, *Progress in Cryptology - AFRICACRYPT 2017 - 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings*, volume 10239 of *Lecture Notes in Computer Science*, pages 184–201, 2017.

[11] Joppe W Bos, Kristin Lauter, and Michael Naehrig. Private predictive analysis on encrypted medical data. *Journal of biomedical informatics*, 50:234–243, 2014.

[12] Joppe W. Bos, Kristin E. Lauter, Jake Loftus, and Michael Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In Martijn Stam, editor, *Cryptography and Coding - 14th IMA International Conference, IMACC 2013, Oxford, UK, December 17-19, 2013. Proceedings*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

[13] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Safavi-Naini and Canetti [52], pages 868–886.

[14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 309–325. ACM, 2012.

[15] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 575–584. ACM, 2013.

[16] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.

[17] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from Ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.

[18] Michael Brenner and Kurt Rohloff, editors. *Proceedings of WAHC'17 - 5th Workshop on Encrypted Computing and Applied Homomorphic Cryptography*, April 2017.

[19] Hao Chen, Kim Laine, and Rachel Player. Simple Encrypted Arithmetic Library - SEAL. In Brenner and Rohloff [18].

[20] Jung Hee Cheon, Kyoohyung Han, and Duhyeong Kim. Faster bootstrapping of FHE over the integers. Cryptology ePrint Archive, Report 2017/079, 2017. `http://eprint.iacr.org/2017/079`.

[21] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics*, 19(A):255?266, 2016.

[22] Jung Hee Cheon, Jinhyuck Jeong, Joohee Lee, and Keewoo Lee. Privacy-preserving computations of predictive medical models with minimax approximation and Non-Adjacent Form. In Brenner and Rohloff [18].

[23] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. Cryptology ePrint Archive, Report 2016/421, 2016. `http://eprint.iacr.org/2016/421`.

[24] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[25] A. Costache, N.P. Smart, S. Vivek, and A. Waller. Fixed point arithmetic in SHE scheme. Cryptology ePrint Archive, Report 2016/250, 2016. `http://eprint.iacr.org/2016/250`.

[26] Ana Costache and Nigel P. Smart. Which ring based somewhat homomorphic encryption scheme is best? In Sako [53], pages 325–340.

[27] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Manual for using homomorphic encryption for bioinformatics. *Proceedings of the IEEE*, 105(3):552–567, 2017.

[28] Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *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*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, 2015.

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

[30] Matthias Geihs and Daniel Cabarcas. Efficient integer encoding for homomorphic encryption via ring isomorphisms. In Aranha and Menezes [4], pages 48–63.

[31] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009.

[32] Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic evaluation of the AES circuit. In Safavi-Naini and Canetti [52], pages 850–867.

[33] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013.

[34] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin E. Lauter, Michael Naehrig, and John Wernsing. CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria-Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 201–210. JMLR.org, 2016.

[35] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

[36] Jeffrey Hoffstein and Joseph Silverman. Optimizations for NTRU. Public-Key Cryptography and Computational Number Theory (Proceedings of the International Conference organized by the Stefan Banach International Mathematical Center Warsaw, Poland, September 11-15, 2000), 2001. Available at: `https://assets.securityinnovation.com/static/downloads/NTRU/resources/TECH_ARTICLE_OPT.pdf`.

[37] Angela Jäschke and Frederik Armknecht. Accelerating homomorphic computations on rational numbers. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings*, volume 9696 of *Lecture Notes in Computer Science*, pages 405–423. Springer, 2016.

[38] Alhassan Khedr, Glenn Gulak, and Vinod Vaikuntanathan. SHIELD: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Transactions on Computers*, 65(9):2848–2858, 2016.

[39] Miran Kim and Kristin Lauter. Private genome analysis through homomorphic encryption. Cryptology ePrint Archive, Report 2015/965, 2015. `http://eprint.iacr.org/2015/965`.

[40] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 3–26, 2017.

[41] Kim Laine, Hao Chen, and Rachel Player. Simple Encrypted Arithmetic Library - SEAL v2.2. Technical report, June 2017.

[42] Kristin E. Lauter, Adriana López-Alt, and Michael Naehrig. Private computation on encrypted genomic data. In Aranha and Menezes [4], pages 3–27.

[43] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[44] Adriana López-Alt and Michael Naehrig. Large integer plaintexts in ring-based fully homomorphic encryption, 2014. Unpublished.

[45] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1219–1234. ACM, 2012.

[46] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):43, 2013.

[47] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-lwe cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the*

Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2013.

[48] Carlos Aguilar Melchor, Joris Barrier, Serge Guelton, Adrien Guinet, Marc-Olivier Killijian, and Tancrède Lepoint. NFLlib: NTT-Based Fast Lattice Library. In Sako [53], pages 341–356.

[49] Michael Naehrig, Kristin E. Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In Christian Cachin and Thomas Ristenpart, editors, *Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011, Chicago, IL, USA, October 21, 2011*, pages 113–124. ACM, 2011.

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

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

[52] Reihaneh Safavi-Naini and Ran Canetti, editors. *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*. Springer, 2012.

[53] Kazue Sako, editor. *Topics in Cryptology - CT-RSA 2016 - The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29 - March 4, 2016, Proceedings*, volume 9610 of *Lecture Notes in Computer Science*. Springer, 2016.

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