More Efficient Functional Bootstrapping for General Functions in Polynomial Modulus

Han Xia[1,2], Feng-Hao Liu[3], and Han Wang[1,2(✉)]

[1] Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{xiahan,wanghan}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[3] Washington State University, Pullman, WA, USA
feng-hao.liu@wsu.edu

Abstract. Functional bootstrapping seamlessly integrates the benefits of homomorphic computation using a look-up table and the noise reduction capabilities of bootstrapping. Its wide-ranging applications in privacy-preserving protocols underscore its broad impacts and significance. In this work, our objective is to craft more efficient and less restricted functional bootstrapping methods for general functions within a polynomial modulus. We introduce a series of novel techniques, proving that functional bootstrapping for general functions can be essentially as efficient as regular FHEW/TFHE bootstrapping. Our new algorithms operate within the realm of prime-power and odd composite cyclotomic rings, offering versatility without any additional requirements on input noise and message space beyond correct decryption.

1 Introduction

Fully homomorphic encryption (FHE) has been identified as a powerful cryptographic tool, allowing arbitrary computation over ciphertexts without first decrypting it. Gentry pioneered FHE in his seminal work [33], sparking numerous subsequent studies such as [5,12,13,15,16,21,22,30,35,44,45]. In addition to theoretical progress, practical strides have been made with the development of several useful FHE libraries along this research trajectory [3,20,24,56,57,73], contributing significantly to potential real-world applications.

Bootstrapping with Polynomial Error Growth. As a pivotal breakthrough introduced by Gentry in [33], bootstrapping plays a crucial role in achieving "fully" homomorphic encryption. In a nutshell, the bootstrapping paradigm takes as input an FHE ciphertext $\boldsymbol{c} \in \mathsf{Enc}(m)$ and some bootstrapping key, and outputs another FHE ciphertext $\boldsymbol{c}' \in \mathsf{Enc}(\mathsf{Dec}(\boldsymbol{c})) \subset \mathsf{Enc}(m)$, with significantly reduced noise. As homomorphic computations in current FHE schemes inevitably incur noise, reaching a point where decryption becomes incorrect, bootstrapping

E. Boyle and M. Mahmoody (Eds.): TCC 2024, LNCS 15367, pp. 130–163, 2025.
https://doi.org/10.1007/978-3-031-78023-3_5

becomes the critical key that enables an arbitrary number of homomorphic operations and thus "F"-HE.

Among various FHE schemes, the work [16] showed for the first time that bootstrapping would only incur a polynomial error growth, though their method requires very large polynomial runtimes due to the reliance on Barrington's Theorem [6]. Thereafter, the work (referred to as AP14) [5] showed how to bootstrap with error growth and runtime being both small polynomials by treating decryption as an arithmetic function. The AP14 method critically relies on the GSW schemes [35] (known as the third generation of FHE schemes), yet their explicit method was in the plain lattice (i.e., LWE [63]) setting, which is not expected to be concretely efficient. Subsequently, FHEW [30] and TFHE [23] refined and optimized the AP14 method in the ring setting, introducing substantial new insights. These optimizations resulted in bootstrapping achievements within subseconds in practical implementations. Their impact is evident in the inclusion of these methods in various libraries, such as OpenFHE [3] and TFHE [24,73], highlighting their tangible real-world relevance to the community.

This work focuses on the setting of (functional) bootstrapping along the line of FHEW/TFHE, i.e., methods with polynomial error growth, which implies smaller FHE parameters and thus smaller FHE keys. We notice that the FHEW/TFHE computation is suited for computation expressed by boolean circuits, e.g., comparisons and decision diagram computations [11,28], with smaller memory requirements.

Functional Bootstrapping. Following the FHEW-like framework [53] (including FHEW [30] and TFHE [23]), the works [8,10] identified that in the power-of-2's cyclotomic rings, the bootstrapping method can be slightly modified with almost no additional cost, outputting $\boldsymbol{c}' \in \mathsf{Enc}(f(m))$ for any negacyclic $f : \mathbb{Z}_p \to \mathbb{Z}_p$ where p is the plaintext domain[1]. This is called functional bootstrapping [10,27,36,42,47,48] (or programmable bootstrapping [7,25,26]), which integrates noise reduction and (small) look-up table computation at comparable efficiency as the bootstrapping. A trivial way to overcome the negacyclicity is to use only half-domain of the plaintext space [10,11,36]. However, this significantly limits the construction of applications. To support both full-domian and general functions, several recent works [7,26,42,47,72] have aimed for efficient designs, which can be broadly categorized into the following two types.

- **Single Input/Output:** In these schemes [26,42,47,72], both the input and output are single LWE ciphertexts. They typically require more than two calls to the regular FHEW/TFHE bootstrapping, and some even impose additional constraints on the noise level of the input LWE ciphertext [47,72].
- **Multiple Inputs/Outputs:** This framework was first introduced in [36] and later optimized in [7,26]. It decomposes the plaintext into multiple bits or digits[2] and encrypts each segment separately. As a result, the number of

[1] For commonly used FHEW/TFHE parameters [53], the plaintext domain is roughly 3-bit to 5-bit. Also notice that f might be more general that works on $\mathbb{Z}_p \to \mathbb{Z}_{p'}$, i.e., the output ciphertext might be associated with a different plaintext domain.

[2] Here, "digits" means decomposing by another integer base B, i.e., $B > 2$.

ciphertexts and the invocations of regular bootstrapping in these schemes increase with the precision of the input plaintext. Moreover, to support digits rather than just bits, there must be constraints on the input ciphertexts [7].

As functional bootstrapping has many applications, such as privacy-preserving machine learning [42,46,49], and it can serve as an important building block of conversion between different types of FHE schemes with high precisions [47,49], it becomes an important open problem to optimize the functionality and efficiency of functional bootstrapping designs, either theoretically or practically. This motivates the main question of this work.

(**Main Question**). Can we *simultaneously* eliminate *all* constraints of functional bootstrapping (within a polynomial modulus) for general functions and make it as efficient as the regular FHEW/TFHE bootstrapping?

1.1 Our Contributions

To tackle the main question, this work designs a series of new full-domain functional bootstrapping algorithms for general functions that are essentially as efficient as the regular FHEW/TFHE bootstrapping. We summarize as follows.

Functionality. We design three new functional bootstrapping algorithms over general cyclotomic rings with prime, prime-power, and odd composite indices[3] for single LWE input[4]. All of them satisfy all the following desirable properties:

- Arbitrary plaintext modulus/encoding for input ciphertext.
- No additional input noise requirement beyond correct decryption.
- General functions $f : \mathbb{Z}_p \to \mathbb{Z}_{p'}$ with arbitrary positive integers p and p'.

Briefly, all of them have no restrictions on the input LWE ciphertext (with a fixed modulus), which implies the following result:

Functional bootstrapping for general functions works for inputs that are any valid LWE *ciphertext (i.e., correctly decryptable).*

Efficiency. Let n be the dimension of the input LWE ciphertext. All of our functional bootstrapping algorithms come at the cost of $n + O(\log n)$ homomorphic multiplications (i.e., FHE external products). Compared to the regular FHEW/TFHE bootstrapping that requires n homomorphic multiplications (in the dominating "Blind Rotation" procedure), this implies that the ratio of efficiency achieves $1 + o(1)$. In Table 1, we present a summary of our results and a comparison with prior full-domain designs.

[3] In summary, we support cyclotomic rings for two general categories – (1) odd and (2) power-of-2 indices.

[4] Our algorithms directly follow the technical line of single input/output. However, they can also be used to remove the constraints on input noise and encoding in schemes with multiple inputs/outputs. In other words, our optimizations lie at the core part of functional bootstrapping algorithms and can be applied to enhance the functionality and efficiency of all existing functional bootstrapping designs.

Table 1. Prior and our full-domain functional bootstrapping schemes for general functions. The modulus of the input LWE ciphertext is q. The minimal cyclotomic index stands for the smallest ring required for bootstrapping the input LWE ciphertext, which directly determines the efficiency of basic ring operations. Note that Blind Rotation is the core part that dominates the efficiency of FHEW/TFHE bootstrapping.

	# of Blind Rotations	Minimal Cyclotomic Index	Type of Cyclotomic Index	Without Restrictions on Input Error
[26]	2	$2q$	Power-of-2	Yes
	3	q	Power-of-2	Yes
[42]	$1 + d_g$ †	q	Power-of-2	Yes
[47]	2	$2q$	Power-of-2	No*
	3	$4q$	Power-of-2	Yes
[7]	$O(\beta d_g)$ ♯	$O(q)$	Power-of-2	Yes/No**
Ours	$1 + o(1)$	q	Prime-Power‡	Yes
	$1 + o(1)$	q	Composite§	Yes

† d_g is the gadget decomposition dimension satisfying $d_g > 1$.
♯ β is the precision (bit-length) of the input plaintext modulus. The value $O(\beta d_g)$ could be even larger for relatively large β (e.g., $\beta > 28$ as reported in [7]).
‡ We support arbitrary prime-power index (including pure prime and power-of-2).
§ We only support odd composite index (see Challenge 2 in Sect. 6).
* It requires the input noise to be less than roughly half of the maximal allowable bound (i.e., the upper bound for correct decryption). This method was also independently discovered in [72].
** "Yes" is only for the case where each input ciphertext encrypts only a single bit. Encrypting digits in a single ciphertext would require plaintext-ciphertext multiplication and homomorphic subtraction of the input ciphertext, which leads to the constraint on the input noise level.

1.2 Technical Overview

In this section, we highlight some critical insights in our designs. First, we briefly review the FHEW/TFHE (functional) bootstrapping framework.

FHEW/TFHE Framework. As discussed before, this framework takes as input an LWE ciphertext $\boldsymbol{c} \in \mathsf{Enc}(m)$ and some bootstrapping key, and aims to output another LWE ciphertext $\boldsymbol{c}' \in \mathsf{Enc}(m)$ or $\boldsymbol{c}' \in \mathsf{Enc}(f(m))$ with reduced noise. We further denote $\boldsymbol{c} = (b, \boldsymbol{a}) \in \mathbb{Z}_q \times \mathbb{Z}_q^n$ and notice that the decryption procedure is $\mathsf{Round}\left((b - \langle \boldsymbol{a}, \boldsymbol{s} \rangle \mod q)\right)$, where $\boldsymbol{s}$ is the secret key and $\mathsf{Round}(\cdot)$ is some rounding function for decoding. To achieve (functional) bootstrapping, the framework first uses the Blind Rotation technique that produces a RLWE ciphertext that encrypts $\zeta_q^{b-\langle \boldsymbol{a}, \boldsymbol{s} \rangle} := \zeta_q^{\alpha}$ where ζ_q is a primitive q-th root of unity, and then extracts an $\mathsf{LWE.Enc}(\mathsf{Round}(\alpha))$ of more general $\mathsf{LWE.Enc}(f(\alpha))$ given $\mathsf{RLWE.Enc}(\zeta_q^{\alpha})$. As α is computed in the exponent of ζ_q's power, it naturally takes the modulo q.

For the complexity, the Blind Rotation procedure takes n homomorphic multiplications (specifically the external products), dominating the overall complexity of the bootstrapping procedure. In power-of-two cyclotomic rings, the extraction procedure can be efficiently achieved (almost free) for the cases of $\mathsf{Round}(\cdot)$ function and negacyclic functions. For general functions however, current methods use different design concepts that require more than two calls to the core bootstrapping (Blind Rotation), and some of them even require ring extension (see Table 1 for the minimal cyclotomic index).

Our Goal. To achieve a more efficient method, we aim to design a more powerful and efficient function evaluation procedure than extraction, particularly at the cost of $o(n)$ homomorphic multiplications (external products). Combining with one call to the Blind Rotation, this would imply the overall cost to be $n + o(n)$ homomorphic multiplications, meaning $1+o(1)$ times the regular FHEW/TFHE bootstrapping. Below, we present our new insights on how we achieve our goal.

Our Blueprint. We aim to resolve the problem of the most general form where the input noise is only required to be bounded by the maximal allowable value of correct decryption, and the function has no restriction. Particularly, we first observe that any discrete function can be expressed by the linear combination of the equality test function: Define the equality test as $\mathsf{EqT}(\zeta_q^\alpha, \beta)$ that takes some power of ζ_q and $\beta \in \mathbb{Z}_q$, and outputs 1 if $\alpha = \beta \bmod q$ or otherwise outputs 0. Then any function $f : \mathbb{Z}_q \to \mathbb{Z}_h$ (for any positive integer h) can be expressed as $f(\alpha) = \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \mathsf{EqT}(\zeta_q^\alpha, \beta)$. Based on this idea, if there exists such a homomorphic equality test, we can construct a function evaluation algorithm that takes as input $\mathsf{RLWE.Enc}(\zeta_q^\alpha)$ and outputs $\mathsf{RLWE.Enc}(f(\alpha))$. So, our remaining task is to find an equality test function that can be efficiently computed homomorphically. In this way, the desired function can be evaluated. Our idea exploits the technique of homomorphic equality test and the algebraic trace over three different types of cyclotomic rings, as we elaborate below.

The Case of Prime Cyclotomic Rings. In this setting, the algebraic trace inherently possesses properties close to what we require. Particularly, when q is prime, the algebraic trace function has exactly two branches as follows:

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^{\alpha-\beta}) = \begin{cases} q-1 & \text{if} \quad \alpha = \beta \mod q \\ -1 & \text{otherwise} \end{cases}.$$

Thus, we can use the function $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\cdot)+1$ as the equality test function (scaled by q) in the following way. Given $\mathsf{RLWE.Enc}(\zeta_q^\alpha)$, we first multiply it by $\sum_\beta f(\beta) \cdot \zeta_q^{-\beta}$ and then perform the homomorphic trace evaluation (then plus $\sum_\beta f(\beta)$), resulting in a ciphertext of $\mathsf{RLWE.Enc}\left(\mathsf{Tr}\left(\sum_\beta \zeta_q^{\alpha-\beta} \cdot f(\beta)\right) + \sum_\beta f(\beta)\right)$. By the linearity of trace, the resulting plaintext would be $\sum_\beta \left(\mathsf{Tr}(\zeta_q^{\alpha-\beta}) + 1\right) \cdot f(\beta)$, and by the equality test's property, this would be equal to $q \cdot f(\alpha)$, successfully extracting the desired term (with the scaling factor q).

The Case of Prime-Power Cyclotomic Rings. In the prime-power setting, i.e., $q = p^r$ where p is any prime number and $r > 1$, the previously discussed

equality test method is no longer applicable. To handle this, we use the equality test observed in [1] that works over arbitrary cyclotomic rings: $\sum_{i=0}^{q-1} \zeta_q^{(\alpha-\beta)i}$, which equals to q if $\alpha = \beta \bmod q$ or otherwise 0. To homomorphically evaluate this equality test however, we need to overcome the following challenges.

Challenge 1. As suggested in [1], homomorphic evaluation of this equality test requires $O(q)$ homomorphic multiplications for a general q, which means directly applying their method would not meet our pre-set goal. Moreover, we cannot utilize the linearity of trace to evaluate all the equality tests in parallel, as we did in the prime case. However, we found that in the prime-power case, this equality test can be related to the algebraic trace and evaluated with only $O(\log q)$ homomorphic multiplications. Our first key observation is a partition for $\mathbb{Z}_q = \{0, 1, \ldots, p^r - 1\}$, which is $\mathbb{Z}_q \setminus \{0\} = \bigcup_{i=1}^{r} p^{r-i} \cdot \mathbb{Z}_{p^i}^*$. Then, we can derive a new equivalent expression for the original equality test (see Lemma 5.2):

$$\sum_{i \in \mathbb{Z}_q} \zeta_q^{(\alpha-\beta)\cdot i} = 1 + \sum_{i=1}^{r} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}} \left(\zeta_{p^i}^{\alpha-\beta} \right), \tag{1.1}$$

which relates the algebraic trace of sub-extensions to the original equality test.

Challenge 2. In the new formula, we need to compute encryptions of $\zeta_{p^i}^{\alpha-\beta}$. How to efficiently obtain these encryptions from $\mathsf{RLWE.Enc}(\zeta_q^{\alpha})$ is a new challenge, e.g., using $O(1)$ homomorphic-friendly operations. To address this issue, we observe that $\zeta_{p^i}^{\alpha} = \zeta_q^{p^{r-i}\cdot\alpha} = \zeta_q^{(p^{r-i}-1)\cdot\alpha} \cdot \zeta_q^{\alpha}$. Thus, we can first perform an automorphism evaluation of $\zeta_q \mapsto \zeta_q^{p^{r-i}-1}$ to get $\mathsf{RLWE.Enc}\left(\zeta_q^{(p^{r-i}-1)\cdot\alpha}\right)$ and then use a homomorphic multiplication with $\mathsf{RLWE.Enc}(\zeta_q^{\alpha})$ to obtain $\mathsf{RLWE.Enc}(\zeta_{p^i}^{\alpha})$.

Challenge 3. If we compute all the trace of sub-extensions in the summation separately, the overall computational complexity could become somewhat large. Fortunately, we further observe that all the trace evaluations in the summation are contained in the tower of field extensions $\mathbb{Q}(\zeta_{p^r})/\mathbb{Q}(\zeta_{p^{r-1}})/\cdots/\mathbb{Q}(\zeta_p)/\mathbb{Q}$. The summation can be computed by adding the encryptions of sub-ring elements (e.g., $\mathsf{RLWE.Enc}(\zeta_{p^i}^{\alpha-\beta})$) to the intermediate result during the evaluation of the trace tower. Consequently, we only need to evaluate the trace once.

In Sect. 5, we elaborate on the new techniques we developed to overcome all the abovementioned challenges.

The Case of Composite Cyclotomic Rings. In the composite setting, i.e., $q = q_1 \cdots q_k$ where the q_i's are distinct prime-powers $p_i^{r_i}$ for $i \in \{1, \ldots, k\}$, we notice that all the prior methods based on algebraic trace cannot serve as the equality test again. To achieve our goal, our key insight is to propose the following equation with two branches for the (scaled) equality test:

$$\prod_{i=1}^{k} \left(\sum_{j \in \mathbb{Z}_{q_i}} \zeta_{q_i}^{(\alpha-\beta)\cdot j} \right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

Intuitively, this design captures the idea that $\alpha = \beta \bmod q$ if and only if $\alpha = \beta \bmod q_i$ for all the branches modulo q_i by the Chinese Remainder Theorem, and each parenthesis is an equality test from [1]. Since each q_i is some prime-power, we can combine it with Eq. 1.1 to get the following equivalent expression:

$$\prod_{i=1}^{k}\left(\sum_{j\in\mathbb{Z}_{q_i}}\zeta_{q_i}^{(\alpha-\beta)\cdot j}\right)=\prod_{i=1}^{k}\left(1+\sum_{j=1}^{r_i}\mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}}\left(\zeta_{p_i^j}^{\alpha-\beta}\right)\right)$$

which relates the equality test to algebraic trace. To homomorphically compute this equality test however, we need to tackle various challenges.

Challenge 1. While we can utilize our method for the prime-power case to handle each branch, the outer product form seems to require additional homomorphic multiplications on the results of several trace functions. This may incur significant computational cost and noise blowup. To address this issue, we find a new equivalent expression that is the sum of several trace functions (see Lemma 6.3):

$$\prod_{i=1}^{k}\left(1+\sum_{j=1}^{r_i}\mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}}\left(\zeta_{p_i^j}^{\alpha-\beta}\right)\right)=1+\sum_{w|q,w\neq 1}\mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right).$$

Challenge 2. In the new formula, we need to compute encryptions of $\zeta_w^{\alpha-\beta}$. How to efficiently obtain these encryptions from $\mathsf{RLWE.Enc}(\zeta_q^{\alpha})$ is a new challenge. Similar to our solution to Challenge 2 for the prime-power case, we can write $\zeta_w^{\alpha} = \zeta_q^{(q/w)\cdot\alpha} = \zeta_q^{(q/w-1)\cdot\alpha}\cdot\zeta_q^{\alpha}$. Unfortunately, $\zeta_q \mapsto \zeta_q^{q/w-1}$ may not be an automorphism when q is general, so we use a more general formula that $\zeta_w^{\alpha} = \zeta_q^{(q/w)\cdot\alpha} = \zeta_q^{(q/w-c)\cdot\alpha}\cdot\zeta_q^{c\cdot\alpha}$ for some $c\in\mathbb{Z}_q$. If both $q/w-c$ and c are in $\mathbb{Z}_q^*$, we can use two automorphism evaluations plus one homomorphic multiplication to obtain the encryption of ζ_w^{α}. We prove that we can always find such c for any $w \mid q, w \neq 1, q$ when q is an odd composite number (see Proposition 6.4).

Challenge 3. There exist many different trace computations in the summation, some of which may not contained in a consecutive tower. It seems that we need to perform these trace evaluations individually. To further improve efficiency, we identify a new algebraic equation (see Lemma 6.6) that allows one single computation of $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}$, integrating all the intermediate trace computations.

In Sect. 6, we further describe our new techniques and designs to address all the aforementioned challenges.

Computational Complexity. In our constructions, the computational complexity is dominated by the homomorphic evaluation of the algebraic trace, which would require $N-1$ homomorphic automorphism evaluations in a trivial way where N is the degree of field extension. It is currently known that there are two typical cases where the trace evaluation can be completed with much fewer (e.g., $O(\log N)$) automorphism evaluations. The first case is when the extension $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ exhibits a tower structure which is widely used in [4,19,44,45]. The second case is when the extension $\mathbb{Q}(\zeta_q)/\mathbb{Q}$ is a cyclic extension (i.e., the

Galois group $\mathsf{Gal}(\mathbb{Q}(\zeta_q)/\mathbb{Q})$ is cyclic), which is first mentioned in [37] and generalized in [74]. These methods can be used to handle our prime-power case (as $\mathsf{Gal}(\mathbb{Q}(\zeta_q)/\mathbb{Q}) \cong \mathbb{Z}_q^*$ is cyclic for an odd prime-power q, and a power-of-2 q exhibits a base-2 logarithmic length tower of field extensions). It appears that there is currently no efficient solution for the composite case.

To address this, we have found that these two approaches can be combined. For example, suppose $q = q_1 q_2$ where q_1 and q_2 are two distinct odd prime-powers. Then we have the tower structure of extensions $\mathbb{Q}(\zeta_{q_1 q_2})/\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}$, which fits the first case. Moreover, the Galois groups $\mathsf{Gal}(\mathbb{Q}(\zeta_{q_1 q_2})/\mathbb{Q}(\zeta_{q_2})) \cong \mathbb{Z}_{q_1}^*$ and $\mathsf{Gal}(\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}) \cong \mathbb{Z}_{q_2}^*$ are both cyclic, which matches the second case. Thus, we can combine these two approaches to achieve a logarithmic complexity for the trace evaluation of the composite case. Furthermore, we found that this combination can be generalized to arbitrary cyclotomic extensions. We give an informal theorem below and refer to Sect. 3.3 for details.

Theorem 1.1 (Informal). *For a cyclotomic extension $\mathbb{Q}(\zeta_m)/\mathbb{Q}$ with $[\mathbb{Q}(\zeta_m) : \mathbb{Q}] = N$ and m being an arbitrary positive integer, $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}$ can be computed with $O(\log N)$ automorphisms. In the FHE context, homomorphic evaluation of $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}$ only requires $O(\log N)$ homomorphic automorphism evaluations.*

Why General Cyclotomic Rings?. Below, we discuss the rationale for considering general cyclotomic rings and further applications of our techniques.

- **More Modulus/Secret Choices:** In the FHEW/TFHE context, some recent works [43,70] have explored general LWE secret key distributions (as opposed to binary or ternary) to support more applications. In such cases, modulus switching may cause noise explosion when the norm of the secret key is relatively large. Hence, with the requirement that the LWE modulus q must divide the cyclotomic index m (to ensure the embedding $\mathbb{Z}_q \to \langle \zeta_m \rangle$), a flexible m allows for more options in the selection of q and secret key.
- **Compatibility with Batch Bootstrapping:** Our new algebraic insights are compatible with the Batch Bootstrapping framework of [44,45]. As the Batch paradigm crucially relies on tensor rings (including general cyclotomic rings for their tensor decompositions), our findings illuminate new paths for achieving SIMD functional bootstrapping for general functions within a polynomial modulus. Moreover, our strategy for trace computation can be employed in the framework to achieve the most flexible parameter choices.
- **General Applications:** Our new techniques are highly versatile and are applicable to schemes like BGV [13]/BFV [12,31], which inherently require general cyclotomic rings to support plaintext slots of finite fields or Galois rings [34,39,65]. Additionally, our new strategies for trace computation and new equality tests from insights on the structure of $\mathbb{Z}_q$ and $\mathbb{Z}_q^*$ may benefit other applications that rely on related computational number theory.

1.3 Other Related Work

A recent work [48] constructs new functional bootstrapping methods based on the BFV [12,31] scheme. However, their method would incur a super-polynomial

noise growth and thus require a super-polynomial modulus, which is not within the scope of the study in this work. Another recent work [52] improves the parameter selection and concrete efficiency of [26,47]. However, regarding functional bootstrapping for general functions, their algorithms do not show improvements in asymptotic complexity and functionality. Notably, the work [41] first discussed functional bootstrapping over general polynomial quotient rings, but their method fails to support both full-domain and general functions.

2 Preliminaries

2.1 Notations

In this paper, we denote the set of the rational numbers by $\mathbb{Q}$, the integers in $\mathbb{Q}$ by $\mathbb{Z}$, the real numbers by $\mathbb{R}$, and the complex numbers by $\mathbb{C}$. For an integer modulus q, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ is the quotient ring of integers modulo q. We use the representative set $\mathbb{Z}_q = \{0, 1, \ldots, q-1\}$ for simplicity and let $[x]_q$ denote the modulo q operation into $\mathbb{Z}_q$ for an integer x. Let $[n] = \{1, \ldots, n\}$, where n is a positive integer. Notation log refers to the base-2 logarithm unless explicitly specified otherwise. We denote $[a, b]$ as the set $[a, b] \cap \mathbb{Z}$ for any integers $a \leq b$. We denote a column vector by a bold lower-case letter, e.g. $\boldsymbol{x}$, and x_i to denote the i-th entry of $\boldsymbol{x}$. The transpose of $\boldsymbol{x}$, namely the corresponding row vector, is denoted by $\boldsymbol{x}^\top$. We define the ℓ_∞-norm of $\boldsymbol{x}$ by $\|\boldsymbol{x}\|_\infty = \max_i\{|x_i|\}$.

Given a set $\mathbb{A}$ and a distribution $\mathcal{P}$ over $\mathbb{A}$, we use $a \leftarrow \mathbb{A}$ to denote that a is uniformly chosen from $\mathbb{A}$ and $a \leftarrow \mathcal{P}$ to denote that a is chosen randomly according to the distribution $\mathcal{P}$.

2.2 Subgaussian Random Variables

We call a random variable X over $\mathbb{R}$ is subgaussian with parameter $s > 0$, if for all $t \in \mathbb{R}$, the (scaled) moment-generating function satisfies: $\mathbb{E}[e^{2\pi tX}] \leq e^{\pi s^2 t^2}$. Especially, any B-bounded symmetric random variable X(i.e., $\mathbb{E}[X] = 0$ and $|X| \leq B$) is subgaussian with parameter $B\sqrt{2\pi}$. Subgaussians satisfy the following properties as discussed in [5,30]:

- *Homogeneity*: If X is a subgaussian variable with parameter s, then cX is subgaussian with parameter cs for any positive $c \in \mathbb{R}$.
- *Pythagorean additivity*: For $s_i \geq 0$, and random variables X_i for $i \in [k]$, if X_i is subgaussian with parameter s_i conditioning on any values of $X_1, ..., X_{i-1}$, then $\sum_{i\in[k]} X_i$ is subgaussian with parameter $(\sum_{i\in[k]} s_i^2)^{1/2}$.
- *Boundedness*: For any subgaussian variable X with parameter s, we have the probability bound $\Pr[|X| > t] < 2 \cdot \exp(-\pi t^2/s^2)$.

Remark 2.1. *For a subgaussian variable X with parameter δ_x, we have $\Pr[|X| > C \cdot \delta_x] < 2 \cdot \exp(-\pi \cdot C^2)$ by the boundedness. Hence, by setting C to be a proper constant, we can deduce that $|X| \leq C \cdot \delta_x$ with overwhelming probability. For another subgaussian variable Y with parameter δ_y that is independent of X, we use a subgaussian variable with parameter $O(\delta_x \cdot \delta_y)$ as an upper bound to demonstrate the asymptotic behavior of $|X \cdot Y|$ in this paper.*

2.3 Cyclotomic Rings

Let ζ_m be an m-th primitive root of unity. Then $K = \mathbb{Q}(\zeta_m)$ is an algebraic number field known as the m-th cyclotomic field, where the number m is referred to as the cyclotomic index. From algebraic number theory, the ring of the integers of the field K, which we usually denote by $\mathcal{O}_K$, is $\mathbb{Z}[\zeta_m]$. Note that we have $\mathbb{Z}[\zeta_m] \cong \mathbb{Z}[X]/(f(X))$ where $f(X)$ is the minimal polynomial of ζ_m with degree $N = \phi(m)$ (the Euler's totient of m). In the cyclotomic extension case, $f(X)$ is the m-th cyclotomic polynomial $\Phi_m(X) = \prod_{i \in \mathbb{Z}_m^*} (X - \omega_m^i) \in \mathbb{Z}[X]$, where $\mathbb{Z}_m^*$ denotes the set of integers in $\mathbb{Z}_m$ that are coprime to m, and $\omega_m \in \mathbb{C}$ is any primitive m-th complex root of unity, e.g., $\omega_m = e^{2\pi\sqrt{-1}/m}$.

Let $\mathcal{R} = \mathbb{Z}[\zeta_m]$ and $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Then the set $\{1, \zeta_m, \ldots, \zeta_m^{N-1}\}$ forms a $\mathbb{Z}$-basis of $\mathcal{R}$ and thus a $\mathbb{Z}_q$-basis of $\mathcal{R}_q$. This basis is often called the power basis of $\mathcal{R}$. For $a \in \mathcal{R}$, we can uniquely write it as $a = a_0 + a_1\zeta_m + \cdots + a_{N-1}\zeta_m^{N-1}$ where $a_i \in \mathbb{Z}$ for $i = 0, 1, \ldots, N-1$. We call $(a_0, \ldots, a_{N-1})$ the representation or the coefficient embedding under the power basis.

Canonical Embedding.T he m-th cyclotomic number field $K = \mathbb{Q}(\zeta_m)$ of degree $N = \phi(m)$ has exactly N ring embeddings $\sigma_i : K \to \mathbb{C}$ that fix every element of $\mathbb{Q}$. Let these embeddings be indexed by $\mathbb{Z}_m^*$. Then for $i \in \mathbb{Z}_m^*$, each embedding σ_i is defined by $\sigma_i(\zeta_m) = \omega_m^i$ where $\omega_m \in \mathbb{C}$ is some fixed complex primitive m-th root of unity (e.g., $\omega_m = e^{2\pi\sqrt{-1}/m}$). Then the canonical embedding $\sigma : K \to \mathbb{C}^N$ is defined as

$$\sigma(x) = (\sigma_i(x))_{i \in \mathbb{Z}_m^*}.$$

Note that it is a ring homomorphism from K to $\mathbb{C}^N$, where addition and multiplication in the latter are both component-wise. For $a \in \mathcal{R}$, the canonical embedding norm of a is defined as the ℓ_∞-norm of $\sigma(a)$, namely, $\|\sigma(a)\|_\infty$. It possesses the following nice property: For $a, b \in \mathcal{R}$, we have

- $\|\sigma(a+b)\|_\infty \leq \|\sigma(a)\|_\infty + \|\sigma(b)\|_\infty$,
- $\|\sigma(a \cdot b)\|_\infty \leq \|\sigma(a)\|_\infty \cdot \|\sigma(b)\|_\infty$.

Due to the independence of the representation of elements in $\mathcal{R}$ and the above property, we can easily bound the canonical embedding norm for elements in general cyclotomic rings. One can refer to [39,51] for the relation between the canonical embedding norm and the norm under some $\mathbb{Z}$-basis of $\mathcal{R}$.

Algebraic Trace. For two number fields $K' \subset K$, suppose the field extension K over K' (denoted as K/K') is a Galois extension with the Galois group $\mathsf{Gal}(K/K')$. Then for any element $a \in K$, the trace of a over K' is defined as

$$\mathsf{Tr}_{K/K'}(a) = \sum_{\tau \in \mathsf{Gal}(K/K')} \tau(a) \in K'.$$

An important fact is that $\mathsf{Tr}_{K/K'}(\mathcal{O}_K) \subseteq \mathcal{O}_{K'}$. Moreover, the trace $\mathsf{Tr}_{K/K'}(\cdot)$ has the K'-*linearity* as follows.

– For $a \in K, c \in K'$, we have $\mathsf{Tr}_{K/K'}(c \cdot a) = c \cdot \mathsf{Tr}_{K/K'}(a)$.
– For $a, b \in K$, we have $\mathsf{Tr}_{K/K'}(a + b) = \mathsf{Tr}_{K/K'}(a) + \mathsf{Tr}_{K/K'}(b)$.

For a tower of number field extensions $K_r/K_{r-1}/\cdots/K_2/K_1$, the trace has the following property, which is the so-called *transitivity*:

$$\mathsf{Tr}_{K_r/K_1}(a) = \mathsf{Tr}_{K_2/K_1}(\cdots(\mathsf{Tr}_{K_{r-1}/K_{r-2}}(\mathsf{Tr}_{K_r/K_{r-1}}(a)))\cdots)$$

for any $a \in K_r$. Moreover, we have the following useful fact for computation.

Lemma 2.2 ([51]). *Let m be a power of a prime p and $m' = m/p$, then for $i \in \mathbb{Z}$,*

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}(\zeta_m^i) = \begin{cases} \varphi(p) \cdot m' & \text{if } \ i = 0 \bmod m \\ -m' & \text{if } \ i = 0 \bmod m' \ \text{ and } \ i \neq 0 \bmod m \\ 0 & \text{otherwise.} \end{cases}$$

2.4 (Ring) Learning with Errors

The learning with errors (LWE) problem was first introduced by Regev [63]. Before the definition of LWE, we first introduce the distribution $A_{\boldsymbol{s},\chi}$. For a distribution χ over $\mathbb{Z}$ and a vector $\boldsymbol{s} \in \mathbb{Z}_q^n$, a sample from the distribution $A_{\boldsymbol{s},\chi}$ is of the form $(b, \boldsymbol{a}) \in \mathbb{Z}_q \times \mathbb{Z}_q^n$ with $b = [\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e]_q$, where $\boldsymbol{a} \leftarrow \mathbb{Z}_q^n$ and $e \leftarrow \chi$.

Definition 2.3 (DLWE). *For a security parameter λ, let $n := n(\lambda)$ be an integer dimension, let $q = q(\lambda) \geq 2$ be an integer modulus, and let $\chi = \chi(\lambda)$ be an error distribution over $\mathbb{Z}$. Given some independent samples from $A_{\boldsymbol{s},\chi}$, the decision version of* LWE*, denoted by* $\mathsf{DLWE}_{n,q,\chi}$*, is to distinguish them from the same number of uniformly random and independent samples from $\mathbb{Z}_q \times \mathbb{Z}_q^n$.*

The $\mathsf{DLWE}_{n,q,\chi}$ problem defined above is known to be at least as hard as certain lattice problems [14,59,63]. To improve the efficiency of LWE-based schemes, the ring version of LWE, namely RLWE, was introduced [50,66]. We use the primal version of RLWE where the secret is defined in the ring rather than its dual. More discussions on different variants of RLWE can be found in [18,29,60,61].

For a distribution χ over $\mathcal{R}$ and a ring element $z \in \mathcal{R}_q$, a sample from the distribution $\mathcal{A}_{z,\chi}$ is of the form $(b, a) \in \mathcal{R}_q^2$ with $b = a \cdot z + e$ where $a \leftarrow \mathcal{R}_q$ and $e \leftarrow \chi$.

Definition 2.4 (RLWE). *For a security parameter λ, let $N := N(\lambda)$ be the degree of the ring, let $q = q(\lambda) \geq 2$ be an integer modulus, and let $\chi = \chi(\lambda)$ be an error distribution χ over $\mathcal{R}$. Given some independent samples from $\mathcal{A}_{z,\chi}$, the decision version of* RLWE*, denoted by* $\mathcal{R}$-$\mathsf{DLWE}_{N,q,\chi}$*, is to distinguish them from the same number of uniformly random and independent samples from $\mathcal{R}_q^2$.*

There are reductions showing that the $\mathcal{R}$-$\mathsf{DLWE}_{N,q,\chi}$ problem defined above is at least as hard as certain computational problems in ideal lattices [50,62].

2.5 (Ring) LWE-Based Symmetric Encryption

We first review a symmetric encryption scheme based on LWE, which is the base scheme to be bootstrapped. We use the most significant bits (MSBs) for the plaintext encoding for ease of description. Note that the algorithms we propose in this work are independent of the plaintext encoding methods for the input LWE scheme. The basic definitions of LWE-based encryption are as follows.

- **Encryption.** Let $\mathbb{Z}_t$ be the plaintext domain, and q be the LWE modulus. Then the set of valid LWE ciphertexts for plaintext $\mu \in \mathbb{Z}_t$ under the secret key $\boldsymbol{s}$, denoted as $\mathsf{LWE}_{\boldsymbol{s}}(\delta \cdot \mu)$[5] is defined as:

$$\mathsf{LWE}_{\boldsymbol{s}}(\delta \cdot \mu) = \{([\langle \boldsymbol{a}, \boldsymbol{s} \rangle + e + \delta \cdot \mu]_q, \boldsymbol{a}) \in \mathbb{Z}_q^{n+1}\},$$

  where $\boldsymbol{a} \in \mathbb{Z}_q^n$ and $e \leftarrow \chi$ for some typical error distribution χ over $\mathbb{Z}$ (e.g., Gaussian), and $\delta = \lfloor \frac{q}{t} \rceil$ is the scaling factor.
- **Decryption.** A ciphertext $\boldsymbol{c} = (b, \boldsymbol{a}) \in \mathbb{Z}_q^{n+1}$ can be decrypted by computing

$$\mathsf{Dec}(\boldsymbol{s}, \boldsymbol{c}) = \mathsf{Dcd}([b - \langle \boldsymbol{a}, \boldsymbol{s} \rangle]_q)$$

  where $\mathsf{Dcd} : \mathbb{Z}_q \to \mathbb{Z}_t$ is the decoding function $\mathsf{Dcd}(x) = [\lfloor \frac{t}{q} \cdot x \rceil]_t$.

The above notation can be extended to RLWE-based schemes. Let $\mathcal{R}_p$ be the plaintext space and Q be the ciphertext modulus. Then, we define the following set for valid RLWE encryptions of the plaintext $\mu \in \mathcal{R}_p$ under the secret key z:

$$\mathsf{RLWE}_z(\Delta \cdot \mu) = \left\{([a \cdot z + e + \Delta \cdot \mu]_Q, a) \in \mathcal{R}_Q^2\right\},$$

where $a \in \mathcal{R}_Q$ and $e \leftarrow \chi$ for some typical error distribution χ over $\mathcal{R}$ (e.g., Gaussian), and $\Delta = \lfloor \frac{Q}{p} \rceil$ is the scaling factor. Note that $\lfloor \cdot \rceil$ and $[\cdot]_Q$ for elements in $\mathcal{R}$ mean coordinate-wise rounding to the nearest integer and coordinate-wise modulo Q with respect to some fixed $\mathbb{Z}$-basis of $\mathcal{R}$, respectively. We use the following notation to denote the error in a ciphertext.

Definition 2.5. *For a ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\mu)$, the error of $\boldsymbol{c}$ is defined as*

$$\mathsf{Err}(\boldsymbol{c}) := \langle (1, -z), \boldsymbol{c} \rangle - \mu.$$

3 Basic Homomorphic Computations

In this section, we review some necessary background on homomorphic encryption, encompassing homomorphic operations, along with some new techniques.

We start with the homomorphic operations on RLWE ciphertexts used in this work. More operations, including the external product with Ring-GSW [23,30,35] ciphertexts over general cyclotomic rings can be found in [44,45]. We assume that the underlying ring for the RLWE scheme is $\mathcal{R} = \mathbb{Z}[\zeta_m]$ with $N = \phi(m)$, and the ciphertext modulus is Q. The proofs of all lemmas, propositions, and theorems in this section are provided in the full version.

[5] An equivalent notation $\mathsf{LWE}_{\boldsymbol{s}}^{t/q}(\mu)$ is also used in the literature.

3.1 BFV Homomorphic Multiplication

Let $\mathsf{BFV.Mul}(\cdot)$ be a homomorphic multiplication algorithm [12,31] that takes as input two RLWE ciphertexts $\boldsymbol{c}_i \in \mathsf{RLWE}_z(\Delta \cdot \mu_i)$ where $i = 1, 2$ and $\Delta := \lfloor Q/p \rceil$, and some relinearization key RelKey, and outputs a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\Delta \cdot \mu_1 \cdot \mu_2)$. Since we will only encounter the case where the plaintext is some root of unity, the following lemma is restricted to this case only for a simpler error bound.

Lemma 3.1. *Suppose* $\boldsymbol{c}' \leftarrow \mathsf{BFV.Mul}(\boldsymbol{c}_1, \boldsymbol{c}_2)$ *and both* $\boldsymbol{c}_1, \boldsymbol{c}_2$ *are encryptions of roots of unity. If* $\|\sigma(\mathsf{Err}(\boldsymbol{c}_1))\|_\infty$ *and* $\|\sigma(\mathsf{Err}(\boldsymbol{c}_2))\|_\infty$ *are upper bounded by subgaussian variables with parameter* δ_1, δ_2, $\|\sigma(\mathsf{Err}(\mathsf{RelKey}))\|_\infty = \|\sigma(z)\|_\infty = O(\sqrt{N})$. *Then* $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ *is upper bounded by a subgaussian variable with parameter* $O(\sqrt{(p/Q)^2\delta_1^2\delta_2^2 + p^2N^2(\delta_1^2+\delta_2^2) + N^2 \log Q})$.

3.2 Homomorphic Automorphism Evaluation

We also use homomorphic automorphism evaluation over general cyclotomic rings as studied and used in [37,39,44,45]. Let $\mathsf{EvalAuto}(\cdot)$ be a homomorphic evaluation algorithm that takes as input a RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\mu)$, an automorphism $\tau \in \mathsf{Gal}(\mathbb{Q}(\zeta_m)/\mathbb{Q})$ and some automorphism key AutKey, and outputs a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\tau(\mu))$. The following lemma demonstrates the error growth in the homomorphic automorphism evaluation.

Lemma 3.2. *Suppose that* $\boldsymbol{c}' \leftarrow \mathsf{EvalAuto}(\boldsymbol{c}, \tau, \mathsf{AutKey})$. *If* $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$ *and* $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ *are upper bounded by subgaussian variables with parameter* δ_c *and* $\delta_{\mathtt{aut}}$ *respectively, then* $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ *is upper bounded by a subgaussian variable with parameter* $O(\sqrt{\delta_c^2 + \delta_{\mathtt{aut}}^2 N \log Q})$.

3.3 Homomorphic Trace Evaluation

By definition, the trace evaluation requires performing all automorphisms in the Galois group $\mathsf{Gal}(\mathbb{Q}(\zeta_m)/\mathbb{Q})$ and summing the results, which trivially necessitates $N-1$ automorphism evaluations. However, it is currently known that there are two typical cases where the trace evaluation can be computed with much fewer (e.g., $O(\log N)$) automorphism evaluations: (1) when the extension $\mathbb{Q}(\zeta_m)/\mathbb{Q}$ exhibits a tower structure [4,19,44,45], and (2) when the extension $\mathbb{Q}(\zeta_m)/\mathbb{Q}$ is a cyclic extension, i.e., $\mathsf{Gal}(\mathbb{Q}(\zeta_m)/\mathbb{Q})$ forms a cyclic group [37,74]. We first discuss each case below in a more general style (i.e., using relative field extensions).

The Tower of Extensions Case. Suppose the tower of number field extensions $\mathbb{Q}(\zeta_m) = K_r/K_{r-1}/\cdots/K_1/K_0 = \mathbb{Q}$ where each K_i/K_{i-1} is a Galois extension for all $i \in [r]$. For $0 \le j < i \le r$, let $\mathsf{EvalTr}_{K_i/K_j}(\cdot)$ be a homomorphic evaluation algorithm that takes as input a RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\mu)$ and some automorphism key AutKey, and outputs a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\mathsf{Tr}_{K_i/K_j}(\mu))$.

Then, owing to the transitivity of the trace, $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}(\mu) = \mathsf{Tr}_{K_r/K_0}(\mu)$ can be homomorphically evaluated as follows (we omit AutKey for simplicity):

$$\mathsf{EvalTr}_{K_1/K_0}\left(\mathsf{EvalTr}_{K_2/K_1}\left(\cdots \mathsf{EvalTr}_{K_r/K_{r-1}}(\boldsymbol{c})\cdots\right)\right).$$

Denote the degree of the extensions by $[K_i : K_{i-1}] = d_i$ for $i \in [r]$ (hence $N = \prod_{i\in[r]} d_i$), thus the above sequence of homomorphic computation requires only $\sum_{i\in[r]}(d_i - 1)$ automorphism evaluations (-1 comes from the identity maps). The following lemma provides the error growth of trace evaluation in this case.

Lemma 3.3. *Suppose that* $\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{K_i/K_j}(\boldsymbol{c}, \mathsf{AutKey})$. *If* $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$ *and* $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ *are upper bounded by subgaussian variables with parameters* δ_c *and* $\delta_{\mathtt{aut}}$, *respectively. Then* $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ *is upper bounded by a subgaussian variable with parameter* $O(\sqrt{d\delta_c^2 + (d-1)\delta_{\mathtt{aut}}^2 N \log Q})$ *where* $d = [K_i : K_j]$.

The Cyclic Extension Case. Suppose the Galois extensions $\mathbb{Q}(\zeta_m)/K/F/\mathbb{Q}$ with $[K : F] = M$ and $\mathsf{Gal}(K/F)$ being a cyclic group with generator τ_g, then we can write $\mathsf{Gal}(K/F) = \{\mathsf{id} = \tau_g^0, \tau_g, \tau_g^2, \ldots, \tau_g^{M-1}\}$ where id is the identity map. Let $T_k(\mu) = \sum_{i\in[0,k-1]} \tau_g^i(\mu)$ for $\mu \in F$, then we have

$$T_k(\mu) = \begin{cases} T_{k/2}(\mu) + \tau_g^{k/2}(T_{k/2}(\mu)) & \text{if } k \text{ is even} \\ T_{(k-1)/2}(\mu) + \tau_g^{(k-1)/2}\left(T_{(k-1)/2}(\mu)\right) + \tau_g^{k-1}(\mu) & \text{if } k \text{ is odd} \end{cases}.$$

Consequently, we can reduce the scale of the required automorphism summations in a recursive way for the homomorphic evaluation of $\mathsf{Tr}_{K/F}(\mu) = T_M(\mu)$. It has been proved in [74] that the homomorphic evaluation of $T_M(\mu)$ requires at most $2\log M$ (which is essentially $O(\log M)$) automorphism evaluations. The following useful fact captures the case of cyclotomic extensions.

Fact 3.4 ([69]) *For a cyclotomic field extension* K/F *with* $K = F(\zeta_t)$ *and* $[K : F] = \phi(t)$, *we have* $\mathsf{Gal}(K/F) \cong \mathbb{Z}_t^*$, *which is cyclic if and only if* $t = 1, 2, 4, p^r, 2p^r$ *where* p *is an odd prime and* r *is some positive integer.*

Let $\mathsf{EvalTr}_{K/F}(\cdot)$ be a homomorphic evaluation algorithm that takes as input a RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\mu)$ and some automorphism key AutKey, and outputs a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\mathsf{Tr}_{K/F}(\mu))$. The following lemma provides the error growth of trace evaluation in the cyclic extension case.

Lemma 3.5 *Suppose that* $\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{K/F}(\boldsymbol{c}, \mathsf{AutKey})$. *If* $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$ *and* $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ *are upper bounded by subgaussian variables with parameters* δ_c *and* $\delta_{\mathtt{aut}}$, *respectively. Then* $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ *is upper bounded by a subgaussian variable with parameter* $O(\sqrt{M\delta_c^2 + (M-1)\delta_{\mathtt{aut}}^2 N \log Q})$.

Our Combination. By integrating the above two methods, we can construct efficient trace evaluation algorithms over arbitrary cyclotomic rings. Specifically, let the cyclotomic index m be an arbitrary positive integer. By the fundamental

theorem of arithmetic, we can write $m = \prod_{i\in[r]} p_i^{e_i}$ where p_i's are distinct primes, and e_i's are positive integers. Then denote $K_i = \mathbb{Q}\left(\zeta_{\prod_{j=1}^{i} p_j^{e_j}}\right)$ for $i \in [r]$, we have the tower of extensions $\mathbb{Q}(\zeta_m) = K_r/K_{r-1}/\cdots/K_1/K_0 = \mathbb{Q}$. To further compute $\mathsf{Tr}_{K_i/K_{i-1}}$ in the tower, we have the following discussion that covers all possible cases based on whether m is odd or even.

- **Case 1:** If $p_i \neq 2$ for all $i \in [r]$, we have $K_i = K_{i-1}\left(\zeta_{p_i^{e_i}}\right)$ which implies $\mathsf{Gal}(K_i/K_{i-1}) \cong \mathbb{Z}_{p_i^{e_i}}$ and thus K_i/K_{i-1} is a cyclic extension by Fact 3.4. Hence, each $\mathsf{Tr}_{K_i/K_{i-1}}$ can be computed with $O\left(\log \phi(p_i^{e_i})\right)$ automorphisms, yielding a total of $\sum_{i\in[r]} O\left(\log \phi(p_i^{e_i})\right) = O(\log N)$ for $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}$.
- **Case 2:** If $p_i = 2$ for some $i \in [r]$, suppose without loss of generality that $p_1 = 2$ and $K_1 = \mathbb{Q}(\zeta_{2^{e_1}})$. Then we have a second layer of tower extensions: $K_1 = \mathbb{Q}(\zeta_{2^{e_1}})/\mathbb{Q}(\zeta_{2^{e_1-1}})/\cdots/\mathbb{Q}(\zeta_2) = \mathbb{Q}$, which implies $\mathsf{Tr}_{K_1/\mathbb{Q}}$ requires $O(\log 2^{e_1})$ automorphisms. As discussed in Case 1, each $\mathsf{Tr}_{K_i/K_{i-1}}$ requires $O\left(\log \phi(p_i^{e_i})\right)$ automorphisms for $i > 1$, yielding a total of $O(\log N)$ for $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}$.

All these strategies can be naturally extended to homomorphic evaluations of $\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}$ in the ciphertext domain for arbitrary cyclotomic index m. We use the following theorem to summarize the above discussion.

Theorem 3.6 *Suppose the underlying ring $\mathcal{R} = \mathbb{Z}[\zeta_m]$ with m being an arbitrary positive integer. For a* RLWE *ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\mu)$ and some automorphism key* AutKey*, there exist efficient algorithms $\mathsf{EvalTr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}(\boldsymbol{c}, \mathsf{AutKey})$ that output $\boldsymbol{c}' \in \mathsf{RLWE}_z(\mathsf{Tr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}(\mu))$ using only $O(\log N)$ automorphism evaluations.*

Moreover, if $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$ and $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ are upper bounded by subgaussian variables with parameters δ_c and $\delta_{\mathtt{aut}}$, respectively. Then, for both Case 1 and 2, the output error norm $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ is upper bounded by a subgaussian variable with parameter $O(\sqrt{N\delta_c^2 + (N-1)\delta_{\mathtt{aut}}^2 N \log Q})$.

3.4 Computational Complexity

For the sake of comparison, we standardize the units of measurement for the computational complexity of homomorphic operations. Same as the FHEW/TFHE bootstrapping algorithms, the most time-consuming operation in our algorithms is the external product [22,23,44,45]. Thus, we propose the following proposition that demonstrates the relationship between the costs of the above homomorphic operations and the external product.

Proposition 3.7 *Suppose the same underlying ring $\mathcal{R} = \mathbb{Z}[\zeta_m]$ with the same modulus for all homomorphic operations. We have the following approximations:*

- *For the same gadget decomposition base, both* $\mathsf{BFV.Mul}(\cdot)$ *and* $\mathsf{EvalAuto}(\cdot)$ *require approximately $1/2$ times the cost of the external product.*
- *For different gadget decomposition bases, both* $\mathsf{BFV.Mul}(\cdot)$ *and* $\mathsf{EvalAuto}(\cdot)$ *require $O(1)$ times the cost of the external product.*

Consequently, $\mathsf{EvalTr}_{\mathbb{Q}(\zeta_m)/\mathbb{Q}}(\cdot)$ *requires $O(\log N)$ times the cost of the external product for arbitrary positive integer m.*

4 Functional Bootstrapping: A Warm-Up

In this section, we first review the techniques in the FHEW/TFHE bootstrapping framework, clarifying the context in which our algorithm will be operational. Then, we present the core building block of our functional bootstrapping algorithm over prime cyclotomic rings, which constitutes the simplest case.

4.1 The Functional Bootstrapping Framework

Recall the task of the functional bootstrapping – basically, the algorithm takes input an LWE ciphertext $\boldsymbol{c}$ with some bootstrapping key and returns an LWE ciphertext $\boldsymbol{c}'$ that encrypts $f(\mathsf{Dec}(\boldsymbol{c}))$ for some pre-determinded function f.

Parameters. We first describe parameters used in the FHEW/TFHE (functional) bootstrapping and this work. Note that all the parameters with magnitudes (i.e., n, q, t, m, Q, h) are polynomially bounded in the security parameter.

- n: The dimension of the input LWE scheme.
- q: The modulus of the input LWE scheme.
- t: The plaintext modulus of the input LWE scheme.
- $\boldsymbol{s}$: The secret key of the input LWE scheme.
- $\mathcal{R}$: The underlying ring $\mathbb{Z}[\zeta_m]$ of the RLWE scheme with $q \mid m$ and $N := \phi(m)$.
- Q: The modulus of the RLWE scheme.
- z: The secret key of the RLWE scheme.
- h: The plaintext modulus of the output LWE ciphertext.

The FHEW/TFHE Framework. On input an LWE ciphertext $\boldsymbol{c} = (b, \boldsymbol{a}) \in \mathsf{LWE}_{\boldsymbol{s}}^{t/q}(\mu)$, the bootstrapping algorithm first performs a Blind Rotation to obtain a RLWE ciphertext $\tilde{\boldsymbol{c}} \in \mathsf{RLWE}_z(v \cdot \zeta_q^{\varphi})$ where $\varphi = [b - \langle \boldsymbol{a}, \boldsymbol{s} \rangle]_q$ and $v \in \mathcal{R}_Q$ is some encoding of the composite of a negacyclic function $f : \mathbb{Z}_t \to \mathbb{Z}_h$ and the decoding function Dcd. The constant term (i.e., the coefficient of the basis 1) of the plaintext of $\tilde{\boldsymbol{c}}$ is actually $\Delta' \cdot f(\mathsf{Dcd}(\varphi)) = \Delta' \cdot f(\mu)$ where $\Delta' \approx Q/h$. Thus, the algorithm proceeds a Sample Extract to obtain an LWE ciphertext $\boldsymbol{c}' \in \mathsf{LWE}_{\boldsymbol{z}}^{h/Q}(f(\mu))$ where $\boldsymbol{z}$ is the coefficient vector of z. Finally, it performs a sequence of modulus switching, key switching, and modulus switching to obtain a ciphertext in $\mathsf{LWE}_{\boldsymbol{s}}^{h/q}(f(\mu))$. The framework is illustrated in Fig. 1. We state the functionality and error analysis of the Blind Rotation in the following lemma and refer to the details and proof in [23] and the full version of this work.

Lemma 4.1 (Blind Rotation [23], Adapted) *For a ciphertext $\boldsymbol{c} = (b, \boldsymbol{a}) \in \mathsf{LWE}_{\boldsymbol{s}}^{t/q}(\mu)$ where $\boldsymbol{s} \in \{0,1\}^n$, there exists an algorithm $\mathsf{BlindRotate}$ that takes input $\boldsymbol{c}$ and the bootstrapping key BK, and outputs a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(v \cdot \zeta_q^{b - \langle \boldsymbol{a}, \boldsymbol{s} \rangle})$ for arbitrary $v \in \mathcal{R}_Q$, requiring n times external product. If $\|\sigma(\mathsf{Err}(\mathsf{BK}))\|_\infty$ is upper bounded by a subgaussian variable with parameter δ_{bk} for all $i \in [n]$, then $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ is upper bounded by a subgaussian variable with parameter $O(\delta_{\mathsf{bk}} \sqrt{nN \log Q})$.*

Remark 4.2 *The secret $\boldsymbol{s}$ can be ternary or general distributions as well. Several recent results [9,43,53,70,71] show how to do Blind Rotation for these cases with comparable (or slightly less) efficiency as the case of binary secrets.*

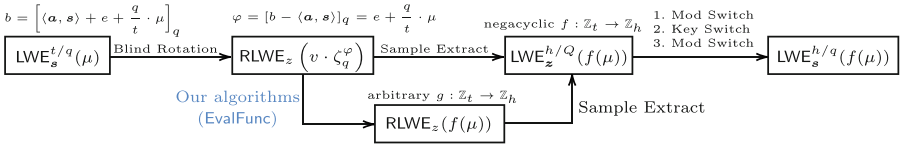

Fig. 1. The relation of FHEW/TFHE bootstrapping framework and our algorithms.

Our Blueprint. As also depicted in Fig. 1, our approach broadly adheres to the same line as the FHEW/TFHE framework. The sole distinction lies in that, subsequent to obtaining the ciphertext through Blind Rotation, we construct a series of new algorithms $\mathsf{EvalFunc}$, each of which works over different cyclotomic rings and only incurs polynomial error growth. The algorithms take as input a ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(v \cdot \zeta_q^{\varphi})$ for some $v \in \mathcal{R}_Q$ and some auxiliary keys, and output a RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(f(\varphi))$ for an arbitrary function $f : \mathbb{Z}_q \to \mathbb{Z}_h$. Let $f = g \circ \mathsf{Dcd}$ for an arbitrary function $g : \mathbb{Z}_t \to \mathbb{Z}_h$, then $\boldsymbol{c}'$ actually encrypts $f(\varphi) = g(\mathsf{Dcd}(\varphi)) = g(\mu)$.

Our advantage of computational efficiency stems from the fact that the algorithm $\mathsf{EvalFunc}$ requires only $O(\log \phi(q))$ times the cost of the external product. By adopting the typical parameter setting in FHEW/TFHE bootstrapping that $q = O(n)$, the overall cost is approximately $n + O(\log n)$ times the cost of the external product. Considering that regular FHEW/TFHE bootstrapping requires n external products plus minimal overhead, our findings suggest that functional bootstrapping for arbitrary functions is essentially as efficient as regular bootstrapping. In other words, the ratio of efficiency between functional and regular FHEW/TFHE bootstrapping approaches $1 + o(1)$.

The functional superiority, such as the support for arbitrary input plaintext modulus/encoding and general functions, is derived from our adoption of new equality test techniques. Specifically, our constructions rely on the fact that any discrete function can be expressed by a linear combination of the equality test function. Define the equality test for the exponent of ζ_q as

$$\mathsf{EqT}(\zeta_q^{\alpha}, \beta) = \begin{cases} 1 & \text{if} \quad \alpha = \beta \mod q \\ 0 & \text{if} \quad \alpha \neq \beta \mod q \end{cases},$$

then for any $\alpha \in \mathbb{Z}_q$, we have $f(\alpha) = \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \mathsf{EqT}(\zeta_q^{\alpha}, \beta)$.

Remaining Tasks. Now, our remaining task is to find efficient solutions for the homomorphic evaluation of EqT over cyclotomic rings, based on which we can instantiate the algorithm $\mathsf{EvalFunc}$. In the following subsection, we present an instantiation of $\mathsf{EvalFunc}$ over prime cyclotomic rings, and subsequently, we will delve into other instantiations for more general cases.

Algorithm 4.1. $\mathsf{EvalFunc}(\boldsymbol{c}, \mathsf{AutKey}, f)$

Parameters:

Δ: the scaling factor $\lfloor Q/(h \cdot q) \rceil$ of the input encoding
Δ': the scaling factor $\Delta \cdot q \approx Q/h$ of the output encoding
q: a prime number satisfying $q \mid m$
h: the plaintext modulus of the output ciphertext
v: an encoding of the function $v := \Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_q^{-\beta} \in \mathcal{R}_Q$

Input:

A RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(v \cdot \zeta_q^{\alpha})$
The key for homomorphic automorphism evaluation AutKey
An arbitrary function $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: A RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\Delta' \cdot f(\alpha))$

1: $\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\boldsymbol{c}, \mathsf{AutKey})$
2: $\boldsymbol{c}' \leftarrow \boldsymbol{c}' + (\Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta), 0)^{\top}$
3: **return** $\boldsymbol{c}'$

4.2 Arbitrary Function Evaluation over Prime Cyclotomic Rings

We first consider the simple case where q is prime. From Lemma 2.2, we identify the following equation that can serve as the equality test: For any $\alpha, \beta \in \mathbb{Z}_q$,

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^{\alpha-\beta}) + 1 = \begin{cases} q & \text{if} \quad \alpha = \beta \mod q \\ 0 & \text{if} \quad \alpha \neq \beta \mod q \end{cases}. \tag{4.1}$$

Then, we present the instantiation of $\mathsf{EvalFunc}$ in Algorithm 4.1 and its correctness together with analysis in Theorem 4.3.

Theorem 4.3 *Algorithm 4.1 is correct, i.e., the input-output behavior satisfies as described. Moreover, it possesses the following properties:*

- *Complexity: it requires $O(\log \phi(q))$ times the cost of the external product.*
- *Error growth: if $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$ and $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ are upper bounded by subgaussian variables with parameters δ_c and $\delta_{\mathtt{aut}}$. Then $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ is upper bounded by a subgaussian variable with parameter*

$$O\left(\sqrt{(q-1)\delta_c^2 + (q-2)\delta_{\mathtt{aut}}^2 N \log Q}\right).$$

Proof. A simple calculation yields that in *line* 2 we have $\boldsymbol{c}'$ encrypts

$$\begin{aligned} & \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_q^{\alpha-\beta}\right) + \Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta) \\ &= \Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^{\alpha-\beta}) + 1\right) && \text{(by } \mathbb{Q}\text{-linearity)} \\ &= \Delta \cdot f(\alpha) \cdot q && \text{(by Eq. 4.1)} \\ &= \Delta' \cdot f(\alpha) && \text{(by the definition of } \Delta' := \Delta \cdot q) \end{aligned}$$

Since the errors can be upper bounded by subgaussian variables, we summarize the corresponding subgaussian parameter in each line as follows.

- $\boldsymbol{c}'$ in *line* 1: $O\big(\sqrt{(q-1)\delta_c^2 + (q-2)\delta_{\mathtt{aut}}^2 N \log Q}\big)$ by Lemma 3.5.
- $\boldsymbol{c}'$ in *line* 2: Adding an error-free ciphertext does not affect the error.

This proves the claim of error growth. The claim of computational complexity follows directly from Theorem 3.6 and Proposition 3.7. □

5 The Case of Prime-Power Cyclotomic Rings

In this section, we focus on the case of prime-power cyclotomic rings, namely $q = p^r$ for any prime number p and integer $r > 1$. In this case, the previously discussed instantiation for the prime case based on trace plus one is no longer applicable (see Lemma 2.2). Alternatively, we consider another instantiation of equality test observed in [1] that works over arbitrary cyclotomic rings:

$$\sum_{i=0}^{q-1} \zeta_q^{(\alpha-\beta)\cdot i} = 1 + \zeta_q^{\alpha-\beta} + \cdots + \zeta_q^{(q-1)(\alpha-\beta)} = \begin{cases} q & \text{if } \alpha = \beta \bmod q \\ 0 & \text{if } \alpha \neq \beta \bmod q \end{cases}. \tag{5.1}$$

To homomorphically evaluate this equality test however, we need to confront the following challenges.

Challenge 1. As also suggested in [1], homomorphic evaluation of the above equality test given an encryption of $\zeta_q^{\alpha-\beta}$ requires $O(q)$ homomorphic multiplications for a general q (and also requires Ring-GSW encryptions), which means directly applying their method would not meet our pre-set goal (Sect. 4.1). Additionally, this formula seems to preclude computing all the equality tests in parallel as we did for the prime case based on the linearity of the trace.

Solution. We will show that this goal can be achieved with only $O(\log \phi(q))$ homomorphic multiplications for the special case of prime-power cyclotomic rings (and only requires RLWE encryption of $\zeta_q^{\alpha-\beta}$). Our approach associates the equality test with the trace to exploit its linearity and computational efficiency. As the trace $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^{\alpha-\beta}) = \sum_{i\in\mathbb{Z}_q^*} \zeta_q^{(\alpha-\beta)\cdot i}$ can handle the powers in $\mathbb{Z}_q^*$, but the equality test requires the powers in $\mathbb{Z}_q$ (we use the representative set $\mathbb{Z}_q = [0, q-1]$), we thus first establish the following relation between $\mathbb{Z}_q^*$ (and its subgroups) and $\mathbb{Z}_q$, which is actually a partition of $\mathbb{Z}_q$ for a prime-power q.

Lemma 5.1 *For any prime number p and positive integer r, we have*

$$\mathbb{Z}_{p^r} \setminus \{0\} = \bigcup_{i\in[r]} p^{r-i} \cdot \mathbb{Z}_{p^i}^* \quad \Big(= \mathbb{Z}_{p^r}^* \cup p \cdot \mathbb{Z}_{p^{r-1}}^* \cup \cdots \cup p^{r-1} \cdot \mathbb{Z}_p^*\Big).$$

Proof. By definition, $\mathbb{Z}_{p^i}^*$ is the set of all numbers in $\mathbb{Z}_{p^i}$ that are coprime to p^i, which is equivalent to the set of all numbers in $\mathbb{Z}_{p^i}$ that are not divisible by p.

Since the set of all numbers in $\mathbb{Z}_{p^i}$ that are divisible by p is $p \cdot \mathbb{Z}_{p^{i-1}}$, we have $\mathbb{Z}_{p^i} = \mathbb{Z}^*_{p^i} \cup p \cdot \mathbb{Z}_{p^{i-1}}$ for all $i \in [r]$. Thus, by induction, we have

$$\begin{aligned}\mathbb{Z}_{p^r} &= \mathbb{Z}^*_{p^r} \cup p \cdot \mathbb{Z}_{p^{r-1}} \\ &= \mathbb{Z}^*_{p^r} \cup p \cdot \mathbb{Z}^*_{p^{r-1}} \cup p^2 \cdot \mathbb{Z}_{p^{r-2}} \\ &= \cdots \\ &= \mathbb{Z}^*_{p^r} \cup p \cdot \mathbb{Z}^*_{p^{r-1}} \cup \cdots \cup p^{r-1} \cdot \mathbb{Z}^*_p \cup \{0\},\end{aligned}$$

which completes the proof. □

Now, we are able to establish the following lemma that relates the algebraic trace to the equality test for the prime-power case.

Lemma 5.2 *For any $\alpha, \beta \in \mathbb{Z}_q$ where $q = p^r$ and p is any prime, we have*

$$\sum_{i \in \mathbb{Z}_q} \zeta_q^{(\alpha-\beta)\cdot i} = 1 + \sum_{i \in [r]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right)$$

Proof. We can verify that

$$\begin{aligned}\sum_{i \in \mathbb{Z}_q} \zeta_q^{(\alpha-\beta)\cdot i} &= \zeta_q^{(\alpha-\beta)\cdot 0} + \sum_{i \in [r]} \sum_{j \in p^{r-i} \cdot \mathbb{Z}^*_{p^i}} \zeta_q^{(\alpha-\beta)\cdot j} && \text{(by Lemma 5.1)} \\ &= 1 + \sum_{i \in [r]} \sum_{j \in \mathbb{Z}^*_{p^i}} \zeta_q^{p^{r-i}\cdot(\alpha-\beta)\cdot j} && \text{(modify the index } j\text{)} \\ &= 1 + \sum_{i \in [r]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right) && \text{(by } \star \text{ and the definition of trace)}\end{aligned}$$

where "$\star$" is the fact that $\zeta_q^{p^{r-i}} = \zeta_q^{q/p^i} = \zeta_{p^i} \in \mathbb{Z}[\zeta_{p^i}]$. □

To homomorphically evaluate this new equation of equality test however, we encounter the following new challenges.

Challenge 2. The input in each trace is of the form $\zeta_{p^i}^{\alpha-\beta}$. We need to efficiently compute this term from ζ_q^α, e.g., using $O(1)$ homomorphic-friendly operations.

Solution: We present an efficient method that only takes two homomorphic-friendly operations. Particularly, consider the automorphism $\tau : \zeta_q \mapsto \zeta_q^{p^{r-i}-1}$ as $p^{r-i}-1 \in \mathbb{Z}^*_q$ for $i \in [1, r-1]$, and then we have $\zeta_{p^i}^\alpha = \zeta_q^{p^{r-i}\cdot\alpha} = \zeta_q^{(p^{r-i}-1)\cdot\alpha} \cdot \zeta_q^\alpha = \tau(\zeta_q^\alpha) \cdot \zeta_q^\alpha$. For the homomorphic evaluation, this requires one homomorphic automorphism and one homomorphic multiplication, which would be roughly $O(1)$ homomorphic multiplication.

Challenge 3. If we compute the trace computations in the summation separately, the overall computational complexity is $\sum_{i \in [r]} O(\log \phi(p^i)) = O(\log^2 \phi(q))$. Although this is already much better than the $O(q)$ complexity of the method in [1], it still does not meet the requirements outlined in our blueprint (Sect. 4.1).

Algorithm 5.1. $\mathsf{EvalFunc}(\boldsymbol{c}, \mathsf{AutKey}, \mathsf{RelKey}, f)$

Parameters:

Δ: the scaling factor $\lfloor Q/(h \cdot q) \rceil$ of the input encoding
Δ': the scaling factor $\Delta \cdot q \approx Q/h$ of the output encoding
q: a prime power $q = p^r$ for some small prime p and integer $r > 1$
h: the plaintext modulus of the output ciphertext

Input:

A RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\Delta \cdot \zeta_q^{\alpha})$
The key for homomorphic automorphism evaluation AutKey
The relinearization key for BFV multiplication RelKey
An arbitrary function $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: A RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\Delta' \cdot f(\alpha))$.

$\boldsymbol{c}' \leftarrow \boldsymbol{c} \cdot \left(\sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_q^{-\beta}\right)$
$\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{p^{r-1}})}(\boldsymbol{c}', \mathsf{AutKey})$
for $i \in [1, r-1]$ **do**
  $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \mathsf{EvalAuto}(\boldsymbol{c}, \zeta_q \mapsto \zeta_q^{p^i - 1}, \mathsf{AutKey})$
  $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \mathsf{BFV.Mul}(\boldsymbol{c}, \boldsymbol{c}_{\mathtt{tmp}}, \mathsf{RelKey})$
  $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \boldsymbol{c}_{\mathtt{tmp}} \cdot \left(\sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_{p^{r-i}}^{-\beta}\right)$
  $\boldsymbol{c}' \leftarrow \boldsymbol{c}' + \boldsymbol{c}_{\mathtt{tmp}}$
  $\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{\mathbb{Q}(\zeta_{p^{r-i}})/\mathbb{Q}(\zeta_{p^{r-i-1}})}(\boldsymbol{c}', \mathsf{AutKey})$
end for
$\boldsymbol{c}' \leftarrow \boldsymbol{c}' + (\Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta), 0)^{\top}$
return $\boldsymbol{c}'$

Solution: We further observe that all the trace evaluations in the summation are contained in the tower of field extensions $\mathbb{Q}(\zeta_{p^r})/\mathbb{Q}(\zeta_{p^{r-1}})/\cdots/\mathbb{Q}(\zeta_p)/\mathbb{Q}$. Hence, we can directly evaluate the trace following the tower when p is small (e.g., $p = 2, 3$, and see Remark 5.4 and Sect. 6 for solutions for large p). The summation can be computed by adding the encryptions of $\sum f(\beta) \cdot \zeta_{p^i}^{\alpha-\beta}$ to the intermediate result during the evaluation of the trace tower. Consequently, the overall computational complexity is reduced to $O(\log \phi(q))$ since we only need to evaluate the trace once following the tower of field extensions.

We are now ready to present our construction. We provide the formal description of $\mathsf{EvalFunc}$ over prime-power cyclotomic rings in Algorithm 5.1 and its correctness along with analysis in Theorem 5.3.

Theorem 5.3 *Algorithm 5.1 is correct, i.e., the input-output behavior satisfies what is described. Moreover, it possesses the following properties:*

- *Complexity: it requires $O(\log \phi(q))$ times the cost of the external product.*
- *Error growth: if $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_\infty$, $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_\infty$ and $\|\sigma(\mathsf{Err}(\mathsf{RelKey}))\|_\infty$ are upper bounded by subgaussian variables with parameters δ_c, $\delta_{\mathtt{aut}}$ and δ_r, respectively. Let $\delta_{\mathtt{aut}}, \delta_r, \delta_z = O(\sqrt{N})$ and assume that $\delta_c < \Delta$, then we have $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ is upper bounded by a subgaussian variable with parameter*

$$O\left(Nh^2 q^{2.5} \sqrt{2\delta_c^2 + N^2 \log Q}\right).$$

Proof. We first analyze the ciphertext $\boldsymbol{c}_{\mathtt{tmp}}$ in the loop from *line* 3 to *line* 9.

- In *line* 4, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z(\Delta \cdot \zeta_q^{(p^i-1)\cdot\alpha})$ by Lemma 3.2.
- In *line* 5, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z(\Delta \cdot \zeta_q^{p^i\cdot\alpha}) \subset \mathsf{RLWE}_z(\Delta \cdot \zeta_{p^{r-i}}^{\alpha})$ by Lemma 3.1.
- In *line* 6, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z\left(\Delta \cdot \sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_{p^{r-i}}^{\alpha-\beta}\right)\right)$.

After *line* 9, we have the plaintext of $\boldsymbol{c}_{\mathtt{tmp}}$ in the i-th iteration (for $i \in [1, r-1]$) goes through the evaluation of

$$\begin{aligned}
&\mathsf{Tr}_{\mathbb{Q}(\zeta_p)/\mathbb{Q}}\left(\cdots \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^{r-i}})/\mathbb{Q}(\zeta_{p^{r-i-1}})}\left(\Delta\cdot\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_{p^{r-i}}^{\alpha-\beta}\right)\right)\cdots\right)\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^{r-i}})/\mathbb{Q}}\left(\Delta\cdot\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_{p^{r-i}}^{\alpha-\beta}\right)\right). \qquad \text{(by transitivity)}
\end{aligned}$$

Thus, after *line* 10, we have that $\boldsymbol{c}'$ encrypts

$$\begin{aligned}
&\Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta) + \sum_{i\in[0,r-1]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^{r-i}})/\mathbb{Q}}\left(\Delta\cdot\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_{p^{r-i}}^{\alpha-\beta}\right)\right)\\
&= \Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta) + \sum_{i\in[r]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\Delta\cdot\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_{p^{i}}^{\alpha-\beta}\right)\right) \qquad \text{(modify index)}\\
&= \Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta) + \Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\left(\sum_{i\in[r]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right)\right) \qquad \text{(by } \mathbb{Q}\text{-linearity)}\\
&= \Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\left(1+\sum_{i\in[r]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p^i})/\mathbb{Q}}\left(\zeta_{p^i}^{\alpha-\beta}\right)\right) \qquad \text{(combine terms)}\\
&= \Delta\cdot\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\begin{cases} q & \text{if } \alpha=\beta \mod q\\ 0 & \text{if } \alpha\neq\beta \mod q\end{cases} \qquad \text{(by Lemma 5.2 and Eqs. 5.1)}\\
&= \Delta\cdot f(\alpha)\cdot q \qquad \text{(only } f(\alpha) \text{ is multiplied by } q\text{, others are multiplied by 0)}\\
&= \Delta'\cdot f(\alpha), \qquad \text{(by the definition of } \Delta' := \Delta\cdot q)
\end{aligned}$$

which proves the correctness. Since the errors can be upper bounded by subgaussian variables, we summarize the corresponding subgaussian parameter in each line as follows.

- $\boldsymbol{c}'$ in *line* 1: $O(h\delta_c\sqrt{q})$ by the fact that $\|\sigma(\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\zeta_q^{-\beta})\|_\infty \le h\sqrt{q}$.
- $\boldsymbol{c}'$ in *line* 2: $O(\sqrt{pqh^2\delta_c^2 + (p-1)N^2\log Q})$ by Theorem 3.6.
- $\boldsymbol{c}_{\mathtt{tmp}}$ in *line* 4: $O(\sqrt{\delta_c^2 + N^2\log Q})$ by Lemma 3.2.
- $\boldsymbol{c}_{\mathtt{tmp}}$ in *line* 5: $O(hqN\sqrt{2\delta_c^2 + N^2\log Q})$ by Lemma 3.1 and $\delta_c < \Delta$.

– $\boldsymbol{c}_{\texttt{tmp}}$ in *line* 6: $O(Nh^2q^{1.5}\sqrt{2\delta_c^2 + N^2 \log Q})$ by the fact that

$$\left\| \sigma \left(\sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_{p^{r-i}}^{-\beta} \right) \right\|_\infty = O(h\sqrt{q}),$$

which is obtained by modeling $f(\beta)$ as a uniformly random variable in $\mathbb{Z}_h$.
– $\boldsymbol{c}'$ after *line* 9: $O(Nh^2q^{2.5}\sqrt{2\delta_c^2 + N^2 \log Q})$ by Theorem 3.6.
– $\boldsymbol{c}'$ in *line* 10: Adding an error-free ciphertext does not affect the error.

This proves the claim of error growth. For the computational complexity, we have $r = \log_p q = O(\log \phi(q))$ for the loop from *line* 3 to 9, and each loop contains one automorphism evaluation and one BFV multiplication, which implies $O(\log \phi(q))$ times the cost of external product by Proposition 3.7. Moreover, all the trace evaluation totally requires $O(\log \phi(q))$ times the cost of the external product by Proposition 3.7, which implies our claim of computational complexity. □

Remark 5.4 *The strategy for trace evaluation in Algorithm 5.1 can achieve a logarithmic complexity only when p is small (e.g., $p = 2, 3$). For larger p, we can employ a new general scaled equality test (see Lemma 6.6), allowing us to perform the trace evaluation of $\mathsf{Tr}_{\mathbb{Q}(\zeta_{p^r})/\mathbb{Q}}$ only once. The complexity in this case corresponds to the trace evaluation of the cyclic extension case as discussed in Sect. 3.3 and thus has a logarithmic complexity. Since this scenario is a special case of the situations we will address in the next section, we omit its details here.*

6 The Case of Composite Cyclotomic Rings

Now we move to the most general case where q is a composite number, i.e., $q = \prod_{i=1}^{k} q_i$, and q_i's are distinct prime-powers. Below, we first describe some critical intuitions and lemmas, and then present the final algorithm.

We first use the plaintext computation for an easier explanation of our idea. Recall our high-level goal of equality test: given $\zeta_q^{\alpha-\beta}$ (for some $\alpha, \beta \in \mathbb{Z}_q$), we can compute γ where $\gamma = q$ if $\alpha = \beta \bmod q$ or otherwise $\gamma = 0$, using some homomorphic-friendly operations. In the setting of composite $q = \prod_{i=1}^{k} q_i$, we identify several challenges where the techniques from the prior section do not carry through easily. Particularly, Eq. 5.1 still holds, but we can not directly apply Lemma 5.2 to relate it with algebraic trace. To tackle this, we identify the following equation:

$$\prod_{i=1}^{k} \left(\sum_{j \in \mathbb{Z}_{q_i}} \zeta_{q_i}^{(\alpha-\beta)\cdot j} \right) = \begin{cases} q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}. \tag{6.1}$$

The intuition is clear: $\alpha = \beta \bmod q$ if and only if $\alpha = \beta \bmod q_i$ for all the branches modulo q_i by the Chinese Remainder Theorem. As each $\sum_{j \in \mathbb{Z}_{q_i}} \zeta_{q_i}^{(\alpha-\beta)\cdot j}$

can serve as the equality test for the branch modulo q_i, one can easily verify the validity of this formula. Denote $q_i = p_i^{r_i}$ where p_i is prime for $i \in [k]$, we have

$$\prod_{i=1}^{k}\left(\sum_{j\in\mathbb{Z}_{q_i}} \zeta_{q_i}^{(\alpha-\beta)\cdot j}\right) = \prod_{i=1}^{k}\left(1+\sum_{j\in[r_i]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}}\left(\zeta_{p_i^j}^{\alpha-\beta}\right)\right) \tag{6.2}$$

by Lemma 5.2, which serves as our new equality test. To homomorphically evaluate this new equation, here comes the first (and main) challenge.

(Main) Challenge 1. While we can utilize the method in Sect. 5 to compute the trace summation, elegantly handling the outer product poses a challenging problem. Specifically, we need to avoid the direct consecutive use of BFV multiplication due to the rapid error growth it would introduce.

Solution. We elaborate a critical equivalent expression as Eq. 6.2. Before introducing our new equation, we need to first discuss several elegant properties we have discovered regarding trace computations in some special cases.

Lemma 6.1 *Suppose $q = q_1 q_2$ where q_1 and q_2 are coprime, then for any $i \in \mathbb{Z}$,*

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) = \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^i)\cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i).$$

Proof. Using the linearity and transitivity of the trace, we can verify that

$$\begin{aligned}
\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^i) &= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_{q_1}^i\cdot\zeta_{q_2}^i) && \text{(by coprimality of } q_1, q_2)\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i\cdot\zeta_{q_2}^i)\right) && \text{(by transitivity)}\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}\left(\zeta_{q_2}^i\cdot\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i)\right) && \text{(by } \mathbb{Q}(\zeta_{q_2})\text{-linearity)}\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i)\cdot\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i) && \text{(by } \star \text{ and } \mathbb{Q}\text{-linearity)}\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^i)\cdot\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^i) && \text{(by } \star)
\end{aligned}$$

where "$\star$" is the fact $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_{q_2})}(\zeta_{q_1}^i) = \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^i)\in\mathbb{Z}$. □

Corollary 6.2 *Suppose $q = \prod_{i=1}^{k} q_i$ where all the q_i's are distinct prime-powers, then for any $j \in \mathbb{Z}$,*

$$\prod_{i=1}^{k}\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_i})/\mathbb{Q}}(\zeta_{q_i}^j) = \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^j) \quad \left(=\mathsf{Tr}_{\mathbb{Q}(\zeta_{\prod_{i=1}^{k} q_i})/\mathbb{Q}}\left(\zeta_{\prod_{i=1}^{k} q_i}^j\right)\right).$$

Proof. By continuously using Lemma 6.1, we have

$$\begin{aligned}
\prod_{i=1}^{k}\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_i})/\mathbb{Q}}(\zeta_{q_i}^j) &= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1})/\mathbb{Q}}(\zeta_{q_1}^j)\cdot\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_2})/\mathbb{Q}}(\zeta_{q_2}^j)\cdots\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_k})/\mathbb{Q}}(\zeta_{q_k}^j)\\
&= \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1q_2})/\mathbb{Q}}(\zeta_{q_1q_2}^j)\cdot\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_3})/\mathbb{Q}}(\zeta_{q_3}^j)\cdots\mathsf{Tr}_{\mathbb{Q}(\zeta_{q_k})/\mathbb{Q}}(\zeta_{q_k}^j)\\
&= \cdots = \mathsf{Tr}_{\mathbb{Q}(\zeta_{q_1\cdots q_k})/\mathbb{Q}}(\zeta_{q_1\cdots q_k}^j),
\end{aligned}$$

which is exactly $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\zeta_q^j)$. □

Now, we can establish the following equivalent expression for Eq. 6.2.

Lemma 6.3 *For any $\alpha, \beta \in \mathbb{Z}_q$ where q is any positive integer with prime-power factorization $q = \prod_{i=1}^{k} p_i^{r_i}$, we have*

$$\prod_{i=1}^{k} \left(1 + \sum_{j \in [r_i]} \mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}} \left(\zeta_{p_i^j}^{\alpha-\beta}\right)\right) = 1 + \sum_{w|q, w \neq 1} \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}} \left(\zeta_w^{\alpha-\beta}\right).$$

Proof. After the direct expansion of the equation on the left, we have the summation of all possible products of $\mathsf{Tr}_{\mathbb{Q}(\zeta_{p_i^j})/\mathbb{Q}} \left(\zeta_{p_i^j}^{\alpha-\beta}\right)$ for each $i \in [k]$ and $j \in [r_i]$. Since all p_i^j's are distinct prime-powers, we can apply Corollary 6.2 to make the products of trace "into" the products of the indices of primitive roots of unity. These products actually traverse all the factors of q (except 1), so the result exactly matches the equation on the right. □

To homomorphically evaluate this new equation of equality test however, we encounter the following new challenges.

Challenge 2. The input in each trace is of the form $\zeta_w^{\alpha-\beta}$. We need to efficiently compute this term from ζ_q^{α}, i.e., using $O(1)$ homomorphic-friendly operations.

Solution. Similar to our solution to Challenge 2 in Sect. 5, we can write $\zeta_w^{\alpha} = \zeta_q^{(q/w)\cdot\alpha} = \zeta_q^{(q/w-1)\cdot\alpha} \cdot \zeta_q^{\alpha}$. Unfortunately, $\zeta_q \mapsto \zeta_q^{q/w-1}$ may not be an automorphism, so we need a more general formula that $\zeta_w^{\alpha} = \zeta_q^{(q/w)\cdot\alpha} = \zeta_q^{(q/w-c)\cdot\alpha} \cdot \zeta_q^{c\cdot\alpha}$ for some $c \in \mathbb{Z}_q$. We hope that both $q/w - c$ and c are in $\mathbb{Z}_q^*$, then we can use two automorphism evaluations plus one homomorphic multiplication to obtain the encryption of ζ_w^{α}. However, we can not always find such c for all $w \mid q, w \neq 1, q$ when q is an arbitrary number. Especially we can not choose an even q, the intuition is straightforward: For any odd factor p of q, we will encounter the case of computing ζ_q^p. Since p is odd, either $p - c$ or c is an even number, which implies either $p - c$ or c is not in $\mathbb{Z}_q^*$ as $2 \mid q$. Instead, we can prove its existence for any odd composite q as below.

Proposition 6.4 *Let q be a odd number has factorization $q = \prod_{i=1}^{k} p_i^{r_i}$ where p_i's are distinct odd primes and $k > 1$. For each $v \mid q, v \neq 1, q$, there exists at least one $c \in \mathbb{Z}_q^*$ such that $[v - c]_q \in \mathbb{Z}_q^*$.*

Proof. Suppose all the prime factors contained in v are p_i for $i \in S \subsetneq [k]$ and $S \neq \emptyset$. Then, we can construct c as $c = \left[\prod_{j \in [k] \setminus S} p_j - v\right]_q$. Now, we will prove that both c and $[v - c]_q$ are in $\mathbb{Z}_q^*$.

- Write $c = \prod_{j \in [k] \setminus S} p_j - v + q \cdot I \in \mathbb{Z}_q$ for some $I \in \mathbb{Z}$ (in fact, $I \in \{0, 1\}$). Suppose $c \notin \mathbb{Z}_q^*$, which implies that c and q share at least one common prime

factor p. Namely, $p \in \{p_i\}_{i\in[k]}$ and $p \mid c$. Then

$$\text{If } p \in \{p_i\}_{i\in S} \Longrightarrow p \mid v \overset{p|q}{\Longrightarrow} p \mid (v - q \cdot I) \overset{p|c}{\Longrightarrow} p \mid \prod_{j\in[k]\setminus S} p_j, \text{ a contradiction.}$$

$$\text{If } p \in \{p_j\}_{j\in[k]\setminus S} \overset{p|q}{\Longrightarrow} p \mid \prod_{j\in[k]\setminus S} p_j + q \cdot I \overset{p|c}{\Longrightarrow} p \mid v, \text{ a contradiction.}$$

Hence, c and q share no common prime factor, which implies $c \in \mathbb{Z}_q^*$.

– Write $[v - c]_q = 2 \cdot v - \prod_{j\in[k]\setminus S} p_j + q \cdot J \in \mathbb{Z}_q$ for some $J \in \mathbb{Z}$, we can easily show $[v - c] \in \mathbb{Z}_q^*$ by a quite similar argument as the prior proof of $c \in \mathbb{Z}_q^*$ since q does not contain the factor 2.

Now, the remaining case is $S = [k]$, i.e., v contains all the prime factors of q. In this case, we can easily verify that $v - 1 \in \mathbb{Z}_q^*$, which completes the proof. □

Remark 6.5 *Our proof of Proposition 6.4 is constructive and provides a direct method to determine one desired c for any v. However, for many values of v, we observe by experiments using* `SageMath` [68] *that $c = 1$ is also usable. We prioritize a usable $v - 1$ as it saves one homomorphic automorphism evaluation.*

Challenge 3. There exist many different trace computations in the summation (Lemma 6.3), where some of them may not contained in a consecutive tower down to $\mathbb{Q}$. Thus, we can not apply the strategy in Sect. 5, and computing them separately will not meet the pre-set goal in our blueprint (Sect. 4.1).

Solution. We present a new technique that "swaps" the order of summation and trace, meaning that we can first aggregate the inputs and just compute the trace once. Particularly, we prove the following lemma as our final equality test:

Lemma 6.6 *For any positive integer $q > 1$ and any $\alpha, \beta \in \mathbb{Z}_q$, we have*

$$\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1 + \sum_{w|q, w\neq 1} \phi(w) \cdot \zeta_w^{\alpha-\beta}\right) = \begin{cases} \phi(q) \cdot q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases}.$$

Proof. We can verify that

$$\begin{aligned}
&\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1 + \sum_{w|q, w\neq 1} \phi(w) \cdot \zeta_w^{\alpha-\beta}\right) \\
&= \phi(q) + \sum_{w|q, w\neq 1} \phi(w) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right) && \text{(by } \mathbb{Q}\text{-linearity)} \\
&= \phi(q) + \sum_{w|q, w\neq 1} \phi(w) \cdot \frac{\phi(q)}{\phi(w)} \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right) && (\star) \\
&= \phi(q) \cdot \left(1 + \sum_{w|q, w\neq 1} \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right)\right) && \text{(combine terms)} \\
&= \begin{cases} \phi(q) \cdot q & \text{if } \alpha = \beta \mod q \\ 0 & \text{if } \alpha \neq \beta \mod q \end{cases} && \text{(by Lemma 6.3 and Eq. 6.1, 6.2)}
\end{aligned}$$

Algorithm 6.1. $\mathsf{EvalFunc}(\boldsymbol{c}, \mathsf{AutKey}, \mathsf{RelKey}, f)$

Parameters:

Δ: the scaling factor $\lfloor Q/(h \cdot q \cdot \phi(q)) \rceil$ of the input encoding
Δ': the scaling factor $\Delta \cdot q \cdot \phi(q) \approx Q/h$ of the output encoding
q: an odd number satisfying $q \mid m$ with prime-power factorization $q = \prod_{i \in [k]} p_i^{r_i}$
$\{c_v\}$: the automorphism index set $\{c_v\}_{v|q, v \neq 1, q} \subset \mathbb{Z}_q^*$ obtained by Proposition 6.4 and Remark 6.5 such that $\{v - c_v\}_{v|q, v \neq 1, q} \subset \mathbb{Z}_q^*$.
h: the plaintext modulus of the output ciphertext

Input:

A RLWE ciphertext $\boldsymbol{c} \in \mathsf{RLWE}_z(\Delta \cdot \zeta_q^{\alpha})$
The key for homomorphic automorphism evaluation AutKey
The relinearization key for BFV multiplication RelKey
An arbitrary function $f : \mathbb{Z}_q \to \mathbb{Z}_h$

Output: A RLWE ciphertext $\boldsymbol{c}' \in \mathsf{RLWE}_z(\Delta' \cdot f(\alpha))$.

$\boldsymbol{c}' \leftarrow \boldsymbol{c} \cdot \phi(q) \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta) \cdot \zeta_q^{-\beta}$
for $w \mid q$ and $w \neq 1, q$ **do**
  $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \mathsf{EvalAuto}(\boldsymbol{c}, \zeta_q \mapsto \zeta_q^{q/w - c_{q/w}}, \mathsf{AutKey})$
  if $c_{q/w} = 1$ **then**
    $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \mathsf{BFV.Mul}(\boldsymbol{c}, \boldsymbol{c}_{\mathtt{tmp}}, \mathsf{RelKey})$
  else
    $\boldsymbol{c}'_{\mathtt{tmp}} \leftarrow \mathsf{EvalAuto}(\boldsymbol{c}, \zeta_q \mapsto \zeta_q^{c_{q/w}}, \mathsf{AutKey})$
    $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \mathsf{BFV.Mul}(\boldsymbol{c}'_{\mathtt{tmp}}, \boldsymbol{c}_{\mathtt{tmp}}, \mathsf{RelKey})$
  end if
  $\boldsymbol{c}_{\mathtt{tmp}} \leftarrow \boldsymbol{c}_{\mathtt{tmp}} \cdot \phi(w) \cdot \left(\sum_{\beta \in \mathbb{Z}_q} \left(f(\beta) \cdot \zeta_w^{-\beta} \right) \right)$
  $\boldsymbol{c}' \leftarrow \boldsymbol{c}' + \boldsymbol{c}_{\mathtt{tmp}}$
end for
$\boldsymbol{c}' \leftarrow \boldsymbol{c}' + (\Delta \cdot \sum_{\beta \in \mathbb{Z}_q} f(\beta), 0)^{\top}$
$\boldsymbol{c}' \leftarrow \mathsf{EvalTr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}(\boldsymbol{c}', \mathsf{AutKey})$
return $\boldsymbol{c}'$

where $(\star)$ follows from the fact that

$$\begin{aligned} \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta}\right) &= \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_w)}(\zeta_w^{\alpha-\beta})\right) && \text{(by transitivity)} \\ &= \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}\left(\zeta_w^{\alpha-\beta} \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_w)}(1)\right) && \text{(by } \mathbb{Q}(\zeta_w)\text{-linearity)} \\ &= (\phi(q)/\phi(w)) \cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_w)/\mathbb{Q}}(\zeta_w^{\alpha-\beta}) \end{aligned}$$

by $\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}(\zeta_w)}(1) = [\mathbb{Q}(\zeta_q) : \mathbb{Q}(\zeta_w)] = \phi(q)/\phi(w)$ and $\mathbb{Q}$-linearity. □

By using this form of the equality test, we are able to design Algorithm 6.1. The following theorem demonstrates its correctness and analysis.

Theorem 6.7 *Algorithm 6.1 is correct, i.e., the input-output behavior satisfies what is described. Let* $r = \prod_{i \in [k]}(r_i + 1)$*, it possesses the following properties:*

- *Complexity: it requires* $O(\log \phi(q) + r)$ *times the cost of the external product.*
- *Error growth: if* $\|\sigma(\mathsf{Err}(\boldsymbol{c}))\|_{\infty}$*,* $\|\sigma(\mathsf{Err}(\mathsf{AutKey}))\|_{\infty}$ *and* $\|\sigma(\mathsf{Err}(\mathsf{RelKey}))\|_{\infty}$ *are upper bounded by subgaussian variables with parameters* δ_c*,* $\delta_{\mathtt{aut}}$ *and* δ_r*,*

respectively. Let $\delta_{\mathtt{aut}}, \delta_r, \delta_z = O(\sqrt{N})$ *and assume* $\delta_c < \Delta$*, then the output error* $\|\sigma(\mathsf{Err}(\boldsymbol{c}'))\|_\infty$ *is upper bounded by a subgaussian variable with parameter* $O(rh^2Nq^4\sqrt{\delta_c^2 + N^2 \log Q})$.

Proof. We start with a step-by-step analysis of the loop from *line* 2 to *line* 12.

- In *line* 3, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z\left(\Delta \cdot \zeta_q^{(q/w - c_{q/w})\cdot\alpha}\right)$ by Lemma 3.2.
- In *line* 5, if $c_{q/w} = 1$, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z\left(\Delta \cdot \zeta_q^{(q/w)\cdot\alpha}\right) \subset \mathsf{RLWE}_z\left(\Delta \cdot \zeta_w^{\alpha}\right)$ by Lemma 3.1.
- In *line* 6–9, if $c_{q/w} \neq 1$, then $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z\left(\Delta \cdot \zeta_q^{(q/w)\cdot\alpha}\right) \subset \mathsf{RLWE}_z\left(\Delta \cdot \zeta_w^{\alpha}\right)$ by Lemma 3.1 and 3.2.
- In *line* 10, we have $\boldsymbol{c}_{\mathtt{tmp}} \in \mathsf{RLWE}_z\left(\Delta \cdot \phi(w) \cdot \left(\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_w^{\alpha-\beta}\right)\right)\right)$.

Hence, after *line* 14, we have $\boldsymbol{c}'$ encrypts

$$
\begin{aligned}
&\mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(\Delta \cdot \sum_{\beta\in\mathbb{Z}_q} f(\beta) + \sum_{w|q, w\neq 1} \Delta\cdot\phi(w)\cdot\left(\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_w^{\alpha-\beta}\right)\right)\right)\\
&= \Delta \cdot \sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot \mathsf{Tr}_{\mathbb{Q}(\zeta_q)/\mathbb{Q}}\left(1 + \sum_{w|q,w\neq 1}\phi(w)\cdot\zeta_w^{\alpha-\beta}\right) && \text{(by } \mathbb{Q}\text{-linearity)}\\
&= \Delta \cdot \sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot \begin{cases}\phi(q)\cdot q & \text{if } [\alpha-\beta]_q = 0\\ 0 & \text{otherwise}\end{cases} && \text{(by Lemma 6.6)}\\
&= \Delta\cdot f(\alpha)\cdot\phi(q)\cdot q && \text{(only } f(\alpha) \text{ is multiplied by } \phi(q)\cdot q\text{, others are multiplied by 0)}\\
&= \Delta'\cdot f(\alpha), && \text{(by the definition of } \Delta' := \Delta\cdot q\cdot\phi(q))
\end{aligned}
$$

which proves the correctness. Since the errors can be upper bounded by subgaussian variables, we summarize the corresponding subgaussian parameter in each line as follows.

- $\boldsymbol{c}'$ in *line* 1: $O(h\delta_c\phi(q)\sqrt{q})$ by the fact that $\|\sigma(\sum_{\beta\in\mathbb{Z}_q} f(\beta)\cdot\zeta_q^{-\beta})\|_\infty \le h\sqrt{q}$.
- $\boldsymbol{c}_{\mathtt{tmp}}$ in *line* 3: $O(\sqrt{\delta_c^2 + N^2\log Q})$ by Lemma 3.2.
- $\boldsymbol{c}_{\mathtt{tmp}}$ in *line* 4–9: $O(hNq^2\sqrt{\delta_c^2 + N^2\log Q})$ by Lemma 3.1 and $\delta_c < \Delta$ (we use the worst case that $c_{q/w} \neq 1$).
- $\boldsymbol{c}_{\mathtt{tmp}}$ in *line* 10: $O(h^2Nq^{3.5}\sqrt{\delta_c^2 + N^2\log Q})$ by plaintext-ciphertext multiplication and the fact that $\left\|\sigma\left(\phi(w)\cdot\left(\sum_{\beta\in\mathbb{Z}_q}\left(f(\beta)\cdot\zeta_w^{-\beta}\right)\right)\right)\right\|_\infty = O(hq^{1.5})$, which is obtained by modeling $f(\beta)$ as a uniformly random variable in $\mathbb{Z}_h$.
- $\boldsymbol{c}'$ in *line* 13: $O(rh^2Nq^{3.5}\sqrt{\delta_c^2 + N^2\log Q})$ by the fact that we have $r-2$ loops from *line* 2 to *line* 12.
- $\boldsymbol{c}'$ in *line* 14: $O(rh^2Nq^4\sqrt{\delta_c^2 + N^2\log Q})$ by Theorem 3.6.

This proves the claim of error growth. For the computational complexity, we have $r-2$ for the loop from *line* 2 to 12, and each loop contains at most two automorphism evaluations and one BFV multiplication, which implies $O(r)$ times the cost of the external product by Proposition 3.7. Finally, the final trace evaluation requires $O(\log \phi(q))$ times the cost of the external product by Proposition 3.7, which completes the proof of our claim of computational complexity. □

Remark 6.8 *In the error analysis of Algorithm 6.1, we use rather loose estimates, i.e., $\phi(w) < q$ for all $w \mid q, w \neq 1$. These terms may be much smaller than q in the concrete parameter settings. Moreover, there is a term r in the upper bound of the overall error norm. As r is generally sub-linear in q [55, 67], we note that the overall term is still polynomially bounded (in the security parameter).*

Asymptotic Setting. In the asymptotic setting, we can set $r = O(\log \phi(q))$ or fix r to some small constant. Then we have that the overall computational complexity of $\mathsf{EvalFunc}$ is $O(\log \phi(q))$ as we desired in our blueprint in Sect. 4.1.

Parallel Computation. Our algorithm $\mathsf{EvalFunc}$ is friendly to the parallel computation architecture, as the computation in each for loop (from *line* 2 to *line* 12) does not have a dependency on the others. Additionally, the homomorphic trace is also parallel friendly, as pointed out by [38]. Thus, the overall throughput can be easily improved on parallel-friendly (e.g., multi-core) platforms.

7 Conclusions and Future Work

In this work, we show for the first time that functional bootstrapping for general functions within a polynomial modulus can work over a wide range of general cyclotomic rings and *simultaneously* satisfy the following two properties:

- Supporting arbitrary correctly decryptable input LWE ciphertexts.
- Essentially as efficient as regular bootstrapping.

Based on the advantages and limitations of our current techniques, we also notice two interesting directions that warrant further investigation:

- **Theory:** As mentioned in the solution to Challenge 2 of Sect. 6, our method cannot handle cyclotomic rings with even composite indices, namely when the cyclotomic index is composite and contains the prime factor 2. Overcoming this limitation to support arbitrary cyclotomic rings while maintaining functionality and efficiency presents a significant challenge.
- **Practice:** Note that our algorithms support cyclotomic rings with indices that are powers of small primes (e.g., powers-of-2), which have been shown to be concretely efficient due to their full compatibility with the standard (or slightly twisted) Fast Fourier Transform (FFT) [24,30,73] and Number Theoretic Transform (NTT) [3,20]. It is also promising for cyclotomic rings with composite indices by combining the Batch framework of [44,45] with insights from [51,56]. We leave it as an interesting follow-up work to determine the concrete efficiency of our algorithms over different cyclotomic rings.

Acknowledgements. We would like to thank the anonymous reviewers of TCC 2024 for their insightful advices that help improve the presentation. Feng-Hao Liu is supported by NSF CNS-2402031. Han Xia and Han Wang are supported by the National Key R&D Program of China under Grant 2020YFA0712303, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDB0690200, and State Key Laboratory of Information Security under Grant TC20221013042.

References

1. Abla, P., Liu, F.H., Wang, H., Wang, Z.: Ring-based identity based encryption - asymptotically shorter MPK and tighter security. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part III. LNCS, vol. 13044, pp. 157–187. Springer, Cham (Nov 2021). https://doi.org/10.1007/978-3-030-90456-2_6
2. Agrawal, S., Lin, D. (eds.): ASIACRYPT 2022, Part II, LNCS, vol. 13792. Springer, Cham (Dec (2022)
3. Al Badawi, A., et al.: OpenFHE: Open-source fully homomorphic encryption library. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp. 53–63. WAHC'22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3560827.3563379
4. Alperin-Sheriff, J., Peikert, C.: Practical bootstrapping in quasilinear time. In: Canetti and Garay [17], pp. 1–20. https://doi.org/10.1007/978-3-642-40041-4_1
5. Alperin-Sheriff, J., Peikert, C.: Faster bootstrapping with polynomial error. In: Garay and Gennaro [32], pp. 297–314. https://doi.org/10.1007/978-3-662-44371-2_17
6. Barrington, D.A.M.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In: 18th ACM STOC, pp. 1–5. ACM Press (May 1986). https://doi.org/10.1145/12130.12131
7. Bergerat, L., et al.: Parameter optimization and larger precision for (T)FHE. J. Cryptol. **36**(3), 28 (2023). https://doi.org/10.1007/s00145-023-09463-5
8. Biasse, J.F., Ruiz, L.: FHEW with efficient multibit bootstrapping. In: Lauter, K.E., Rodríguez-Henríquez, F. (eds.) LATINCRYPT 2015. LNCS, vol. 9230, pp. 119–135. Springer, Cham (Aug 2015). https://doi.org/10.1007/978-3-319-22174-8_7
9. Bonte, C., Iliashenko, I., Park, J., Pereira, H.V.L., Smart, N.P.: FINAL: Faster FHE instantiated with NTRU and LWE. In: Agrawal and Lin [2], pp. 188–215. https://doi.org/10.1007/978-3-031-22966-4_7
10. Boura, C., Gama, N., Georgieva, M., Jetchev, D.: Simulating homomorphic evaluation of deep learning predictions. In: International Symposium on Cyber Security Cryptography and Machine Learning, pp. 212–230. Springer (2019). https://doi.org/10.1007/978-3-030-20951-3_20
11. Bourse, F., Sanders, O., Traoré, J.: Improved secure integer comparison via homomorphic encryption. In: Jarecki, S. (ed.) CT-RSA 2020. LNCS, vol. 12006, pp. 391–416. Springer, Cham (Feb 2020). https://doi.org/10.1007/978-3-030-40186-3_17
12. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini and Canetti [64], pp. 868–886. https://doi.org/10.1007/978-3-642-32009-5_50
13. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012, pp. 309–325. ACM (Jan 2012). https://doi.org/10.1145/2090236.2090262

14. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) 45th ACM STOC, pp. 575–584. ACM Press (Jun 2013). https://doi.org/10.1145/2488608.2488680
15. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) 52nd FOCS, pp. 97–106. IEEE Computer Society Press (Oct 2011). https://doi.org/10.1109/FOCS.2011.12
16. Brakerski, Z., Vaikuntanathan, V.: Lattice-based FHE as secure as PKE. In: Naor, M. (ed.) ITCS 2014, pp. 1–12. ACM (Jan 2014). https://doi.org/10.1145/2554797.2554799
17. Canetti, R., Garay, J.A. (eds.): CRYPTO 2013, Part I, LNCS, vol. 8042. Springer, Berlin, Heidelberg (Aug (2013)
18. Castryck, W., Iliashenko, I., Vercauteren, F.: Provably weak instances of ring-lwe revisited. In: Fischlin, M., Coron, J.-S. (eds.) Advances in Cryptology – EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I, pp. 147–167. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49890-3_6
19. Chen, H., Dai, W., Kim, M., Song, Y.: Efficient homomorphic conversion between (Ring) LWE Ciphertexts. In: Sako, K., Tippenhauer, N.O. (eds.) Applied Cryptography and Network Security: 19th International Conference, ACNS 2021, Kamakura, Japan, June 21–24, 2021, Proceedings, Part I, pp. 460–479. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-78372-3_18
20. Chen, H., Laine, K., Player, R.: Simple encrypted arithmetic library - SEAL v2.1. In: Brenner, M., Rohloff, K., Bonneau, J., Miller, A., Ryan, P.Y.A., Teague, V., Bracciali, A., Sala, M., Pintore, F., Jakobsson, M. (eds.) FC 2017 Workshops. LNCS, vol. 10323, pp. 3–18. Springer, Cham (Apr 2017). https://doi.org/10.1007/978-3-319-70278-0_1
21. Cheon, J.H., Kim, A., Kim, M., Song, Y.S.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 409–437. Springer, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70694-8_15
22. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Faster Fully Homomorphic Encryption: Bootstrapping in Less Than 0.1 seconds. In: Cheon, J.H., Takagi, T. (eds.) 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, pp. 3–33. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_1
23. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: fast fully homomorphic encryption over the torus. J. Cryptol. **33**(1), 34–91 (2019). https://doi.org/10.1007/s00145-019-09319-x
24. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: TFHE: Fast fully homomorphic encryption library. GitHub (2023). https://github.com/tfhe/tfhe
25. Chillotti, I., Joye, M., Paillier, P.: Programmable bootstrapping enables efficient homomorphic inference of deep neural networks. In: Cyber Security Cryptography and Machine Learning: 5th International Symposium, CSCML 2021, Be'er Sheva, Israel, July 8–9, 2021, Proceedings 5, pp. 1–19. Springer (2021). https://doi.org/10.1007/978-3-030-78086-9_1

26. Chillotti, I., Ligier, D., Orfila, J.B., Tap, S.: Improved programmable bootstrapping with larger precision and efficient arithmetic circuits for TFHE. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part III. LNCS, vol. 13092, pp. 670–699. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92078-4_23
27. Clet, P.E., Zuber, M., Boudguiga, A., Sirdey, R., Gouy-Pailler, C.: Putting up the swiss army knife of homomorphic calculations by means of TFHE functional bootstrapping. Cryptology ePrint Archive, Report 2022/149 (2022). https://eprint.iacr.org/2022/149
28. Cong, K., Das, D., Park, J., Pereira, H.V.L.: SortingHat: Efficient private decision tree evaluation via homomorphic encryption and transciphering. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022, pp. 563–577. ACM Press (Nov 2022). https://doi.org/10.1145/3548606.3560702
29. Crockett, E., Peikert, C.: Challenges for ring-LWE. Cryptology ePrint Archive, Report 2016/782 (2016). https://eprint.iacr.org/2016/782
30. Ducas, L., Micciancio, D.: FHEW: bootstrapping homomorphic encryption in less than a second. In: Oswald and Fischlin [58], pp. 617–640. https://doi.org/10.1007/978-3-662-46800-5_24
31. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144 (2012). https://eprint.iacr.org/2012/144
32. Garay, J.A., Gennaro, R. (eds.): CRYPTO 2014, Part I, LNCS, vol. 8616. Springer, Berlin, Heidelberg (Aug (2014)
33. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher [54], pp. 169–178. https://doi.org/10.1145/1536414.1536440
34. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluation of the AES circuit. In: Safavi-Naini and Canetti [64], pp. 850–867. https://doi.org/10.1007/978-3-642-32009-5_49
35. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti and Garay [17], pp. 75–92. https://doi.org/10.1007/978-3-642-40041-4_5
36. Guimarães, A., Borin, E., Aranha, D.F.: Revisiting the functional bootstrap in TFHE. IACR TCHES **2021**(2), 229–253 (2021). https://doi.org/10.46586/tches.v2021.i2.229-253, https://tches.iacr.org/index.php/TCHES/article/view/8793
37. Halevi, S., Shoup, V.: Algorithms in HElib. In: Garay and Gennaro [32], pp. 554–571. https://doi.org/10.1007/978-3-662-44371-2_31
38. Halevi, S., Shoup, V.: Bootstrapping for HElib. In: Oswald and Fischlin [58], pp. 641–670. https://doi.org/10.1007/978-3-662-46800-5_25
39. Halevi, S., Shoup, V.: Design and implementation of HElib: a homomorphic encryption library. Cryptology ePrint Archive, Report 2020/1481 (2020). https://eprint.iacr.org/2020/1481
40. Hazay, C., Stam, M. (eds.): EUROCRYPT 2023, Part III, LNCS, vol. 14006. Springer, Cham (Apr (2023)
41. Joye, M., Walter, M.: Liberating TFHE: programmable bootstrapping with general quotient polynomials. In: Proceedings of the 10th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, pp. 1–11 (2022). https://doi.org/10.1145/3560827.3563376
42. Kluczniak, K., Schild, L.: FDFB: Full domain functional bootstrapping towards practical fully homomorphic encryption. IACR TCHES **2023**(1), 501–537 (2023). https://doi.org/10.46586/tches.v2023.i1.501-537
43. Lee, Y., et al.: Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption. In: Hazay and Stam [40], pp. 227–256. https://doi.org/10.1007/978-3-031-30620-4_8

44. Liu, F.H., Wang, H.: Batch bootstrapping I: a new framework for SIMD bootstrapping in polynomial modulus. In: Hazay and Stam [40], pp. 321–352. https://doi.org/10.1007/978-3-031-30620-4_11
45. Liu, F.H., Wang, H.: Batch bootstrapping II: bootstrapping in polynomial modulus only requires $\tilde{O}(1)$ FHE multiplications in amortization. In: Hazay and Stam [40], pp. 353–384. https://doi.org/10.1007/978-3-031-30620-4_12
46. Katsikas, S., Abie, H., Ranise, S., Verderame, L., Cambiaso, E., Ugarelli, R., Praça, I., Li, W., Meng, W., Furnell, S., Katt, B., Pirbhulal, S., Shukla, A., Ianni, M., Dalla Preda, M., Choo, K.-K.R., Pupo Correia, M., Abhishta, A., Sileno, G., Alishahi, M., Kalutarage, H., Yanai, N. (eds.): Computer Security. ESORICS 2023 International Workshops: CPS4CIP, ADIoT, SecAssure, WASP, TAURIN, PriST-AI, and SECAI, The Hague, The Netherlands, September 25–29, 2023, Revised Selected Papers, Part II. Springer Nature Switzerland, Cham (2024)
47. Liu, Z., Micciancio, D., Polyakov, Y.: Large-precision homomorphic sign evaluation using FHEW/TFHE bootstrapping. In: Agrawal and Lin [2], pp. 130–160. https://doi.org/10.1007/978-3-031-22966-4_5
48. Liu, Z., Wang, Y.: Amortized functional bootstrapping in less than 7 ms, with $\tilde{\mathrm{O}}(1)$ polynomial multiplications. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VI. LNCS, vol. 14443, pp. 101–132. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8736-8_4
49. Lu, W.J., Huang, Z., Hong, C., Ma, Y., Qu, H.: PEGASUS: Bridging polynomial and non-polynomial evaluations in homomorphic encryption. In: 2021 IEEE Symposium on Security and Privacy. pp. 1057–1073. IEEE Computer Society Press (May 2021). https://doi.org/10.1109/SP40001.2021.00043
50. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Berlin, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_1
51. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology – EUROCRYPT 2013: 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings, pp. 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_3
52. Ma, S., Huang, T., Wang, A., Zhou, Q., Wang, X.: Fast and accurate: efficient full-domain functional bootstrap and digit decomposition for homomorphic computation. IACR TCHES **2024**(1), 592–616 (2024). https://doi.org/10.46586/tches.v2024.i1.592-616
53. Micciancio, D., Polyakov, Y.: Bootstrapping in FHEW-like cryptosystems. In: WAHC '21: Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography, Virtual Event, Korea, 15 November 2021, pp. 17–28. WAHC@ACM (2021). https://doi.org/10.1145/3474366.3486924
54. Mitzenmacher, M. (ed.): 41st ACM STOC. ACM Press (May / Jun 2009)
55. from MO (https://mathoverflow.net/users/11919/gh-from mo), G.: Upper bound for product of exponents of prime factorization. MathOverflow. https://mathoverflow.net/q/256452
56. Open Source: HElib. GitHub. https://github.com/shaih/HElib
57. Open Source: Palisade lattice cryptography library. GitLab. https://gitlab.com/palisade
58. Oswald, E., Fischlin, M. (eds.): EUROCRYPT 2015, Part I, LNCS, vol. 9056. Springer, Berlin, Heidelberg (Apr (2015)

59. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Mitzenmacher [54], pp. 333–342. https://doi.org/10.1145/1536414.1536461
60. Peikert, C.: How (not) to instantiate ring-LWE. In: Zikas, V., De Prisco, R. (eds.) SCN 16. LNCS, vol. 9841, pp. 411–430. Springer, Cham (Aug / Sep 2016). https://doi.org/10.1007/978-3-319-44618-9_22
61. Peikert, C., Pepin, Z.: Algebraically structured LWE, revisited. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part I. LNCS, vol. 11891, pp. 1–23. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-36030-6_1
62. Peikert, C., Regev, O., Stephens-Davidowitz, N.: Pseudorandomness of ring-LWE for any ring and modulus. In: Hatami, H., McKenzie, P., King, V. (eds.) 49th ACM STOC, pp. 461–473. ACM Press (Jun 2017). https://doi.org/10.1145/3055399.3055489
63. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (May 2005). https://doi.org/10.1145/1060590.1060603
64. Safavi-Naini, R., Canetti, R. (eds.): CRYPTO 2012, LNCS, vol. 7417. Springer, Berlin, Heidelberg (Aug (2012)
65. Smart, N.P., Vercauteren, F.: Fully homomorphic SIMD operations. Des. Codes Crypt. **71**(1), 57–81 (2012). https://doi.org/10.1007/s10623-012-9720-4
66. Stehlé, D., Steinfeld, R., Tanaka, K., Xagawa, K.: Efficient public key encryption based on ideal lattices. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 617–635. Springer, Berlin, Heidelberg (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_36
67. Tenenbaum, G.: Introduction to analytic and probabilistic number theory, vol. 163. American Mathematical Soc. (2015)
68. The Sage Developers: Sagemath, the Sage Mathematics Software System (Version 10.2) (2023). https://www.sagemath.org
69. Vinogradov, I.M.: Chapter VI: Primitive roots and indices. In: Elements of number theory. pp. 105–121. Dover Publications (2003). https://books.google.com/books?id=xlIfdGPM9t4C&pg=PA105
70. Wang, R., et al.: Circuit bootstrapping: Faster and smaller. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 342–372. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58723-8_12
71. Xiang, B., Zhang, J., Deng, Y., Dai, Y., Feng, D.: Fast blind rotation for bootstrapping FHEs. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part IV. LNCS, vol. 14084, pp. 3–36. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38551-3_1
72. Yang, Z., Xie, X., Shen, H., Chen, S., Zhou, J.: TOTA: Fully homomorphic encryption with smaller parameters and stronger security. Cryptology ePrint Archive, Report 2021/1347 (2021). https://eprint.iacr.org/2021/1347
73. Zama: TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data (2022). https://github.com/zama-ai/tfhe-rs
74. Zheng, X., Li, H., Wang, D.: A new framework for fast homomorphic matrix multiplication. Cryptology ePrint Archive, Paper 2023/1649 (2023). https://eprint.iacr.org/2023/1649